Dissertation

University of Bamberg

Information Systems and Applied Computer Sciences

Spatial Composition

Integrating Recommendations into Spatial Hypertext

Daniel Roßner

0000-0002-2539-569X

Bamberg 2025

Diese Arbeit hat der Fakultät *Wirtschaftsinformatik und Angewandte Informatik* der Otto-Friedrich-Universität Bamberg als Dissertation vorgelegen.

Gutachter
Prof. Dr. Tom Gross

Gutachter
Prof. Dr. Claus Atzenbeck

Tag der mündlichen Prüfung
15. Mai 2025

URN: urn:nbn:de:bvb:473-irb-1083129

DOI: https://doi.org/10.20378/irb-108312

Zusammenfassung

Diese Arbeit beschäftigt sich mit der Integration von Vorschlägen in Spatial-Hypertext-Anwendungen. Spatial Hypertext beschreibt in der Regel 2D-Oberflächen, innerhalb derer Anwender Objekte hinzufügen, verschieben und somit räumlich anordnen können. Hypertext meint normalerweise die Verknüpfung von Dokumenten; im Falle von Spatial Hypertext erfolgt diese Verknüpfung jedoch nicht explizit, sondern implizit durch visuelle Attribute. Dabei spielen vor allem die Position und die Anordnung von Objekten eine wichtige Rolle. Ein solches System lässt sich mit systemgenerierten Vorschlägen ergänzen: Die vom Anwender angelegten räumlichen Strukturen werden interpretiert; eine Wissensbank liefert dann zu Inhalt und Struktur passende Vorschläge.

Eine bisher unbeantwortete Frage ist die nach der Darstellung der Vorschläge. Der Fokus dieser Arbeit liegt darin, Methoden zu begründen, zu implementieren und zu untersuchen, wie Vorschläge sinnvoll in die bestehenden visuellen Strukturen eingebettet werden können. Dies soll in einer Weise geschehen, die es dem Anwender einfacher macht, Zusammenhänge zu verstehen und somit eine effizientere und effektivere Wissensexploration ermöglicht. In diesem Zusammenhang wird der Begriff „Spatial Composition" eingeführt – angelehnt an Verfahren, die visuelle Strukturen interpretieren („Spatial Parsing").

Es werden exemplarisch fünf Algorithmen entwickelt und beschrieben, die dieses Ziel erreichen sollen. Deren Beschreibung erfolgt chronologisch, sodass aufgetretene Probleme und deren Lösungen ersichtlich werden. Die Algorithmen werden untersucht und anhand verschiedener Merkmale – die im Kontext von Spatial Composition als wichtig definiert werden – verglichen. Eine Nutzerstudie untersucht, welche Auswirkungen die Integration von Vorschlägen in die räumlichen Strukturen auf Effektivität und Effizienz hat, im Vergleich zu einer Darstellung in Listenform. Darüber hinaus werden die Gründe dafür untersucht, auch vor dem Hintergrund zukünftiger Weiterentwicklungen. Eine wei-

tere Nutzerstudie untersucht den Zusammenhang von räumlicher Distanz und Gewichtung der Beziehung zwischen zwei Objekten. Dies ist wichtig, weil die vorgeschlagenen Algorithmen vornehmlich die räumliche Nähe von Objekten nutzen, um Vorschläge für den Anwender nachvollziehbar zu integrieren.

Die Ergebnisse der Nutzerstudien zeigen, dass unter den gegebenen Bedingungen eine messbare Verbesserung der Effizienz erreicht wird. Eine weitergehende Analyse weist darauf hin, dass dieser Effekt tatsächlich auf die räumliche Darstellung zurückzuführen ist. Dies wird durch verschiedene Merkmale deutlich, die während der gesamten Test-Sitzungen beobachtet wurden. Darüber hinaus wurde festgestellt, dass die Studienteilnehmer einen linearen Zusammenhang zwischen räumlicher Nähe und der Gewichtung von Objekt-Beziehungen wahrnehmen und nutzen, um solche Beziehungen auszudrücken.

Um die Praktikabilität der entwickelten Ansätze zu zeigen, wurden einige der Algorithmen in Forschungsprototypen implementiert. Diese entstanden überwiegend in Zusammenarbeit mit Partnern aus der Industrie und decken ein breites Feld an Anwendungsdomänen ab. Spatial Composition, zusammen mit einer Wissensbank und einem Recommender-System, erleichtert die Exploration von Informationen, indem Zusammenhänge für den Anwender bedeutungsvoll dargestellt werden.

Abstract

This work explores the integration of recommendations within spatial hypertext applications. Typically, spatial hypertext involves 2D interfaces where users can add, move, and spatially arrange objects. While hypertext usually refers to linking documents, in the case of spatial hypertext, these links are implicit, created through visual attributes rather than explicit connections. The position and arrangement of objects play a crucial role in this system. Such a system can be enhanced with system-generated recommendations: the user-created spatial structures are interpreted, and a knowledge base then provides recommendations that match the content and structure.

A previously unanswered question is how to present these recommendations effectively. This work focuses on developing, implementing, and evaluating methods for meaningfully embedding recommendations within existing visual structures. The goal is to make it easier for users to understand relationships and, thereby, enable more efficient and effective knowledge exploration. In this context, the term “spatial composition” is introduced, analogous to methods that interpret visual structures (“spatial parsing”).

Five algorithms are developed and described as examples to achieve this goal. Their descriptions are chronological and highlight the problems encountered and their solutions. The algorithms are evaluated and compared based on various characteristics deemed important in the context of spatial composition. A user study examines the impact of integrating recommendations into spatial structures on effectiveness and efficiency, compared to a list-based presentation. In addition, the reasons behind these effects are investigated, considering future developments. Another user study explores the relationship between spatial proximity and the weighting of relationships between two objects. This is crucial because the proposed algorithms primarily use spatial proximity of objects to integrate recommendations in a way that is

understandable for the user.

The results of the user studies demonstrate that, under the given conditions, a measurable improvement in efficiency is achieved. Further analysis indicates that this effect is indeed attributable to spatial composition. This is evident from various features observed throughout the testing sessions. Furthermore, it was found that the study participants perceive and use a linear relationship between spatial proximity and the weighting of object relationships to express such connections.

To demonstrate the practicality of the approaches developed, some of the algorithms have been implemented in research prototypes. These prototypes were primarily developed in collaboration with industry partners and cover a wide range of application domains. Spatial composition, combined with a knowledge base and a recommender system, facilitates the exploration of information by meaningfully presenting relationships to the user.

Publications

Throughout the work on this thesis, research results were continuously published. Sections of this document that explicitly draw from these publications are marked accordingly at their respective locations. The following is a list of the publications in chronological order.

1. Claus Atzenbeck et al. "Revisiting Hypertext Infrastructure". In: *HT 2017 - Proceedings of the 28th ACM Conference on Hypertext and Social Media*. HT '17. New York, NY, USA: ACM, 2017, pp. 35–44. ISBN: 9781450347082. DOI: 10.1145/3078714.3078718 [13]

2. Daniel Roßner and Claus Atzenbeck. "Spatial Hypertext for End-User Development Tools". In: *HUMAN 2018 - Proceedings of the 1st Workshop on Human Factors in Hypertext, Co-located with HT 2018*. New York, New York, USA: ACM Press, 2018, pp. 9–15. ISBN: 9781450356589. DOI: 10.1145/3215611.3215612 [103]

3. Claus Atzenbeck, Daniel Roßner, and Manolis Tzagarakis. "Mother - An Integrated Approach to Hypertext Domains". In: *HT 2018 - Proceedings of the 29th ACM Conference on Hypertext and Social Media*. New York, New York, USA: ACM Press, 2018, pp. 145–149. ISBN: 9781450354271. DOI: 10.1145/3209542.3209570 [12]

4. Daniel Roßner, Claus Atzenbeck, and Tom Gross. "Visualization of the Relevance: Using Physics Simulations for Encoding Context". In: *HT 2019 - Proceedings of the 30th ACM Conference on Hypertext and Social Media*. New York, New York, USA: ACM Press, 2019, pp. 67–76. ISBN: 9781450368858. DOI: 10.1145/3342220.3343659 [107]

5. Susanne Purucker, Claus Atzenbeck, and Daniel Roßner. "Intelligent Hypertext for Video Selection: A Design Approach".

In: *HUMAN 2019 - Proceedings of the 2nd International Workshop on Human Factors in Hypertext.* New York, New York, USA: ACM Press, 2019, pp. 19–26. ISBN: 9781450368995. DOI: 10.1145/3345509.3349279 [98]

6. Daniel Roßner, Claus Atzenbeck, and Daniel Urban. “Weblinks: Augmenting Web Browsers with Enhanced Link Services”. In: *Proceedings of the 3rd Workshop on Human Factors in Hypertext, HUMAN 2020.* New York, NY, USA: Association for Computing Machinery, Inc, Dec. 2020, pp. 1–5. ISBN: 9781450380584. DOI: 10.1145/3406853.3432663 [108]

7. Claus Atzenbeck and Daniel Roßner. “Thoughts Reflection Machine”. In: *Proceedings of the 31st ACM Conference on Hypertext and Social Media, HT 2020.* New York, NY, USA: Association for Computing Machinery, Inc, July 2020, pp. 117–121. ISBN: 9781450370981. DOI: 10.1145/3372923.3404837 [11]

8. Eelco Herder, Daniel Roßner, and Claus Atzenbeck. “Structuring and Exploring User Behavioral Patterns in Social Media Traces”. In: *Mensch und Computer 2020 - Workshopband.* Bonn: Gesellschaft für Informatik e.V., 2020. DOI: 10.18420/muc2020-ws111-343 [59]

9. Eelco Herder, Daniel Roßner, and Claus Atzenbeck. “Hypertext as a Tool for Exploring Personal Data on Social Media”. In: *Proceedings of the 31st ACM Conference on Hypertext and Social Media, HT 2020.* New York, NY, USA: Association for Computing Machinery, Inc, July 2020, pp. 135–136. ISBN: 9781450370981. DOI: 10.1145/3372923.3404831 [58]

10. Claus Atzenbeck, Peter Nürnberg, and Daniel Roßner. “Synthesising Augmentation and Automation”. In: *New Review of Hypermedia and Multimedia* 27.1-2 (Apr. 2021), pp. 177–203. ISSN: 1361-4568. DOI: 10.1080/13614568.2021.1942237 [8]

11. Daniel Roßner, Claus Atzenbeck, and Tom Gross. “Spatial Layout Versus List Layout: A Comparative Study”. In: *Human-Computer-Interaction – INTERACT 2021.* Springer, Cham, Aug. 2021, pp. 495–498. DOI: 10.1007/978-3-030-85607-6_65 [105]

12. E. Herder, D. Roßner, and C. Atzenbeck. “Reflecting on Social Media Behavior by Structuring and Exploring Posts and Comments”. In: *i-com* 19.3 (2021). ISSN: 21966826. DOI: 10.1515/icom-2020-0019 [57]

13. Daniel Roßner and Claus Atzenbeck. “Demonstration of Weblinks: A Rich Linking Layer Over the Web”. In: *Proceedings of the 32st ACM Conference on Hypertext and Social Media.* New York, NY, USA: ACM, Aug. 2021, pp. 283–286. ISBN: 9781450385510. DOI: 10.1145/3465336.3475123 [102]

14. Lisa Eidloth, Daniel Roßner, and Claus Atzenbeck. “User Study on Link-Service Usage and Information Processing in the Context of the World Wide Web”. In: *Proceedings of the 5th Workshop on Human Factors in Hypertext.* New York, NY, USA: ACM, June 2022, pp. 1–9. ISBN: 9781450394017. DOI: 10.1145/3538882.3542802 [42]

15. Daniel Roßner, Claus Atzenbeck, and Tom Gross. “The Effects of Spatial Visualization versus Ranked Lists on Quality, Time Efficiency, and Interaction”. In: *HT 2022: 33rd ACM Conference on Hypertext and Social Media - Co-located with ACM WebSci 2022 and ACM UMAP 2022* 11.22 (June 2022), pp. 132–142. DOI: 10.1145/3511095.3531286 [106]

16. Daniel Roßner, Jae Sook Cheong, and Claus Atzenbeck. “From Maintenance in Industry to Bibliographic Data: Spatial Hypertext as Communication Medium Between User and Machine”. In:

Proceedings of the 5th Workshop on Human Factors in Hypertext, HUMAN 2022 (June 2022). DOI: 10.1145/3538882.3542803 [109]

17. Claus Atzenbeck, Eelco Herder, and Daniel Roßner. "Breaking the Routine: Spatial Hypertext Concepts for Active Decision Making in Recommender Systems". In: *New Review of Hypermedia and Multimedia* 29.1 (Jan. 2023), pp. 1–35. ISSN: 1361-4568. DOI: 10.1080/13614568.2023.2170474 [7]

18. Claus Atzenbeck, Sam Brooker, and Daniel Roßner. "Storytelling Machines". In: *Proceedings of the 6th Workshop on Human Factors in Hypertext.* New York, NY, USA: ACM, Sept. 2023, pp. 1–9. DOI: 10.1145/3603607.3613481 [6]

19. Daniel Roßner, Claus Atzenbeck, and Sam Brooker. "SPORE". in: *Proceedings of the 34th ACM Conference on Hypertext and Social Media.* New York, NY, USA: ACM, Sept. 2023, pp. 1–6. DOI: 10.1145/3603163.3609075 [104]

20. Johannes Wirth, Daniel Roßner, René Peinl, and Claus Atzenbeck. "SPORENLP: A Spatial Recommender System for Scientific Literature". In: *Proceedings of the 19th International Conference on Web Information Systems and Technologies.* SCITEPRESS - Science and Technology Publications, 2023, pp. 429–436. ISBN: 978-989-758-672-9. DOI: 10.5220/0012210400003584 [132]

21. Daniel Roßner, Lisa Eidloth, and Claus Atzenbeck. "PAIRWISE – From Spatial Structure to Knowledge". In: *Proceedings of the 35th ACM Conference on Hypertext and Social Media.* HT '24. Poznan, Poland: Association for Computing Machinery, 2024, pp. 208–216. ISBN: 9798400705953. DOI: 10.1145/3648188.3675137 [110]

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Spatial hypertext, as well as hypertext research in general, resides in a crucial niche aimed at bridging the gap between human associative thinking and the development of adaptive applications. These applications support human cognitive and creative processes by implementing hypertext paradigms that have been researched and developed since as early as the 1940s. In recent years, there has been a transition in the perception of hypertext: from viewing it merely as a system or infrastructure to recognizing it as a method for addressing questions in various research and application domains [9, 4]. Building upon this acknowledgement, this thesis offers an account of contributions to current advancements in the field of spatial hypertext.

Spatial hypertext is a visual abstraction of classic hypertext. It uses computer-generated visual structure to support users in fulfilling tasks that are cognitively demanding or focus on associative/creative thinking. Work in the field of spatial hypertext intersects with a wide range of research areas, primarily within the realms of human-computer interaction (HCI), hypertext, information visualization, and cognitive science. Spatial hypertext originated from efforts to enhance the traditional node-link model by visually representing nodes and links within a 2D space [78]. Today, there is a widespread adoption of this method of visually representing and arranging information in the context of 'preserving' as well as stimulating thought processes (e.g. mind maps, technical diagrams). This also includes commercial tools, such as the visual collaboration platform Miro[1]. Nonetheless, there are still

[1] https://miro.com

unresolved questions regarding the interaction of users with machine-provided intelligence within such visual work spaces. This intelligence can manifest itself in various ways: chat-based retrieval systems (e.g. chatbots), implementations of retrieval augmented generation (RAG) or recommender functionality, to name a few. The focus of this work is on spatial hypertext systems with recommender functionality: Depending on the context of a spatial hypertext, users are provided with recommendations intended to enhance knowledge retrieval and stimulate associative thinking. These recommendations need to be presented in relation to the user's current context. The visualization of this relationship to the (dynamically changing) context is the underlying focus of this thesis.

1.1 Motivation

The notion of using spatial hypertext to visually model knowledge and make it accessible to users (i.e. usable) has been established for several decades [119, 79]. However, solutions for dynamically enriching spatial hypertext with additional information retrieved from sources such as a knowledge base were missing. Additionally, there were no insights into how to visually and spatially integrate recommendations so that this supplementary information is coherent with the user's understanding while simultaneously promoting creative thinking.

Based on these limitations and building upon the expertise of the Visual Analytics Research Group at the Institute for Information Systems (iisys) in the field of spatial hypertext, as well as in close collaboration with the Human-Computer Interaction Group at the University of Bamberg, this work was motivated to advance knowledge in these areas. A vital part of this work was the development of suitable strategies for integrating recommended information within user-generated spatial hypertext. Our focus is on making the relationships between these pieces of information visible to users through spatial arrangement.

As a first practical use case, individuals involved in sustainable development within the building sector were chosen. Sustainability is, especially today, an important topic for many different people. The need to conserve natural resources for future generations and to take a responsible approach to nature is a responsibility that all humans bear. For the construction sector, this necessitates the rapid development of novel processes, technologies, and materials. Keeping up with these developments is a challenge for experts, and even more so for laypeople trying to gather knowledge in this field. This need for information and knowledge was already addressed by König et al. [70], who mainly worked on the issue of data scattered across various web pages and its lack of semantic relation. By drawing upon the fundamentals of the Semantic Web [17], detailed queries supported by reasoning could be made. This approach proved beneficial for individuals who already possess a clear idea of what information they are seeking. Nevertheless, laypeople may require a more implicit approach, as they may not be able to explicitly state, what they are seeking for. A similar argument holds true for experts in the field tasked with creating a database: Tagging links with semantics is a notoriously difficult and elusive task, as shown by Marshall and Shipman [81].

These deliberations led to the conclusion that an application allowing people to visually *explore* an information space is necessary – not only for the use case described above but for any task where the explicit expression of intention is difficult, where it is crucial to preserve a user's knowledge, where the goal is to enhance a user's knowledge on a certain topic, or all of these combined (cf. Chapter 2). These requirements are fitting with a (spatial) hypertext system, as its core concept is the user-centric augmentation of human intellect [43] by the exploratory use of information supported by spatial parsing [46]. In the course of this work, we developed an application that allows users to explore domain-specific knowledge bases. It offers a 2D space where users can arrange information pieces. The visual arrangement is interpreted by the system to provide further recommendations. The application is based on the findings presented in this thesis, as it integrates the recommendations provided in a 'meaningful' way into the user-generated visual structure. Vital components and how they relate to the research objectives are explained throughout this work.

1.2 Research Objectives

This section details the scientific contributions and significance of this work to the research community, outlining the motivation for the thesis and the research gaps identified along the way. The main contribution is presented in the form of three hypotheses, a descriptive comparative analysis of five spatial composition algorithms, and case studies that show the general applicability of the proposed interface. These contributions aim to advance the field of computer-generated structure in spatial hypertext [66, 119]; this will be coined as *spatial composition*, in reference to *spatial parsing* [46, 116].

Spatial hypertext finds its application in a variety of use cases. Some of them can be considered as spatial interfaces, where the part of hypertext is limited to human creation and perception of visual structure. Others incorporate implicit connections of nodes into their system. For example, Tinderbox[2] [19], a note making and organizing tool, uses the structure given to notes, so that autonomous agents work on the structure and query the data. Shipman et al. suggested VKB [119], a hypertext system for "visual knowledge building". It presents the idea of different types of "suggestions" that e.g. should support the user in structuring their space. By employing different spatial parsers that facilitate the shared comprehension of visual structures between machines and users, this acquired knowledge can be used to provide assistance to users. A possibility examined in this work is to provide users with

[2] `http://www.eastgate.com/Tinderbox/`

recommendations from a knowledge base that fit the current context (by content and structure). Such a system leverages exploratory search tasks and visualizes relevant information.

A crucial point for any recommender system is the question of how to visualize the results. Ideally, the visualization not only contains the suggested content but also unveils relationships – the (semantic) network backing the data. This is important because the quality of recommendations has improved significantly in recent years, often with a lack of transparency on why a particular item was suggested and how it relates to user needs [101, 28]. Therefore, it is necessary to examine to what extent a semantic visualization of recommendations in a spatial hypertext system supports users in exploring a knowledge graph.

A recommender system requires knowledge to be available, which is typically stored in a knowledge base. In addition to the aforementioned main contributions, the creation of knowledge bases will be discussed, as they are a fundamental necessity to test and evaluate the system as a whole. During the preparation of this thesis, various research projects were undertaken that integrate parts of the system to be implemented. Different application domains and use cases (cf. Chapter 2) require specialized knowledge bases that influence the system.

First, we will describe the framework of the system. The requirements of various projects form the basis for constructing this framework in which the research questions are posed and explored. Following this, a discussion of the methods designed to address these questions and to advance research in these corresponding areas follows.

1.2.1 System Framework

At the beginning of this thesis, the Visual Analytics research group started with an implementation of the Asgard layer within the Mother hypertext architecture (cf. Section 3.2.3), including a set of various spatial parsers [116], and the application of the results of these parsers to a recommender system. As there is a lack of research in this specific area, the requirements regarding the system's implementation were not given, but had to be investigated. Developing a visualization strategy to display the information was crucial for this. This part will explain the conceptual limits of the intended system and specify key and recurring terms throughout this thesis.

A unique feature of spatial hypertext is its implicit nature and not only the consideration but the incorporation of uncertainty into its mode of operation. All great ideas about the interconnectedness of objects – everything is "intertwingled" as coined by Ted Nelson [36, 91] – are extended by uncertainty. This uncertainty arises from the subjective interpretation of visual relationships between objects by humans or machines. This suggests certain drawbacks, because the simple 'grammar' of spatial hypertext does not allow explicit relations, which complicates expressiveness and a coherent interpretation. Although spatial hypertext is well suited for tasks such as learning, brainstorming, or browsing, where uncertainty is a natural element and will not hinder, but may even be of use for the process. This is because users are not required to express their ideas in a complete and clear way.

Keep it implicit The display of search results or recommendations in general is a wide-ranging field with numerous possible approaches (cf. Section 3.3). Given that the system is designed to utilize the spatial hypertext paradigm, the aim is to present the recommended information as a natural and inherent part of the spatial hypertext. Its relation to the search context should be encoded with the very same graphic variables that human users use to structure the space. That said, every aspect of the visual representation of information within a workspace should stay within the boundaries of the spatial hypertext.

Content-Independence Visual symbols in a spatial hypertext, sometimes referred to as entities, concepts, notes, or nodes, wrap any piece of information. Information may be multimodal and can manifest itself in the form of simple keywords, documents, images, resources (cf. URIs) – simply put *everything*. From the perspective of a visualization approach, this means that wrapped information must not influence the result; only the relations do. Since the *Visual Information Seeking Mantra*[3] does not specify the method of retrieving "details on demand", the visual symbols might partially or fully embody the information, or they might not at all. Consider, for example, a red rectangle displaying a keyword as overlaid text – in this case, the rectangle represents and embodies the information (keyword).

Screen Size Spatial hypertext needs enough screen space to accelerate the user experience. Although the system is not restricted to any exactly defined screen size or input method (keyboard/mouse or touch), it should be designed to work on tablets[4] and ordinary screens[5].

Adaptability In particular, the user interface should be able to adapt to specific application domains. While a spatial hypertext on its own is simply about visual properties of symbols, a user interface might impose additional features to support a certain application domain. For example, a tool that models the data flow from one component to another component needs to represent this data flow e.g. in the form of lines or arrows between these components. Other scenarios may require that visual symbols should depict images or, in general, have a certain size (i.e. pixels).

1.2.2 Research Gap

Visually incorporating recommendations within spatial hypertext intersects primarily with two research domains: at a broader level, it aligns

[3] Coined by Ben Shneiderman as "Overview first, zoom and filter, then details-on-demand." cf. Section 3.2.3
[4] 7 to 12 inches
[5] 12 to 32 inches

with the fields of information visualization and visual querying, which are both related to research in recommender systems. More specifically, it resonates with the concept of "Generative semantic clustering in spatial hypertext", as introduced by Kerne et al. [66] in 2005.

Research in recommender systems focuses predominantly on data and techniques to offer highly personalized recommendations to individual users. Although numerous systems employ visual elements and a 2D space for information querying, they adhere to explicit rules; for example, visual symbols might represent specific query parameters [53]. However, these systems often overlook the use of the spatial/visual structure of the space itself. The VKB system [119], mentioned above, represents the most significant recent effort to integrate spatial structure and to base recommendations on it. Exploring the benefits (and drawbacks) of this approach remains an uncharted area of research, a gap that the theses *T2* and *T3*, discussed in the following section, aim to address.

Furthermore, there is a trend in modern recommender systems to position the user as merely a consumer of a recommendation feed, mainly due to perceived convenience [7]. However, a spatial hypertext interface alters this dynamic, transforming the user into an active decision-maker who controls the direction of 'journey'. The viability of this approach is exemplified in *T5*.

Visualizing recommendations[6] is, in fact, a more general problem of information visualization that requires specialized solutions. For example, in the context of graph visualization, it is important to (efficiently) incorporate thousands of symbols, possibly avoiding/minimizing crossing edges. In the context of augmenting a spatial hypertext with said recommendations, other factors are important: Recommendations must conform to the established structure, and their behavior

[6] In this work, recommendations are represented as a weighted, undirected graph. In this graph, the vertices symbolize the recommendations and concepts of the user, while the weighted edges indicate the level of relatedness between these vertices.

should be consistent with the iterative process found in working with spatial hypertext. With VKB, Shipman et al. [119] introduced the term "Placement Suggestions" to describe a functionality that assigns information pieces a position that augments the structure already created. However, the placement method is relatively simple and static and is not designed for recommendations that change continuously. Consequently, we have pinpointed a research gap in translating knowledge into the most significant graphical variable for encoding relationships, which is spatial proximity. This aspect is investigated through *T1* and further complemented by a comparative analysis of the proposed composition algorithms in *T4*.

1.2.3 Theses

Subsequent theses delineate the primary contributions of this research. They are substantiated through a combination of quantitative data (comparative studies), qualitative evidence (case studies and expert feedback), and observational comparative analysis.

Hypotheses

T1: Relation of Knowledge and Proximity

Thesis 1 asserts that there is a discernible relationship between knowledge, defined as an undirected weighted graph with concepts as vertices, and its visual representation, which is created and interpreted by humans. This representation includes visual symbols and their spatial relationships within a two-dimensional space, emphasizing the interplay between the graph structure of knowledge and its human-centric visual interpretation.

The hypothesis that humans use proximity to signify relationships has been frequently tested, not least through the evaluation of Schedel's implementation of spatial parsing. T1 aims to excel this previous work, as it assumes a relation between *weight* (as used in the knowledge graph) and distance (Euclidean). The proposed hypothesis is evaluated through a user study, in which participants initially contribute to establishing a baseline 'ground truth' in the form of a graph and then replicate a similar task in a spatial context. The findings indicate a linear correlation between the weight of the relationships in the graph and their spatial distance. Furthermore, the results imply that future research could benefit from enhancing the concept of "weight" with a factor representing uncertainty.

T2: Improved Efficiency for Spatial Composition

Integrating recommendations into a spatial hypertext interface that utilizes spatial composition to navigate a knowledge base improves efficiency in exploratory search tasks, compared to using an established ranked list representation.

The baseline was selected due to its widespread use, as evidenced in the work of Klouche et al. [68]. Their approach, similar to the proposed system, employs a spatial environment to facilitate the exploration of recommendations. The results of a user study show that indeed the participants solved an exploratory search task in shorter time, with a similar result (effectiveness) and satisfaction.

T3: Reasoning about Efficiency

The increase in efficiency is due to an improved understanding of the recommendations displayed. Spatial composition, particularly through the physics-based composition in this context, aids in faster identification of pertinent information during every phase of the exploratory search process.

A detailed analysis of the aforementioned user-study allows the creation of a replay of each recorded session. The duration measured between user interactions and documented navigation events in this period implies that participants were actively engaged in the investigation of the space and were more rapid in determining their subsequent actions. The same analysis suggests that there is an upper bound of information units examined by users before they proceed with their next action. In particular, this upper limit is identical for both tested interfaces. This information is therefore helpful to implement composition with this limit in mind, to e.g. improve space utilization.

Analysis

T4: Comparative Analysis of Spatial Composition Algorithms

An important contribution to the research field is the creation and mathematical foundation of five spatial composition algorithms, along with a comparative analysis. This involves two key aspects:

1. Implementation: The development of these algorithms includes a detailed mathematical basis, ensuring a robust and theoretically grounded framework.
2. Comparative Analysis: These algorithms are critically evaluated on several fronts. Firstly, their efficiency in space utilization is examined, assessing how effectively they use the available area in the spatial interface. Second, the manner in which they encode knowledge through the concept of distance is scrutinized. This involves understanding how different levels of relatedness or relevance are spatially represented. Lastly, their ability to handle smooth transitions is a crucial factor. This aspect is important as it directly influences user interactivity, determining how seamlessly users can navigate between different states or sets of recommendations within the spatial interface.

In general, these algorithms are interconnected, each building on the insights and foundations of the others. Their comprehensive assessment not only demonstrates their individual strengths and limitations, but also contributes significantly to the understanding of spatial composition.

T5: Case Studies/Proofs of Concept

While the earlier mentioned studies focused on particular tasks, we propose that a spatial hypertext interface, which integrates recommendations through spatial composition, can be effectively applied in various application domains where exploratory search is a crucial element.

This is a bold assertion and, although it cannot be universally validated, the suggested interface has been used in various research projects in multiple application domains. Feedback from domain experts (or users at large) has been instrumental in refining the system, impacting the composition approach, and generally enhancing our comprehension of key elements in exploratory search and spatial hypertext. The following is a compilation of published work detailing the integration of the proposed interface. A more detailed view is given in Chapter 2.

1. Recommender system supporting development tasks [103]
2. Recommender system for movies in cooperation with a TV manufacturer [98]
3. "Structuring and Exploring User Behavioral Patterns in Social Media Traces" [59, 58]
4. Support for maintenance workers in an industrial setting [109]
5. Recommender system for scientific publications about NLP and AI [109, 132]
6. Recommender system and story breaking tool [104, 6]

1.3 Structure

The following Chapter 2 details various projects that have implemented components of the system which this thesis attempts to describe. This chapter shows the general applicability of the later proposed composition algorithms and should help the reader understand the challenges and opportunities of such a system. Chapter 3 revisits research of related fields and sets the scientific framework for the contribution of the thesis. As spatial composition has close relations with research done in information visualization, a highlight on that topic is given.

Chapter 4 contains a main contribution of the thesis, as it discusses and details the implementation of a set of spatial composition algorithms. Their explanation follows chronological order of development to highlight the *lessons learned* and the complexity of the topic. Next to this core section, the implementation of auxiliary components is shown, namely of the knowledge base that is able to give recommendations based on queries that are derived with help of the structure layer of the system.

Following this, Chapter 5 discusses the user studies and tests conducted. These are crucial for providing empirical evidence to support the thesis's assertions about the relationship between proximity and knowledge construction. In addition, they are instrumental in evaluating the differences between the spatial composition of recommendations and established list representations.

The thesis concludes with Chapter 6 that summarizes the key findings. Still open questions and further research directions are discussed to set the foundation for future research that may advance the topic of spatial composition.

2 Exemplary Application Domains

This chapter provides a detailed overview and analysis of the research prototypes developed during the course of this thesis. These prototypes played an important role in the research process and served as practical applications of the theoretical concepts and methodologies discussed in the following chapters. They provided valuable insights and empirical data that informed the ongoing development of the thesis. The goal is to demonstrate how the conceptional idea of integrating recommender functionality into a spatial hypertext application can augment the way users interact with said applications and support them in the presented application domains.

However, it is necessary to recognize that the prototypes developed in the earlier phases of the research do not include all the features and findings that were later identified and integrated. As research evolved, new understandings and advances were made, leading to the enhancement of prototypes with additional functionalities and improvements. These subsequent developments reflect the progressive nature of the research, where each prototype built on the learnings of its predecessors, progressively refining and expanding the scope of the research. Additionally, the prototypes were specifically designed to meet the requirements of their respective research projects and application domains. Therefore, the recommendation of information within the context of spatial hypertext represents only one component of these more comprehensive software solutions. This approach ensured that each prototype was not only a testbed for theoretical concepts, but also a practical tool tailored to specific domain needs. Consequently, when examining these prototypes, it is important to consider the context in which they were developed and the stage of research they represent.

2.1 End-User Development Tools

> This section builds on work previously published in [103], which has been further extended and adapted for the purposes of this thesis.

As software manages tasks in daily life more and more, the need for user-friendly development processes has grown. Despite advances in usability, the complexity of software development still requires professional skills. However, there is a growing demand among laypeople to modify or create applications, a concept known as *End-User Development* (EUD) [74]. This form of development, distinct from professional programming, is often driven by the need for specialized solutions or the absence of suitable professional tools [94]. Many EUD tools leverage graphical interfaces to minimize the learning curve, aligning programming actions more closely with real-world tasks [44]. Examples include Yahoo! Pipes, VisPro, WebMakeup, and ResEval Mash, each requiring users to locate, arrange, and define the use of various components within a multidimensional workspace [137, 37, 62].

The ODIN project ("Open Data Innovation") aimed to create a platform for sophisticated developers and laypeople (e.g. journalists) alike, who need to work with open data. A typical use case consists of collecting publicly available data, aggregating the said data, and eventually the generation of a visual representation. The core component is the *browser based development environment* (BDE), which essentially acts as a 2D canvas where users can design the data flow of pre-existing

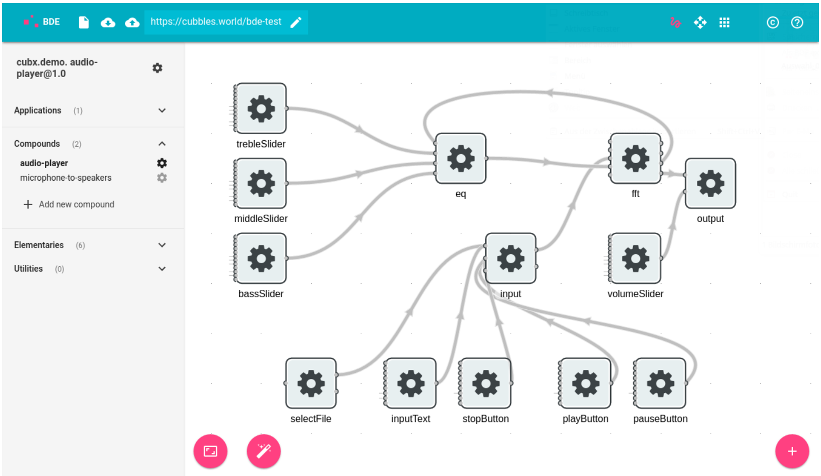


Figure 2.1: BDE workspace with Cubbles modeling a music player

software components that are accessible to all system users. These software components are called *Cubbles*[1] and are based on meme media technology [127]. Cubbles have input and output slots that can be linked through pipes. As illustrated in Figure 2.1, a BDE screenshot demonstrates the graphical construction of a music player using Cubbles. These Cubbles are JavaScript-based Web components, executed in the user's Web browser. The browser fetches all necessary modules from the Cubbles Base, a comprehensive global repository, enhancing their reusability as software components.

After all, the BDE can be seen as a spatial hypertext, where Cubbles ("nodes") are interlinked by visual means (e.g. proximity) and with a navigational structure that builds the data flow from the output to the input slots of the components. This characteristic renders the application suitable for integration with a recommender system, which takes advantage of (visual) structure to offer recommendations. What is the benefit and why is it needed?

[1] `https://cubbles.github.io`

In the context of the application domain, there are hundreds of Cubbles, which are responsible for connecting to various data providers or model data in a way that makes it possible to create meaningful visualizations. Laypeople, in particular, often find it challenging to identify the specific Cubbles required for their projects. When exploring and experimenting with open data, they may initially be unaware of their precise needs. This lack of clarity typically persists until they gradually refine and fulfill their information requirements through iterative exploration. Therefore, the system permanently analyzes the structure and user interactions to recommend other Cubbles, which may be helpful or interesting in the current context. In this way, end-users are not forced to explore the Cubbles Base by hand but can work with the recommended modules. The ODIN project does not seamlessly incorporate recommended information into the spatial hypertext or the BDE in this context. Instead, it displays recommended Cubbles in a list adjacent to the spatial workspace.

Throughout the project's duration, we gathered extensive feedback from project partners and test users. They all reached a consensus that a recommendation system, which evaluates users' actions on the workspace, offers significant benefits compared to traditional text-based search methods. This perspective was shared even by experienced developers of the Cubbles platform, who often encountered difficulties in locating the necessary software modules. Further learning was that displaying recommendations on a side panel makes it difficult to identify where a recommended Cubble would fit in the current context. This challenge served as a key motivation to pursue the integration of recommendations directly within the spatial workspace.

2.2 Recommender System for Movies

This section builds on work previously published in [98, 107], which has been further extended and adapted for the purposes of this thesis.

The DemoMedia project, developed collaboratively with Loewe Technologies GmbH – a company specializing in consumer electronics – is an embodiment of the proposed system in the domain of movie (and actor) recommender systems. Its objective is to create an application that allows users to organize and search for movies, actors, and other elements related to films on tablet devices. A 2D space facilitates this organization, enhanced by the system's recommendations. Organization involves adding, moving, resizing, and deleting information entities. In this demonstrator, users input text to create such entities, represented as rectangles within the visual space. When the system identifies additional information related to a typed keyword, such as an actor's name, users have the option to change the display format from mere text to various media formats offered by the application. Likewise, recommendations appear as rectangular objects that contain information about movies and actors.

The foundational data and knowledge base are aggregated and processed from

1. IMDb (Internet Movie Database),
2. a television program dump provided by Gracenote[2], and
3. YouTube.

Figure 2.2 presents a screenshot of the application, displayed on an Android-operated tablet device. The three larger rectangles represent entities that were added by the user, either because he or she searched

[2] https://www.gracenote.com

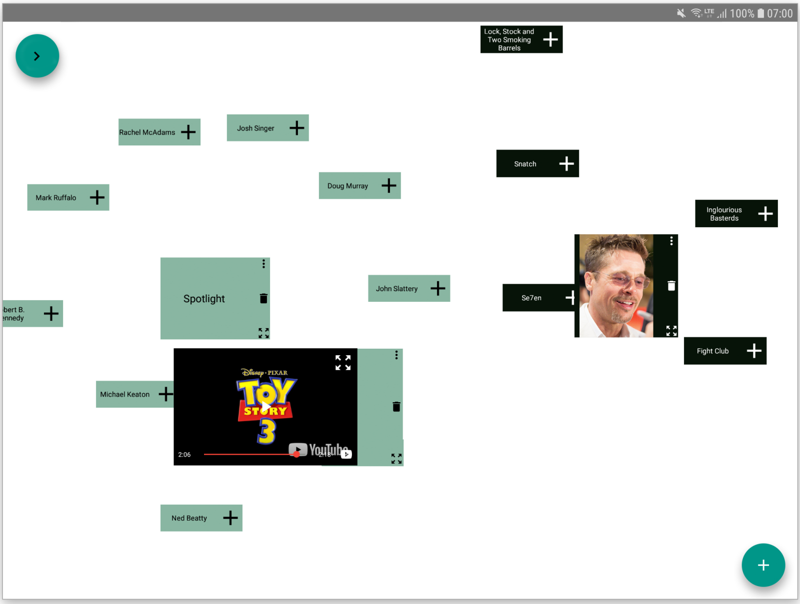


Photo of Brad Pitt extracted and cropped from `https://www.flickr.com/photos/31029865@N06/38338138066/` – by Dick Thomas Johnson, licensed under CC BY 2.0: `https://creativecommons.org/licenses/by/2.0/`

Figure 2.2: Shows three nodes: Spotlight, Toy Story 3 and Brad Pitt – colors of the nodes indicate visual groups as identified by the spatial parsers

for them with an ordinary text-based search or because they were recommended and accepted by the user. The recommended information is depicted as smaller rectangles, each marked with a "+" symbol, indicating that they can be added to the space. Users do not select colors; instead, colors are utilized to represent the current interpretation of the spatial parser. Large rectangles of identical color are perceived as constituents of the same group, and the recommendations associated with this group also bear the same color. The composition of the recommendations is controlled by an algorithm based on a physics metaphor with damped springs connecting the objects; it is detailed in Section 4.2.5.

The user and the system engage in an iterative dialogue that leads to the formation, modification, or dissolution of structures. Every user interaction, be it repositioning entities or introducing new ones, dynamically alters the spatial layout of the recommendations. This process hides the complexity of traditional query languages like SQL or hidden filters in drop-down menus and sliders. Users are freed from the complexities of fine-tuning their queries. With capabilities to pan and zoom within the space, users can develop multiple, visually distinct queries in separate areas, which can later be converged when needed. The system also provides tools to adjust the number of visible recommendations and to filter specifically for movies or actors.

For users seeking detailed information, the content within the nodes is customizable. Options include displaying a Wiki page, a slideshow, or even the movie itself, which can be streamed to a TV. Interaction with this content occurs within the workspace, ensuring seamless integration. Objects can be resized or even expanded to full screen, allowing for a fluid transition between searching and consuming content. For example, a user might listen to the theme music of a movie while rearranging the workspace nodes or reading reviews.

In summary, incorporating recommendations into the spatial hypertext of the system provides users with a more intuitive, contextually appropriate, and visually cohesive means of interacting with and deriving value from these recommendations. In this context, as with other use cases, exploration is a crucial component of the process. The aim is not to merely present users with a limited selection of matching movies but to stimulate further exploration, encouraged by serendipity.

2.3 Reflecting Social Media Behavior

> This section builds on work previously published in [58, 59, 57], which has been further adapted for the purposes of this thesis.

Motivation to take advantage of the user interface and infrastructure of the DemoMedia project, as outlined in the previous section, stemmed from the – back then – recent implementation of the GDPR regulations. These regulations enable users to obtain their data from e.g. social media providers in a structured format, as denoted in article 20, the "Right to data portability"[3], which states that:

> The data subject shall have the right to receive the personal data concerning him or her, which he or she has provided to a controller, in a structured, commonly used and machine-readable format [...]

The application was initially designed for exploring and discovering information, a task that might appear unusual when applied to one's own familiar social media posts and data. After all, the user is both the owner and the creator of this data. The primary goal is to utilize the application's recommendation and visualization capabilities to reveal the connection between social media posts and keywords, thus allowing users to reflect on their activity. A motivation to do so might be the intention to identify usage patterns, such as triggers that lead to regretful social media behavior.

In a widely referenced study on Facebook regrets [130], participants cited various reasons for their regrets: sharing sensitive information, particularly regarding the use of alcohol and illegal drugs, religious and political opinions, or personal and family matters; making negative or

[3] https://gdpr-info.eu/art-20-gdpr/

offensive remarks, or disclosing lies and secrets. The motivations behind these posts and comments often stemmed from a belief in their humor, a need to vent frustrations, or well-meaning intentions that backfired. Typically, these posts were regretted after unexpected outcomes or reactions occurred.

To facilitate this task, we extracted posts and relevant keywords, correlating them using various metrics such as temporal association (date and time of day), content similarity, and sentiment analysis, among others. Figure 2.3 shows an exemplary screenshot of the application. The system underwent an exploratory evaluation, as documented in [57], which revealed that a majority of participants encountered posts they had forgotten about and were surprised to have shared. Interestingly, although these posts were not necessarily sources of regret. This discovery was additionally supported by a subsequent survey. The evaluation indicated that the tool prompted participants to actively reflect on their social media posting behaviors, leading to increased awareness and introspection regarding their online activities.

2.4 Maintenance in Industry

> This section builds on work previously published in [109], which has been further adapted for the purposes of this thesis.

In an industrial context, maintenance generally involves the upkeep of complex machinery in extensive operational settings. This task is critical and can be divided into different categories, with predictive maintenance emerging as a particularly prominent category in recent years. The growing interest in predictive maintenance is attributed to its benefits in preventing failures and reducing downtime in production lines. Enhanced sensor technology and the advent of machine learning

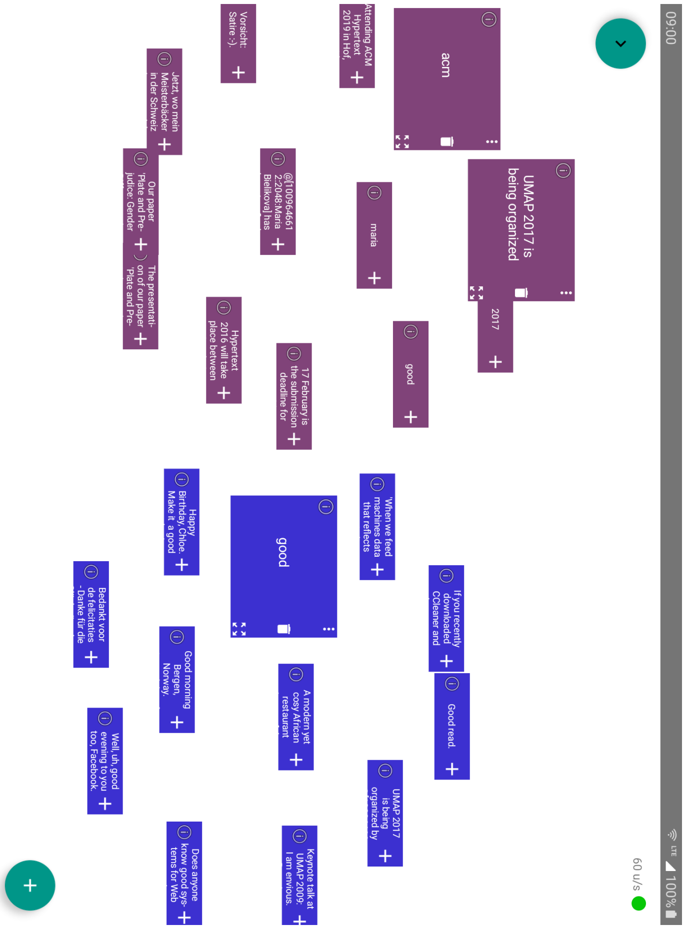


Figure 2.3: A prototype screenshot showing the addition of one post and two keywords

have increasingly shifted the focus from time-based to condition-based maintenance approaches. These and more approaches are classified by the European standard EN13306:2001 [25].

On the contrary, "corrective maintenance" relies significantly on the expertise of maintenance professionals, particularly in diagnosing and resolving problems. However, there is a lack of access to this vital human knowledge, as it requires explicit and time-consuming input methods such as "experience factory concept" or expert feedback. Consequently, in practice, systems supporting this kind of maintenance capture only a small fraction of the employee's valuable experiential knowledge.

The IWInxt project led to the development of a demonstrator in cooperation with two industry partners which provided data, domain knowledge, and feedback. The driving idea was to make already existing explicit knowledge (like documentations or user manuals) easily accessible within a spatial hypertext interface; similar to what was already developed for the movie domain, mentioned in the previous sections. But as explained, the expertise and knowledge of professionals is the crucial resource in this process, therefore, we devised a maintenance workflow that involved the documentation of the maintenance process within the very same spatial hypertext interface. The process of documenting issues, capturing images, tagging responsible colleagues, and more could be streamlined through the creation of simple visual nodes. These nodes, termed "Maintities" – a portmanteau of 'maintenance' and 'entity' – can be organized in a free-form manner, similar to a pin board. This approach is in line with the core principles of spatial hypertext, allowing for flexible and intuitive organization without rigid rules.

Then, an algorithm analyzes both the content and layout of the Maintities, applying spatial parsing to deduce their visual arrangement. This information is used to augment and refine a knowledge base. As a result, maintenance workers using this system can indirectly share their

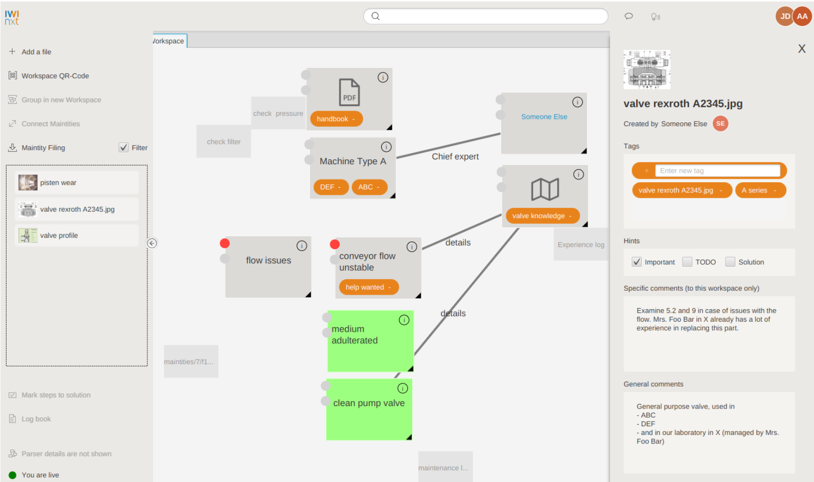


Figure 2.4: Workspace view of the IWInxt desktop client with an entity filing on the left, 2D space in the middle and a detail view for a selected entity on the right

expertise with colleagues. This is achieved by providing recommendations to other users through the same interface while they engage in corrective maintenance tasks, thereby facilitating knowledge transfer. A screenshot of the system is shown in Figure 2.4.

The novel aspect of this demonstrator lies in the use of spatial hypertext as a communication medium between the user and the machine. This is achieved through spatial parsers that translate human-arranged structures into machine-interpretable formats. Additionally, spatial composition, a key focus of this thesis, serves to convert knowledge data into a structured, interactive visual representation. Essentially, it facilitates a two-way translation process: from human input to machine understanding and from machine insights back to a user-friendly visual composition.

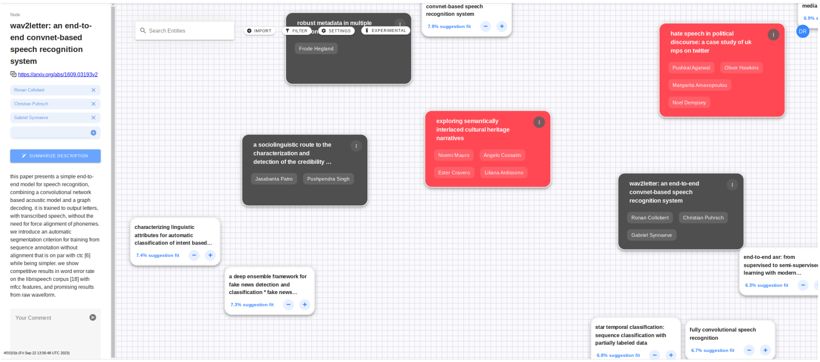


Figure 2.5: SPORE user interface for publication exploration

2.5 Further Application Domains

The most recent advancement of the projects mentioned earlier has been the creation of a versatile framework for knowledge management and exploration. This development has culminated in a web application called *SPORE* (Spatial-Oriented Recommendation System). SPORE hosts various applications, including one for researchers aimed at exploring publications, as illustrated in cf. Figure 2.5 [109, 132], and another focused on story-breaking, particularly in the context of fairy tales, shown in cf. Figure 2.6 [6, 104]. This framework represents a significant step in facilitating efficient and intuitive knowledge discovery and management in diverse domains.

The integration of spatial parsing, recommender functionality, and spatial composition has proven to be a promising strategy in application domains where human expertise and creativity are important. This approach utilizes a spatial hypertext interface for implicit human-machine communication. Although it is efficient for exploratory purposes, it may not be the most effective method for users who have a precise idea of what they are looking for. In general, it is more suited to application domains where exploration is prioritized over direct finding. All the application domains discussed share a common characteristic:

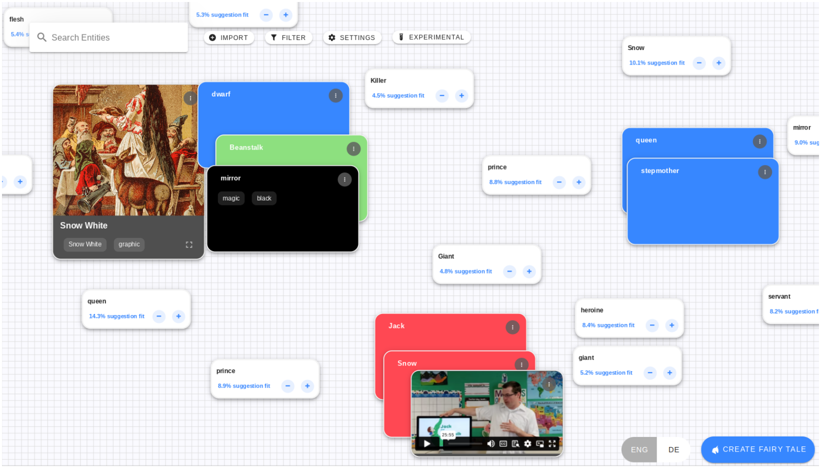


Figure 2.6: SPORE user interface for story-breaking

they are geared towards exploration. Users engage with these systems not only to discover new information, but also to contribute their insights. They do this by spatially organizing information, thus enriching the system with their perspectives and knowledge. This process of mutual information exchange, in which users learn from the system while simultaneously enhancing it with their input, is a key element of these domains.

3 Research Background

> This chapter collects relevant literature and short historical overviews of research done in related fields. Parts of this chapter are based on already published related work sections of the author, in an updated and extended version. These publications are [107, 103, 105, 106].

3.1 Hypertext

Vannevar Bush pioneered the idea of hypertext in 1945: In his today well-known article "As we may think" [22] Bush described the need for systems that facilitate the rapid access, organization, and synthesis of large amounts of information. His envisioned machine with the name "Memex" (Memory + Expansion), utilized microfilm to store documents, and allows to link information in a non-linear fashion. This vision was ahead of its time, but lays the foundation for how we access and trace information today. The impact of Bush's essay is still visible today: During his keynote at the *30th ACM Conference on Hypertext and Social Media* in 2019 [35], Andries van Dam made a compelling plea to the attendees that "everybody needs to read it and re-read it about once a year before going to a [hypertext] conference."[1]

In his 1962 report, Douglas Engelbart described a conceptual framework that was "taking a new and systematic approach to improving the intellectual effectiveness of the individual human being" [43]. Engelbart

[1] See keynote video at `https://www.youtube.com/watch?v=g0yx-F1FGnc&t=1100`

saw great promise, especially in computer systems that support tasks that are performed more quickly and with better comprehension. He also argued that systems support the acquisition of a useful degree of comprehension in complex situations and the ability to acquire better solutions faster and "the possibility of finding solutions to problems that before seemed insoluble" [43].

The term *hypertext* was first mentioned by Theodor Holm ("Ted") Nelson: He described it as "written or pictorial material [that is] interconnected in such a complex way that it could not conveniently be presented or represented on paper" [90]. For Nelson, the prefix *hyper* points to the inability to transform these *objects* into linear media of any kind. Thus, one can say that hypertext is *more* than text, since there is an additional emphasis on its *structure*.

Hypertext has a close relation to *associative thinking*, a cognitive process in which ideas, images, and words are freely linked to one another in a non-linear or non-sequential manner. Hypertext mirrors this process: Just as associative thinking allows the mind to jump from one idea to a related one, hypertext allows users to navigate through information in a similar manner. This makes hypertext a powerful tool for representing and navigating complex networks of related ideas, much like the networks created by associative thinking. Furthermore, hypertext can facilitate associative thinking by providing a structure for exploring and connecting information in a non-linear way, because users can follow their own path through the information, making connections between different pieces of information as they go.

Over the past decades, the hypertext research community has raised, worked on, and solved various questions. Hypertext, as a term, was usually used to describe a certain type of system, but can also serve as a perspective to look at arbitrary systems [9, 4]. Researchers observed that the notion of *structure* can be articulated in a multitude of ways, such as by applying navigational traces or the delineation of hierarchies. Other researchers have worked on the development of infrastructure that accommodates these types of structure and enhances existing applications to incorporate hypertext features.

Today, the *World Wide Web* (W3 or WWW) is a constantly present system in the everyday life of almost everyone. Its foundation was developed in 1989 by Tim Berners-Lee due to the need to efficiently organize and share information between scientists and universities around the world [18]. The WWW is one of many hypertext systems, which aims at solving this and similar issues. Its success is often attributed to its simplicity and adaptability [e.g. 17], which have been proven in the last decades. The WWW may be criticized for its lack of supporting other structure types than a simplistic implementation of navigational hypertext structure [54, 14].

This work promotes the application of spatial structure to enhance associative thinking within the framework of a recommender system and the visual arrangement of information in space. The foundation of this paper is rooted in past hypertext research, which provides the context for the arguments presented. The overarching objective remains to augment human intellect [43] and facilitate knowledge work.

3.2 Spatial Hypertext

3.2.1 History and Motivation

This work builds upon prior research conducted in the field of spatial hypertext systems, where the integration of spatial parsers and antecedent systems occurred within the hypertext community during the early 1990s. Noteworthy systems such as Aquanet [79], VIKI [80], VKB [118], and CAOS [99] introduced distinct structural mechanisms that diverged from the conventional nodes and links typically associated with hypertext, commonly referred to as navigational hypertext.

In his reflections on NoteCards [55, 56] 1988, Frank Halasz criticized the explicit nature of navigational hypertext systems and formulated a vision for the next generation of hypermedia systems: "The static nature of hypermedia networks could be largely eliminated (when appropriate) by including in the hypertext model a notion of *virtual or dynamically-determined structures*".

The conceptual model describes nodes as visual symbols that serve as wrappers of documents or any other entity; therefore, they have visual properties such as color, location, and shape [86]. Spatial hypertext is based on implicit, visually created structures, an approach to address the above-mentioned determined structures. It follows a "cards on a table metaphor" [116], where links (or better: associations) are expressed by all forms of visual attributes. One of the most significant factors is proximity, which serves as a powerful visual indicator to represent relatedness in a two-dimensional (2D) space [23]. However, beyond proximity, there exists an extensive domain of data visualization literacy [20] that encompasses techniques for encoding quantitative aspects (such as the strength of a relation, as in a weighted graph) and qualitative aspects (such as the nature of the relation). More details can be found in Section 3.3.1).

The inherent characteristics of these associations include fuzziness, fragility, and ambivalence. As a result, the interpretation of implicit associations and their perceived strengths can vary among individuals. This variability poses challenges when it comes to automatically arranging nodes within a spatial hypertext.

3.2.2 Spatial Parser

Spatial parsing is a computational technique that focuses on analyzing and understanding the spatial organization and structure of information present in a spatial hypertext. It aims to extract meaningful representations and relationships from visual data, enabling machines to comprehend and interpret visual scenes in a manner similar to human perception. Unlike traditional parsing methods that focus primarily on linguistic or syntactic analysis, spatial parsing considers the spatial arrangements, layouts, and contextual cues present in visual data. Therefore, the term “parsing” is, although well established, misleading. Spatial parsing enables the machine to *interpret* a visual language with a certain syntax [50]. This syntax is mostly based on the mentioned visual cues, but is also highly bound on the individual creating the structure. Its emerging characteristic cannot be covered by a formal grammar. Instead, spatial parsers use heuristics to uncover *some* of the intended structure. Notable implementations can be found in VIKI, VKB, CAOS (cf. Section 3.2.1) and VITE [61].

3.2.3 Mother

Mother [12] is a component-based open hypermedia system (CB-OHS) and, as such, builds an architectural framework for various structure services, such as navigational (node-link), hierarchical, and spatial hypertext. It is used as a basis for the components implemented in this work and offers spatial parsers, which encompass a temporal vision of the space, in addition to traditional visual cues [116]. Furthermore, it allows the implementation of clients that integrate multiple structure services.

Mother has a 3-layered architecture, whose names are derived from norse mythology:

Midgard includes all application components, mainly the user interface and auxiliary modules,

Asgard holds (partly intelligent) components that deal with structures, such as a spatial structure services with parsers or hierarchy aware services; and

Hel hosts a collection of knowledge-centric components, e.g., knowledge bases or intelligent software that is used to populate those.

Originally, the components of these layers communicated with each other through a proprietary binary protocol. The protocol was tuned to have a low footprint, in terms of size and delay, but was difficult to use and implement in the Web environment, as browsers and servers apply many rules and restrictions to improve security. Thus, the layers define a clean API that allows communication with any- or no protocol (it can be linked during the compile time). Figure 3.1 depicts an exemplary setup within the architectural frame of Mother. The most important characteristic is the independence of the layers and components within a layer. A challenge of this and other similar frameworks is the communication among those components. Mother itself is agnostic to the communication layer used; its current API is influenced by the extensive work done in this area, e.g. the Dexter model [54] and others [108, 99, 86].

3.3 Information Visualization

In their well-noticed *Readings in Information Visualization: Using Vision to Think*, Card et al. defined visualization as

> [...] the use of computer-supported, interactive, visual representations of data to amplify cognition. [23]

What does this mean? Cognition, derived from the Latin term *cognoscere* (to know, recognize, or experience), refers to conscious and unconscious processes by which human perception of sensory input manifests itself as thoughts or knowledge. The primary objective of information visualization is to facilitate the interpretation and analysis of data by visually encoding them. It is important to differentiate between "data" as described by Card et al., and "information". Data consist of raw, unprocessed values, whereas information represents the meaningful insights derived from these values. For example, visually

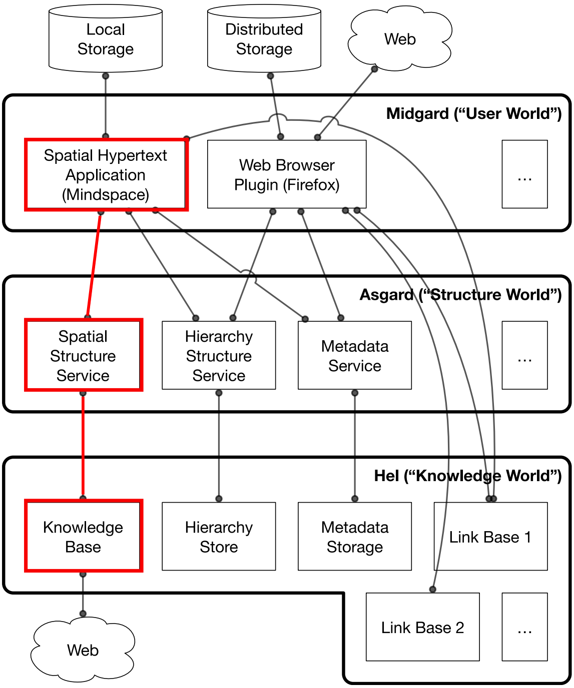


Figure 3.1: Mothers architecture with a highlighted exemplary setup

displaying population ages in a bar chart enables humans to easily categorize and understand these data as information. When machines preprocess data – for example, by identifying all individuals aged 30 based on a user's interest – they minimize the need for further human cognitive effort. Nevertheless, visualization remains essential not only to make these processed data or information accessible to human observers, but also as a foundation for further reasoning and exploration. Users can utilize visualizations to probe a variety of interconnected questions, prioritizing them based on relevance and significance. To facilitate the process of knowledge gain by reasoning on data, a system must provide suitable and effective ways to visualize and represent data under consideration of a user's various needs.

Card et al. addresses this discourse by drawing a distinction between scientific data, which are intended to be grounded in the physical realm, and abstract data, which lack a natural physical counterpart. The latter category, e.g. consider financial data, necessitates the use of abstract visualizations (as exemplified by the population age example).

How does information visualization amplify cognition? In [23, cf. Table 1.3], six different ways are highlighted:

Increased Resources: A lot of cognitive work can be off-loaded to the perceptual system. Users benefit from expanded working memory and the ability to process some visuals in parallel.

Reduced Search: Locality and density of information help users identify what is relevant.

Enhanced Recognition of Patterns: Organization and abstraction of data eases the interpretation and identification of underlying patterns within the data.

Perceptual Inference: Human perceptual abilities and sensory experience enable effective inference of information from visualizations.

Perceptual Monitoring: As in the prior point, the human perceptual system is well trained, and given the threats our ancestors faced, it is very good at detecting changes in our (visual) environment.

Manipulable Medium: Interaction with the visualization helps users refine their needs.

Although the term "computer-supported" does not require additional elaboration, it is essential to emphasize the importance of the term "interactive". The representation component, originating from the field of computer graphics, focuses on the process of mapping data to a visual representation and how that representation is rendered on the display. However, the interaction component refers to the communication and exchange between the user and the system, as the user explores the data set to discover meaningful insights. The roots of the interaction component can be traced back to the realm of human-computer interaction (HCI). Despite being discussed as distinct components, representation and interaction are not mutually exclusive. In fact, interaction with a system can trigger alterations in the representation itself, highlighting the interconnectedness between these two aspects [133].

Shneiderman depicted the "Visual Information Seeking Mantra" [121] as

> Overview first, zoom and filter, then details-on-demand.
> Overview first, zoom and filter, then details-on-demand.
> Overview first, zoom and filter, then details-on-demand.
> And so on.

This mantra illustrates the close relation between interaction and information visualization. Furthermore, any interaction with the system contributes to the context, which may be utilized to refine the resulting information and/or its visualization. Consequently, fostering interaction becomes crucial, as it serves as a valuable resource that helps to fulfill the users' needs.

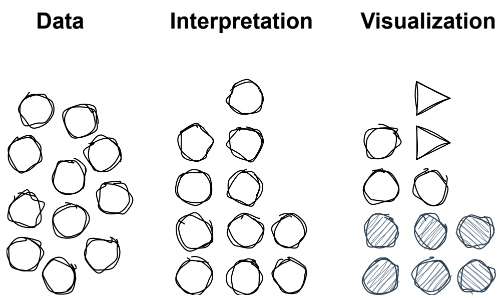


Figure 3.2: Transformation from data to information by automated and visual interpretation

Last but not least, we take a closer look at the term "visual representation": Representation means that the system has to translate information into visual coding. Visual coding is achieved by a rich set of visual cues, which are described in Section 3.3.1. The essential part is to identify a combination of cues that transport the information as clear as possible to the user, since the goal is to amplify cognition.

In Figure 3.2 a sketch depicts the various states of illustrative data units and their interpretation. First, raw and unorganized units; second, organized structure by grouping the data (done by the machine); and third, added visual cues to transport information. The described figure itself is an information visualization: The depicted "data-units" are *represented* in a way to visually support the written explanation. This example demonstrates the interconnected nature of interpretation and visualization, highlighting their intertwined state.

A machine-generated interpretation has to be visualized[2] and any visualization will be interpreted by humans in varying ways. Human interpretation is a subjective task based on expertise, experience, abilities, and other factors. Hence, the wide range of requirements for a 'perfect' visualization may not be addressed by any kind of visualiza-

[2]In order to maintain focus, this work excludes consideration of other human sensory systems, such as hearing or tactile sensations or the implications of cultural background.

tion in total. Therefore, the application of information visualization must focus on basic elements that facilitate a universal understanding of the information that is supposed to be transported by its visual representation.

3.3.1 Visualization Literacy

Visualization literacy describes the ability to read and possibly create visual representations of information. This literacy is based on a set of visual codings that are comparable to the vocabulary of an ordinary written language. These *visual variables* are applied to *graphic symbols* [20] – their mutual combinations are used to transport any kind of information. Interaction with the visualization (zoom, locate, filter, details on demand, history, etc.) and mutation of those variables complete the information visualization.

Depending on the kind of information, we can define the variables shown in Figure 3.3, applied on a simple circle-shaped object (except for the shape variable itself).

3.3.2 Information Visualization and Spatial Hypertext

Both areas aim to make information accessible to humans through visual means. But while the research focus of information visualization lies in computer-generated visuals, which are short-lived and consumed by users, spatial hypertext is usually "hand crafted", long-lived and emergent. Little research was done in bridging both fields: Lyon et al. [75] proposed to add common information visualization techniques, to offer users an additional view of their created spatial structure. Especially if many symbols are part of a workspace, typical *overview, zoom, and filter* tasks are difficult to achieve. These second views would allow one to transform the human-generated structure to suitable counterparts, e.g. by utilizing layers, tree maps, or force-directed layouts.

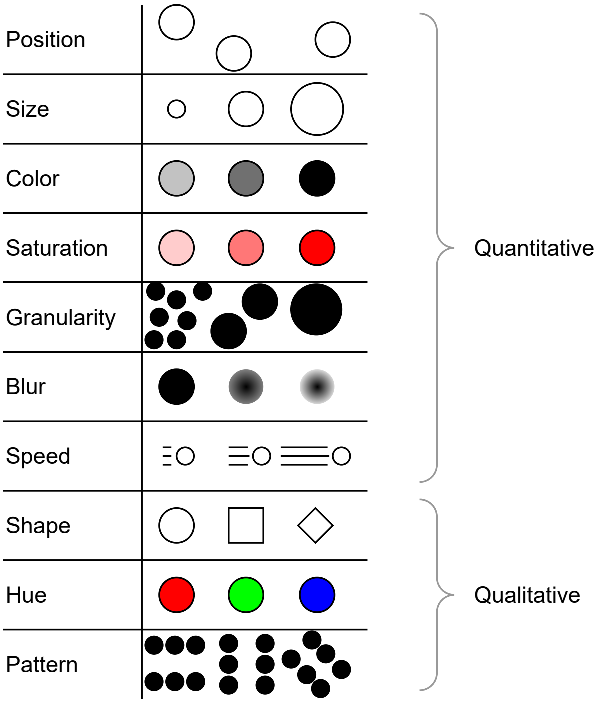


Figure 3.3: Visual variables

In the context of recommender systems, there are a variety of proposed systems incorporating elements from both domains: Computer-generated visualizations represent the results of queries, along with interfaces based on spatial metaphors for executing those queries. The following sections give a brief overview of these systems and categorize them with respect to their relationship to information visualization, spatial hypertext, and recommender systems. Although they may vary in approach and objectives, these prior systems serve as the foundation for the ideas presented in this work.

The field of recommender systems (and information retrieval, in general) has produced great research on how to generate suggestions for products and services for users who do not need to be experts in the respective domain [100]. Here, it has been shown that, in addition to the algorithmic quality that leads to adequate suggestions, models for presenting and explaining the recommendations are essential [16].

In the following, it is differentiated between systems which focus on visual querying, spatial representation of the results, and those which incorporate both.

Visual Querying

In the context of this work *visual querying* describes the process in which users work with a spatial interface to manipulate graphics. These manipulations aim at assisting the users in formulating queries towards an information or recommender system.

This approach is helpful when the creation of a query is rather complex and/or the users are not experts of the (often) specialized query language. Consider OWL[3] as an example. It is extremely powerful and has been developed to query linked data of all kinds. Haag et al. proposed *QueryVOWL* [53], a query extension for VOWL, a visual representation of OWL ontologies. VOWL, describes a common set of

[3] Web Ontology Language

graphical primitives[4], e.g. circles represent a class of an ontology, rectangular property labels and data types, and so on. QueryOWL reuses these primitives and enables users to build their query visually in a 2D space. When this is done, the VOWL is mapped to SPARQL. Although this approach does not cover all features of OWL, a user study demonstrated its usefulness for laypeople.

Targeting non-experts is mostly the motivation for such interfaces. Translating graphical representations to an underlying query language is the often reapplied approach, originating in the late 1980s (e.g. QBD – Query by Diagram [2]). Moreover, visual querying interfaces can also depict query-related information, such as the data flow from the source through filtering, merging, and grouping processes to the final result.

Another classic work was introduced by Anick et al. [3]: The *AI STARS interface* encoded terms of a Boolean information retrieval system on a 2D canvas. The original query was expressed in natural language, which was then visualized with the search aspects arranged in lines and columns, following the conventional left-to-right and top-to-bottom reading order. Lines and columns represent the operators *AND* and *OR*, respectively. Users could reorganize the aspects of the Boolean query to refine the results, because the system was interpreting the structure. This could be seen as a simplistic approach to what was later described as spatial parsing (cf. Section 3.2.2). Other solutions employed the metaphor of "movable filters", implemented as a lens [45] to formulate queries, or used space to allow users to organize filters and flows [134].

In a similar vein, Russel et al. [111] introduced a system called *2dsearch*[5]. It offers a 2D canvas and containers with an *AND*/*OR*-semantic, which can be populated with search terms. Users are free to structure their search as they wish. While there is no meaning for the machine regarding the (relative) positions of the containers and terms,

[4] Reference: `https://service.tib.eu/webvowl/`

[5] `https://app.2dsearch.com/`

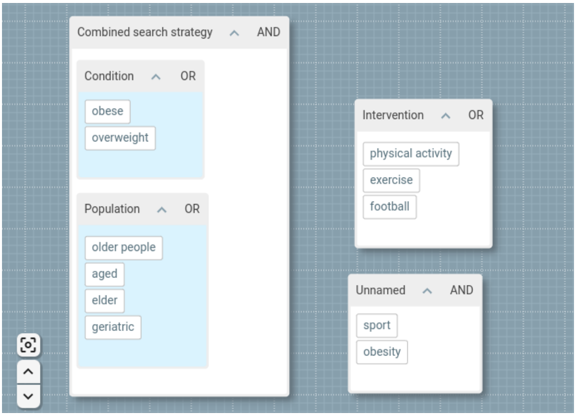


Figure 3.4: Screenshot taken from [1] – translates to the query "((obese OR overweight) AND ("older people" OR aged OR elder OR geriatric)) AND ("physical activity" OR exercise OR football) AND (sport AND obesity)"

this helps users to organize their query and trace its iteration over time.

A recent study by Svarre et al. [126] compared form-based search interfaces, which use multiple, possibly nested search boxes combined by rules (and, or, xor, etc.), with visual query builders. The 2dsearch interface was one of the systems considered. Their main findings were as follows.

1. Participants using a visual interface integrated more facets in their queries and used the interface to structure their search, rather than using pen and paper, which was the approach of many other participants. Furthermore, the authors attributed the users of the visual interfaces a more "holistic view" of the domain, compared to other participants who applied a "reductionist" approach on the form-based queries.
2. The user experience related metrics collected during the study were (significantly) rated higher than those of the form-based interface. The authors acknowledged the limited generalizability of their findings due to the examination of a small set of interfaces.

Additionally, they acknowledged that familiarity with a specific type of interface may also influence this finding.

3. The results revealed that people familiar with form-based approaches appreciated the user experience offered by the visual approach. However, they also relied on the familiarity of conventional interfaces, where the history mechanism provided explicit visibility into past search queries and their specifics.

Although these systems are not rooted in spatial hypertext research, they still build upon the same foundation. They allow their users to utilize space for organization. In addition to query derivation, this is a central characteristic.

Spatial Representation

When a recommender system generates suggestions or when a user submits a query to an information retrieval system, it is imperative to visually depict the results. In this context, the term "Spatial Representation" refers to the utilization of *position* (cf. Figure 3.3) as a fundamental attribute. Retrieved information is allocated a position that can convey its relevance or relationship to other pieces of information within the space. Additional graphical variables may support the *visualization*.

Consider Bead (1992) [27] as a basic example of this approach. Its goal is to display bibliographic data in such a way that related documents appear close to each other. As relatedness may be measured by many variables, it becomes challenging to represent this complexity within a limited visual environment. This process of representing high-dimensional data (and its distance) in a Cartesian space is called *multidimensional scaling* (MDS).

The authors of Bead decided to use a 3D space and to represent individual data points as particles, which means that they do not have a spatial extent. MDS is usually achieved by defining a loss function, which calculates pairwise (dis-)similarities and position configurations to a real number. By minimizing the result of this function, MDS aims to find a configuration, where the distances between all the data points – mostly in 2D or 3D Cartesian space – map "well" to the real-world data.

Bead's solution is based on the physical metaphor of damped springs which apply forces of attraction and repulsion to the particles. Following the principle of minimum total potential energy, the system of particles strives to reach a state of minimal potential energy. This implies that all simulated springs in the system attempt to approach their 'preferred distance' as closely as possible, considering the limitations imposed by MDS in general. Due to restrictions on computation power, the authors of Bead chose to implement a clever trick: They defined so-called "metaparticles", which represent a cluster of closely related atomic particles, or other metaparticles, so that the entire data are structured in a tree. A metaparticle has a mass that is equal to the sum of all enclosed particles, and the center of mass is calculated accordingly. From here, the physics simulation needs to process far fewer force computations per particle. Although this approach was chosen to optimize the complexity of computation, it can be seen with a hypertext lens: A (meaningful) structure is applied, considering that closely related particles should be displayed in a cluster, since the human perceptual system will recognize this.

Therefore, the relaxed position of a spring is derived by document distances, which means that a user can use his or her perception of proximity to recognize strongly related particles. Even if not utilized in this system, the physics-based simulation allows for real-time calculation of positions while a user is manipulating particles. After the simulated annealing process leads to a settled system, users are free to interact with the interface. Zooming (focus), moving (context switch) and selecting particles to retrieve the document (details-on-demand).

Visual Querying and Representation

This category encompasses systems that employ spatial representations to depict the information space or recommended information pieces while simultaneously enabling users to define and refine their information needs within that space.

While Bead is an example of a system that populates a non-restricted space that cannot be altered, Olsen et al. introduced a system to visualize documents in a user-defined 2D space, called VIBE (visualization by example) [93]. This information space is confined by a number of so-called "points-of-interest" (POIs). A POI "describes a (simplified) example or prototype object that is given an example or prototype position". Each document has a score/relevance to each POI, which is used to compute a position by calculating a linear combination; thus, this process is similar to a scatter plot with multiple axes (one per POI). If a document depends on one POI only, it is placed right on top of it; if it is equally relevant to two POIs, it is placed in the middle of the line between those. Moving POIs around is a very important task for the user, because it may reveal related logical clusters of POIs and documents. VIBE focuses on relations between documents and POIs, it omits those between documents. But this might be relevant information when it comes to discovering an unknown information space because it helps to find relevant clusters of information, even though they are barely related to the POIs.

The size of the symbols representing the documents encodes the general importance with respect to the information space. Thus, a document with a larger icon has a higher mean relevance. A VIBE diagram is not browsable, in the sense of what is possible with BEAD (focus, context switch). As mentioned above, the main interaction with the system is to set up and arrange the POIs. This approach is less easy to use because it does not provide a helpful visualization without any input. However, this adds some sort of control to the visualization that can be used to better understand the information space.

This concept was improved by Klouche et al. [68] with an additional result list view and the capability of re-ranking the result. Recommended documents are rendered as opaque circles of varying radius. The opacity encodes the density of documents, and the radius the overall relevance. By clicking anywhere in the visual space, the result list is re-ranked, such that documents 'closer' to the click are shown first. As in VIBE, this system does not make the relationships of the documents to each other visible to the user. After all, there is a trade-off between losing structural information and ambiguity when too many relationships influence positioning. An accompanying user study revealed that, compared to a baseline system without spatial representation and interaction, tasks involving perception and retrieval were accomplished in reduced time (111% and 70% faster, respectively), without compromising overall effectiveness.

Additionally, there are other examples of applications which are not solely based on distance-coding. Spoerri introduced InfoCrystal [123] as a solution to address certain challenges faced by VIBE and BEAD. When employing extensive queries, such as incorporating numerous POIs in VIBE, scaling down of multidimensional relationships becomes necessary. InfoCrystal draws inspiration from Venn diagrams, but introduces a structured approach to overcome the limitation of representing relationships among more than three sets. Instead of displaying documents directly, they are represented by iconic representations, with each icon symbolizing a different combination of possible Boolean queries. By selectively enabling or disabling these inner icons, the output can be filtered accordingly. Although this approach simplifies the rendering and querying of documents, it comes at the cost of losing valuable information, such as weighted relationships.

VINETA is a good example in which the query and browsing process blend into each other [72]. Based on a 3D space, documents are represented as spheres, and terms are represented as flying arrows. Interaction is limited to clicking on these arrows. When a keyword is clicked, the user is navigated in the corresponding direction within the space. The more keywords are marked, the less impact the preceding keywords have on the result. To capture the predominant associative structure between terms and documents in the high-dimensional vector space, VINETA employs a graphical multivariate technique known as “biplot”. This technique facilitates the creation of graphical subspaces that effectively represent the main associations among terms and documents. By mapping documents and terms from the vector space to these subspaces, VINETA captures the underlying structure of their associations.

DARE [136] provides the user with a special view, the so-called “visual space” to display the multidimensional document space in 2D. Users build a query vector and choose a reference point, which can sit anywhere and be movable at any given time. The two orthogonal axes in the visual space represent the distance and cosine measure. The angle is formed by two lines, starting at the document vector and the freely chosen reference point, both intersecting at the query vector. The model is suitable for different evaluation models, which essentially shows the strength of a combination of different measures in one visualization environment. This concept is carried out by TOFIR [135], using an angle-angle-based visual space, whereas e.g. GUIDO [92] promotes a distance-distance-based one.

3.3.3 Graph Visualization

Graph visualization is a subfield of information visualization that focuses on structured data. Usually, the goal of graph visualization is similar to what should be achieved in this work: Presenting pieces of information within a 2D or 3D space, inclusive of their interconnections. Herman et al. describe the basic graph drawing problem as follows: “Given a set of nodes with a set of edges (relations), calculate the position of the nodes and the curve to be drawn for each edge" [60]. To accomplish this, prevalent techniques often utilize a conventional *tree layout* (hierarchical approach), spanning tree methods, hyperbolic, and force-directed layouts. Modern graph visualization tools often provide interactive features. Users can zoom in and out, move nodes around, highlight specific paths, and more. This interactivity helps users explore and understand complex graphs more effectively. In the case of dynamic visualizations, where the e.g. positions change, it is important to have enough stability to preserve the viewer's *mental map* [15].

As rendering a huge amount of structured data is computationally expensive and not helpful to users, several approaches are based on incremental exploration techniques, where only a small portion of the complete data set is shown to the user. This shifts the issue of handling the rendering and providing interactive features for huge graphs towards the generation of those subviews, sometimes referred to as *logical frames* [39]. Furthermore, systems need to determine which interactions influence the current frame in which direction.

Again, this is similar to the issue of visualizing recommended information in spatial hypertext systems, yet there are some differences.

Drawing of edges Graph visualization involves the representation of edges, ideally with the aim of minimizing or eliminating crossings. Within the realm of spatial hypertext, relationships are typically conveyed through visual cues, as this approach facilitates the desired implicit and ambiguous interpretation by viewers.

Logical Frames The described logical frames aim to reduce the cognitive and visual load of users, because they have to handle only a small portion of the full picture. A recommendation system does not show this full picture as well, but the focus is different: The provided information is curated and filtered based on user preferences, past behavior, and potentially other users' behavior with similar profiles. This personalized approach aims to guide users to relevant and interesting content. Considering the realm of spatial hypertext again, the change of the logical frame is controlled by the context; the visual structure of the space.

Focus/Context In graph visualization, the issue of focus and context is often solved by groupings or applying other means of hierarchies to hide details when users are interested in the context (e.g. by zooming out). As spatial hypertext follows a "cards on a table metaphor" [116], where the context is influenced by user-interactions (e.g. a user wants to have a specific entity at a specific position), this approach is not applicable without distorting the user's intent.

3.3.4 Summary

In contrast to this previous work, we try to broaden the scope of the system by combining concepts from spatial hypertext research, especially visual/spatial parser to give the machine a sense of what a user is expressing spatially, and visualizing techniques of information retrieval tools. Apart from that, we demand interaction and manipulation of the information space to be a very important possibility to refine queries.

3.4 Exploratory Search

Exploratory search is a type of information-seeking behavior that is more complex than just looking up a specific piece of information. It involves learning, investigating, comparing, and evaluating information on a broad topic. This type of search is often used when the goal or information need is not clearly defined at the beginning, and the searcher may not know exactly what they are looking for. It goes beyond the query-response paradigm [131], as it "describe[s] an information-seeking problem context that is open-ended, persistent, and multi-faceted; and

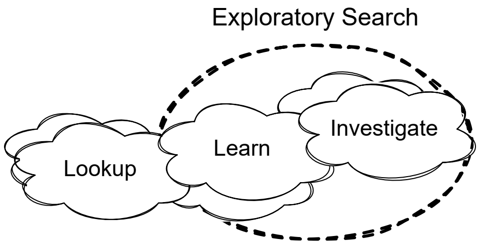


Figure 3.5: Exploratory search in context of search activities

[a] process that [is] opportunistic, iterative, and multi-tactical" [76]. Common search tasks (cf. Figure 3.5) are about fact checking and question answering, usually suited to use cases where the problem statement is clear and users exactly know what information they are missing. Instead, exploratory search is less efficient in the first place. Hence, it is suitable for long-term tasks and when the information need is not clear when starting. End users justify their less efficient strategies to find information by holding onto the belief that accidental or unplanned learning will eventually lead to a positive cumulative effect [77]. *Serendipity* plays an important role in such a context, as it can guide users in the right direction and helps to explore a new area of research.

White et al. defined a set of characteristics that are necessary to support effective and exploratory information retrieval tasks [131]. The utilization of a spatial hypertext interface as a means of communication between user and machine helps in meeting some of these.

Rapid query refinement As users learn and structure information during a session, it is important to give them the ability to update their intention (expressed as a search query) in real time. This implies that visualizations must update and react to those changes in real time as well. In the context of spatial hypertext, the goal is to infer queries based on the visual structure in a 2D space. Although this is easy for users (e.g. drag and drop a node), it sets some restrictions on how the visualization must work.

Facets filtering This describes a very basic filter approach, where users can filter by facets to retrieve information that matches certain criteria. Primary purpose is to swiftly discard information that is clearly irrelevant to the user. Although a spatial hypertext interface does not inherently provide a way to manage these filters, it can be accomplished through additional user-interface elements which need to be added.

Leverage search context This term describes everything the user contributes to the search. His or her background, interests in the past, age, selected favorites[6], and more; when considering spatial hypertext, the space itself. Inferring the context from the visual space has some advantages and downsides. On the positive side, it simplifies the representation of the present state of a search, and even allows for the existence of multiple *subcontexts* within a larger 2D space. However, the downside is that visual relationships are implicit and ambiguous, which can lead to potential misinterpretation.

Offer visualizations The goal of them is to support insight and decision making. Ideally, they are customizable and help to identify relevant information and trends. To achieve this within a spatial hypertext environment, composition algorithms need to meet these requirements. However, given the constraint that everything must occur within the space and follow the cards on a table metaphor, the application cannot utilize conventional information visualization techniques specifically designed to e.g. display trends.

Support learning Systems should help users gain knowledge about the topic they are interested in, regardless of their current level of expertise.

[6]Everything which is marked as useful by the user.

A strength of spatial hypertext is that it allows users to freely organize information. This emerging structure can be memorized as a mental map that supports understanding.

3.5 Collaborative Search

Collaborative aspects in working with computers are increasingly important in an always-online society, where people work from home and are involved in projects with people around the world. For search, collaborative aspects are most important for exploratory tasks when a group is developing a topic together. It is important to differentiate between multi-user settings, where people sit together, interacting with a single system (e.g. on a large screen) and multi-device scenarios, when all users have their own interface. In all cases, applications for collaborative search face similar problems, such as ensuring communication between users – especially in multi-device settings – and finding consensus if there is a common goal [52]. In addition to that, researchers focused on various aspects, such as the division of work between collaborators [87] or advanced communication features, such as the usage of gaze [138] to make others aware of the current focus.

Spatial hypertext may be suitable for both scenarios: A large screen has advantages anyway, as it facilitates a larger visual overview of the space. Users would share the same (over-) view, while it is still possible to focus on local areas. In a multi-device environment, the situation becomes more complex because users might not have the same perspective, which can make communication more challenging. Furthermore, multi-device scenarios may be asynchronous, if not all users contribute at the same time; again, this influences communication. Apart from these challenges, building a mind-map-like structure with multiple people is easier than actually writing text-based queries to lookup information.

Finding consensus is more related to what the system suggests to users than to the interface itself. Therefore, this issue is not addressed in this work. In general, the module responsible for generating recommendations might employ specialized techniques to satisfy the needs of a larger group.

4 Implementation

This chapter describes the architecture of the implemented recommender system and the algorithms for composing recommendations in a spatial hypertext interface in the context of this thesis. Parts of this chapter are based on, or are evaluated by user tests in Chapter 5. The objective of the system outlined here is to meet the requirements specified in Section 1.2.1, while facilitating the resolution of the corresponding research questions. When a section is based on previously published work, it is noted accordingly. As the system grew over time, the various modules of the complete system are detailed in chronological order, to maintain the path which led to certain decisions. The chapter is divided into three sections, which match the three layers of the Mother architecture: Hel (knowledge base), Asgard (structure aware components) and Midgard (UI). The chapter starts with Hel because it introduces some common terms and concepts which are imported for the understanding of the following section about the spatial composition algorithms within the Midgard layer. The chapter concludes with a section on the Asgard layer. It details some ideas on how spatial structure can be utilized to reflect on knowledge bases and to derive queries.

4.1 Hel – Knowledge Base

The knowledge base serves as the foundation of the recommender system, thoroughly curated to align with the requirements of our spatial hypertext architecture. It is designed to support composition algorithms that control the dynamic interplay of recommendations within the spatial hypertext framework. Although the structure of this knowledge base may not be novel or unprecedented, its design is crucial to ensure that it adheres to standards and meets the specific requirements of our system. These requirements are not embodied in the choice of programming language or interface, but rather in the structure that represents information.

A knowledge base imparts structured semantic meaning onto data, enabling interpretation by both humans and machines, thus serving as a necessary step in the transition from mere data to meaningful information. Within our system, we utilize a *knowledge graph* that emphasizes the interconnections between entities represented within the graph. These entities can encompass a range of elements, including objects, events, circumstances, or abstract concepts. Usually, the connections between these entities represent semantic relationships that augment the data with relational context and significance.

This is done because incorporating semantics has advantages, such as the ability to e.g. reason over the data. We have adopted a more versatile framework that aligns with the principles of spatial hypertext. Our approach acknowledges the inherent uncertainties by integrating weighted relationships. These weights help quantify the degrees of association between entities, allowing for a nuanced interpretation that can accommodate the often ambiguous nature of knowledge representation. This flexibility is especially beneficial when the relationships between pieces of information are not always clear.

The setup of the knowledge graph is tailored to the objectives of a particular project or application domain. Each entity within the graph should represent a distinct concept relevant to the domain of interest. In an application designed to offer recommendations for movies and actors, the entities within the graph should correspond precisely to those specific elements. Relationships can be manifold, but must be weighted with $w \in [0, 1]$; in practice, $w = 0$ is the same as no relation at all. For example, consider a knowledge graph with scientific publications [132], and the various possible relations that can be drawn among them. Perhaps they have exactly the same authors; thus the relationship might be weighted with $w = 1$. The keywords are similar but differ to some extent; consequently, this relationship could be weighted with $w = 0.7$. A bag of words embedding reveals that they are about the same topic but employ two totally different studies, thus the weight of this relationship might be even smaller; and so on.

We enforce a fundamental rule on the weight variable w: it must be interpreted linearly. This means that the absolute difference between any two values of w reflects a consistent level of increase or decrease, regardless of the initial values of w'. This is important because – later on – the composition algorithms are based on this rule and produce misleading results if this principle is not applied by the knowledge graph.

Figure 4.1 provides an illustrative example that shows how a typical query is constructed, the structure of the knowledge graph and the format of the corresponding response. Queries focus on entities and are structured as Boolean expressions that combine entities using the *OR* operator. As the knowledge graph may host multiple relationship types, the query can restrict which relationship types should be considered and how they should contribute to the weighted average in the response. This streamlining in the response is due to the design of the composition algorithms, which operate independently of the type or direction of relationships, relying solely on a singular weight metric to signify the connection between two entities. Entities included in a query are identified by spatial parsers, as outlined in Section 3.2.2. Relationship types are not inherent to the spatial hypertext; rather, they must be set up separately by the user when necessary for the use case.

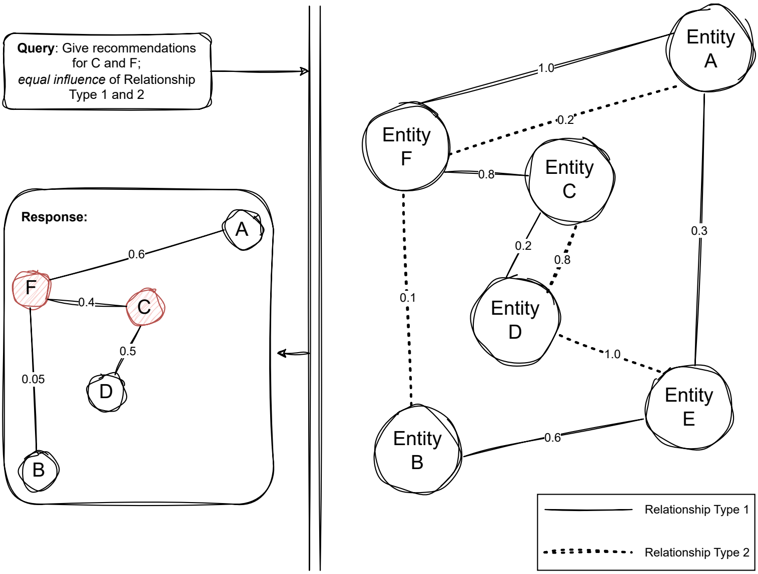


Figure 4.1: Exemplary query towards a knowledge graph that promptly provides the response

4.2 Midgard – UI and Spatial Composition

The user interface plays a key role in a spatial hypertext and recommendation application. It provides the communication platform between the user and backend services. This exchange of information between the user and the recommender system should primarily be facilitated through visual elements. By analyzing the user-created visual structure, the system infers the user's intent to create a query to a knowledge base. Subsequently, the system employs the same visual space to convey its recommendations and feedback, creating a more intuitive and integrated experience for the user.

The look and feel of a user interface, its implementation and technological foundation depends on the specific projects needs, in which the application is embedded. Take Chapter 2 as a reference: If an application is meant to be used by end-users searching for movies for entertainment, the application may be optimized for mobile touch devices; while a tool that supports the development of software is implemented rather for larger screens, operated by a mouse and keyboard. However, we define the following guidelines for implementing a spatial hypertext user interface with recommender functionality.

Endless Space While analog counterparts, such as paper or boards, are limited by their space, this does not naturally apply to digital implementations. The available space should not be restricted in any dimension, to let naturally emerge the structure over time; as big as the user needs it to be. To overcome the problem of getting "lost in hyperspace" [33, 41], the interface may provide clues to help users navigate large spaces e.g. by providing mini maps, rapid zooming [5] or coordinates.

Zooming and Panning In the endless expanse of the spatial interface, users need the ability to navigate effectively. They can do so in two directions: horizontally and vertically. Horizontally, users have the option to move the viewport to focus on different areas of the space. Vertically, they can zoom in to examine details more closely or zoom out to obtain a broader overview. This kind of traversal provides a flexible approach to exploring space, accommodating both granular examination and big-picture perspectives. A smooth zoom, without any predefined zoom steps, is preferred, as it is closer to how users work with paper and can easily be implemented based on mouse-wheel movements and pinch-zoom-gestures on touch devices [10].

Visual Symbols The user interface represents information in the space with visual symbols. Although agnostic to the kind of information, symbols should display a representation of the information piece, such as the name of the concept, a document preview, or the name of the linked resource, and offer an easy way to show details on demand. This applies both to symbols added by the user and to recommendations as well. As it might be difficult to find the sweet spot between too small to get the information and too large to be part of an overarching structure, users should be able to resize symbols. Other visual means, such as color, shape, or orientation, may be controlled by the user.

Additional Controls Depending on the specific purpose of the application, additional controls might be necessary, for example to control filter settings, communicate with colleagues, an so on. They must not permanently cover the spatial workspace. The goal of the spatial hypertext interface is to encompass all elements related to information, structure, and knowledge within the spatial workspace.

Responsiveness Interactions which lead to system service calls and responses should be handled in “real-time”. Real-time in this context means that the system should not introduce any notable delays; responses might be delayed due to network conditions and I/O tasks, but should not take longer than approximately one second. This definition is to some extent arbitrary, but also supported by research on the negative effects of user interface delays, as reported by Simpson et al. [122], or Diehl and Sterman [38]. Users should be able to maintain their flow while working with the application.

Regarding interactions with spatial structure, we adhere to the formal definition of Schedel [p.68-69 114], which says that symbols within spatial hypertext can change:

1. position
2. orientation
3. color
4. shape
5. size

Furthermore, as the space is meant to be *2.5D*[1], symbols can be shifted to another layer. The represented piece of information – Schedel refers to this as "payload", cannot be changed or altered. To change the payload of a visual symbol, it has to be removed from the space, and a new symbol with the updated payload has to be created.

4.2.1 Preliminary Considerations

The Midgard layer comprises the user interface, ideally following the guidelines mentioned above. The spatial hypertext workspace is meant as a tool to organize information visually and as a basis for a spatial parser to retrieve the user-intended structure to provide recommendations from knowledge basis. The integration and visualization of these recommendations within the workspace are discussed in the following sections. Following the information visualization literacy, algorithms can utilize (a combination of) multiple visual variables to encode the relationship between recommendations and user-generated structure, as well as between recommended information pieces. A well-designed

[1]the .5 dimension refers to multiple, ordered layers

visual encoding helps the user to quickly grasp complex relationships. It can reveal hidden patterns or trends, provide clarity, and reduce cognitive load. The encoding should be consistent and align with common principles of visual literacy, making it intuitively understandable even for users who are new to the system. Just as spatial parsing transforms implicit spatial structures into explicit weighted graphs, the objective here is to perform what can be referred to as a *spatial composition*. In this context, the recommendations, represented by a weighted graph, are meaningfully composed within the given space. Spatial composition comprises more than just finding a suitable arrangement; it should react to user interactions to gradually update the composition.

For this thesis, we focus on *distance* as visual variable. Other variables, e.g. the color of the recommendations, are not considered, although they might improve the perception of relationships. The visual (2D) space, with its potentially high number of symbols, lacks the capability to generate an appropriate mapping of all pairwise weighted relationships to distances. Consequently, all subsequent approaches face a balancing act between maintaining precise distance coding and ensuring seamless integration within an interactive hypertext application. Sections 4.2.2, 4.2.3, and 4.2.4 describe first experiments and our findings, which influenced the implementations in sections 4.2.5 and 4.2.6 – both are used in production-ready applications, as detailed in Chapter 2.

As most parsers for spatial hypertext are agnostic to the payload of any visual symbols, visualizers do not interfere with the actual recommendation itself. They should provide a bounding rectangle, its position, and size in the space. The actual recommendation must be within these limits and may be represented as a simple text or as a rich visual composition.

Context This term is heavily used in the following sections and seeks some clarification. A general definition such as e.g. given by the Cambridge Dictionary already offers a good idea of the term: "[T]he situation within which something exists or happens, and that can help explain it"[2]. In the discussion about spatial composition, the term *context* is characterized by the complete set of attributes that affect how spatial hypertext is understood by parsers, following the description provided by Schedel [114]. Considering that the spatial hypertext is made up of information units depicted as dimensional visual symbols, which are herein referred to as *user concepts*, this implies that the context is influenced by:

- any visual change of a symbol; parsers recognize a change of color, shape, orientation, and size
- a positional change of a user concept
- any interaction with user concepts; parsers recognize click/touch interactions
- the addition and deletion of user concepts

The content of the user concept, as previously established, is explicitly not considered; alterations or modifications to the content are not permitted. Furthermore, the user's viewpoint, the perspective, or the scale ('zoom') on the boundless canvas are also not included. Simply put, context covers the machine's perspective of the spatial hypertext. Although it strives to offer an interpretation that aligns with human reasoning, its abilities are inherently limited. Since the composition is executed by a machine, it is logical to define the context in terms of what the machine can perceive and process.

Given the definition of contextual characteristics, it becomes imperative to precisely convey the intended meaning. The parsers operate on a complete weighted graph, representing the spatial and visual relations between user concepts. To facilitate the graph, the interpretation engine discerns groups of user concepts that share a close visual relation. These groups embody what is referred to as *context*. The term *meta*

[2]https://dictionary.cambridge.org/dictionary/english/context

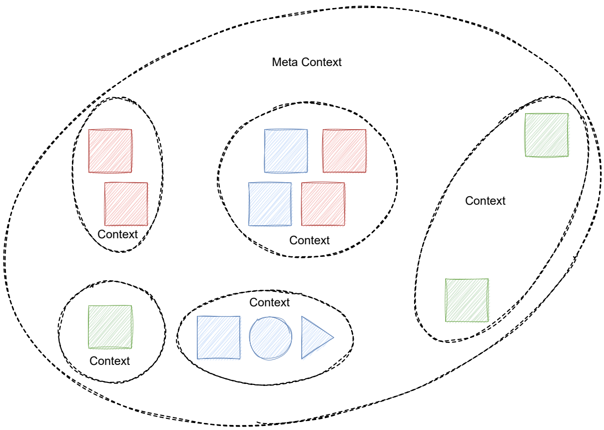


Figure 4.2: Illustration of context and meta context

context denotes the interrelations among these groups. An alteration in any of the aforementioned attributes is characterized as a *context update*.

The illustration in Figure 4.2 presents a draft of a spatial hypertext, filled with user concepts of various shapes and colors. These concepts are encircled to denote groupings – or *contexts* – as recognized by the parsers. The collective ensemble of these contexts is referred to as the *meta context* and is indicated by the outer circle encompassing it.

4.2.2 Circular Composition

The circular composition is a simple composition algorithm to set the scene. It is the result of first experiments, and its main purpose is to identify challenges and to collect early user feedback. It draws inspiration from the concept of fish-eye views, where a central point is highlighted and information is easily readable, while details outside the focus gradually diminish in size and clarity towards the edges of the viewport. This concept can be adapted to a spatial hypertext interface. First of all, it is necessary to redefine the focal point, as there is no natural counterpart in spatial hypertext. Instead, we utilize spatial parsers to identify groups of visual symbols that are visually related (context). The bounding rectangle around these symbols can be used as *focal area* instead of a point.

In principle, visual symbols of a group can be scattered across the space, as their relations do not solely rely on proximity but also on other visual properties and even user interactions. This would lead to potentially overlapping focal areas, hence relationships and the belonging of recommendations would become unclear. Typically, this situation is quite rare, as distance predominantly influences the parsing result, and hence the presented algorithm does not mitigate this potential issue.

Algorithm

Sarkar et al. [113] noted that "[g]enerating a fisheye view involves magnifying the vertices of greater interest and correspondingly demagnifying the vertices of lower interest. In addition, the positions of all vertices and bend points must also be recomputed to allocate more space for the magnified portion [...]". Further, the authors detail a transformation function that maps vertices and edges of a graph to new positions and changes their size, depending on a distortion factor and the focal point. When composing recommendations, there is no pre-existing structure that can be transformed. Rather, to create a fish-eye effect, the position and size of the recommendations must be calculated in advance. The focal point – or area in this case – is determined solely by the bounding rectangle that encompasses the symbols of a group. Therefore, the algorithm does not introduce any distortion in this area as it would conflict with the structure created by the user.

As for all composing algorithms described later on, the circular composition is fed with the aforementioned group of symbols and recommendations in a graph format; including the recommendations themselves and user-added concepts as vertices, and weights from the knowledge base as edges (cf. Figure 4.1). To simplify the composition, weights between individual recommendations and visual symbols of the user are not considered. This means that the input graph of recommendations is simplified to a minor[3] of the original one. An edge contraction is applied on all edges between the user-added concepts, effectively merging these vertices into one remaining. The resulting parallel edges are aggregated by calculating the mean, considering non-existing edges with a weight of 0. Eventually, the algorithm is fed with specific recommendations, accompanied by their average weight in relation to the concepts added by the user.

As the name suggests, the occupied space is circular and shaped around the focal area. To establish both the magnitude of the magnetization and the position, the algorithm generates what are termed *layers* L_i. Each of these layers contains recommendations that fall within a specific weight range. In its default settings, the algorithm distributes recommendations in the three intervals I_i $[1, \frac{2}{3}[$, $[\frac{2}{3}, \frac{1}{3}[$, and $[\frac{1}{3}, 0]$ – in general, those intervals are meant to be evenly distributed. An example rendering is shown in Section 4.2.2, where the three layers comprise 10, 20 and 40 recommendations, respectively. If the algorithm is fed with more recommendations than can be accommodated within a given interval, additional layers will be created.

Before composing the recommendations in the layers, the algorithm determines the necessary size. It depends on the upper bound of the current layer's interval $max(I_i)$, the radius of the circle enclosing the bounding box of the group R_{center}, and a custom factor $\beta > 0$ that defaults to 0.25, as formalized in Equation 4.1. R_1 is the radius that describes the circles for the recommendations in the very first layer, R_0

[3] A minor of a graph G can be created from G by removing edges and vertices, or by contracting edges

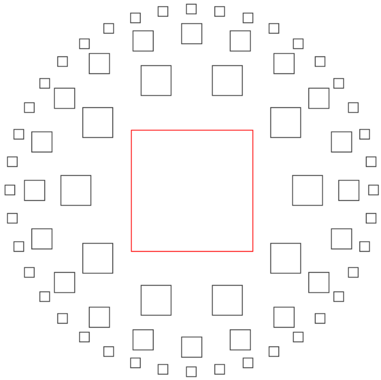


Figure 4.3: Circular composition comprising 3 layers with 10, 20, 40 recommendations each

is set to be equal to R_{center}.

$$R(i) = \beta \cdot \max(I_i) \cdot R_{center} \tag{4.1}$$

The distance between layers and between the first layer and the center's circumcircle is called *altitude* A_i and is calculated similarly to R_i, but with a custom factor $\alpha \geq 0$ that defaults to 0.1, cf. Equation 4.2. When the radius of the prior layer/circumcircle is added as offset, this results in the radius of the inner border; adding R_i results in the outer border of the layer.

$$A_i = \alpha \cdot \max(I_i) \cdot R_{center} \tag{4.2}$$

To determine how many n_i recommendations fit in a layer L_i, the algorithm uses Equation 4.3:

$$n_i = \frac{\pi}{\arctan\left(\dfrac{R_i}{R_{center} + R_i + A_i + R_{i-1}}\right)} \tag{4.3}$$

If there are more recommendations for a certain interval than fit

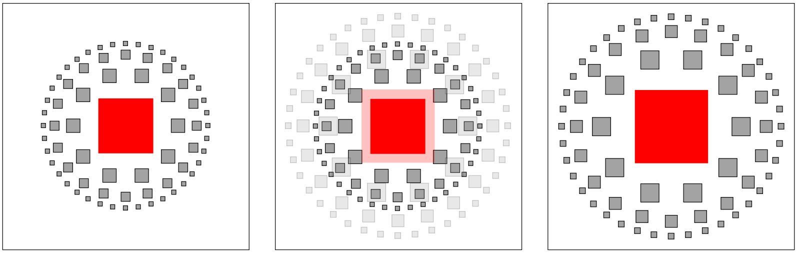


Figure 4.4: Circular composition: left and right images show slightly different contexts, the middle image does overlay both versions

in the respective layer, an additional layer is created. The space allocated for the recommendations remains constant, but the distance from the focal area obviously increases. As a result, although the recommendations have weights within the same interval, they are sorted by priority, ensuring that the most relevant recommendations appear in a closer layer.

Conclusion

The circular composition is capable of visualizing many recommendations. Layering and magnifying them makes it easy to grasp important information, as they are close to the area of interest. Furthermore, the composition is able to handle user interactions quite well: If the focal area changes, e.g. because the user updates the structure of the inner group, the layers and recommendation areas in- or decrease accordingly with a smooth transition. Figure 4.4 shows how a change in context influences the composition. However, this potential solution comes with certain challenges that need to be addressed in subsequent iterations.

Input Dependence Although the algorithm consistently produces visually appealing results irrespective of the input knowledge graph, it does not fully meet all objectives from a spatial hypertext perspective. First, the relationships between recommendations and individual symbols of the group vanish, as when contracting edges, all parallels are merged to a single average weight. Depending on the number of objects in the group and the kind of structure, this may lead to situations where the averaged relation weight may lose its significance. The lack of an accessible visual relationship between recommendations adds to this situation. Secondly, discrete layering obscures the continuous nature of the values by confining them within fixed interval boundaries. Depending on the knowledge graph and the context, it might be the case that all recommendations fit in one interval. Again, this would make it impossible for users to gain a deeper understanding of the interconnectedness of the incoming knowledge. Last, this composition reserves a lot of slots for recommendations of potentially lower interest for the context. However, the space allocated for each recommendation is quite limited, making it challenging to effectively display their actual content, and thus reducing the utility for users.

Context Updates If the context undergoes a change – such as the addition of new concepts to the group – the recommendations could also change. Specifically, they may either lose their relevance or experience a shift in their averaged weight, which in turn would affect their position and magnification. The described approach lacks a strategy on how to handle these changes e.g. by providing transitions for recommendations from their original to their new position. This is important for users to keep track of how the context change affected the results.

Missing Structure Adaptability Apart from calculating the bounding box of a group of objects, the actual structure is not recognized or utilized. Therefore, the composition remains the same, regardless of whether the structure of the group forms e.g. a list, where the order of objects has a special meaning for the user, or is a loose accumulation

– as long as the bounding box remains. Complex structures might be more aptly represented by concave polygons, possibly featuring interior voids, which provide sufficient space to meaningfully display recommendations.

The circular composition was very helpful to get some first impressions of how spatial composition can work in a spatial hypertext system. Although not perfect, this implementation was used in a research prototype (cf. Section 2.2), which was discussed with the project partners to obtain early feedback. The subsequent composition algorithms are designed with the aim of addressing the previously identified shortcomings.

In particular, we identified space utilization as a strong weakness of the circular composition; something we wanted to improve with the next experimental algorithm.

4.2.3 VIBE Composition

As the name suggests, this composition draws its inspiration from a system called *VIBE* (cf. Section 3.3.2). VIBE refers to the user organizing visual symbols, which represent various concepts, within a workspace. This arrangement establishes a spatial context in which recommendations are displayed. The VIBE composition differs from the original VIBE visualization because it attempts to avoid overlapping recommendation areas while maintaining an accurate positioning. Visual symbols encompass the concept they represent as payload in some way. The overlap of these symbols may be tolerated only if the hidden parts do not convey important information. This is necessary to properly assess, for example, a recommendation. Like the circular composition, this visualization is based on detected groups of user concepts. The improved utilization of space compared to the circular composition is achieved because the room between user concepts is utilized to compose the recommendations.

Furthermore, this approach allows for the encoding of relationships between recommendations and individual user concepts of a group such that it is not necessary to build a minor of the recommendation graph in advance.

Algorithm

First, each visual symbol that represents a user concept is converted to a single point that marks the center of the symbol. For example, the center point of a rectangle would serve as its representation. The resulting points p_i are the basis for the following calculations and represent the display information (x_i, y_i). Olsen et al. [93] used the following mapping function, where the weight α is normalized such that $\sum \alpha_i = 1$ and n is the number of "POIs" – user-controlled concepts in our case.

$$(x, y) = \sum_{i=1}^{n} \alpha_i (x_i, y_i) \tag{4.4}$$

Because the weight – α – is normalized as described above, the resulting position is *relative*; it hints the user to POIs with the highest relative weight with respect to a recommendation. In the case of VIBE, this approach works quite well, because the system uses additional visual cues to represent the *absolute* values. The overall relevance of a recommendation is represented by the mean weight, which is encoded through the size of the visual symbol, similar to what is done in the circular composition. The specific influence of individual POIs can optionally be displayed through additional vectors, called "star-option", superimposed on the recommendation. The length of these vectors indicates their absolute value.

As our composition aims to encode knowledge in a spatial hypertext environment, relative positioning is not suitable. Taking into account a situation with e.g. three POIs, and a recommendation with a relationship weight of $w_1 = 0.2$ with the first POI and $w_2 = w_3 = 0$, α-values would result in $\alpha_1 = 1$ and $\alpha_2 = \alpha_3 = 0$. Consequently, the recommendation would be placed on top of the first POI, as it is the only one exerting any influence, even though the absolute weight is relatively low.

Therefore, we decided to adapt the VIBE approach in a way that takes the absolute weights into account. In a first step, the algorithm determines a center point, which is defined as c, the result of Equation 4.4, with $\alpha_i = \frac{1}{n}$. Then the product of the vectors – between c and p_i – and w_i is added to c, as formalized in Equation 4.5. Depending on the exact placement of user concepts, a nearby composition of a recommendation would suggest a high weight.

$$(x, y) = c + \sum_{i=1}^{n} w_i \cdot (p_i - c) \tag{4.5}$$

When a group has more than three concepts, the overlap of two recommendations can be genuine, which means that they are identical with respect to the specific concepts – or it could be misleading – arising from the spatial superposition of distinct locations. In other words, the apparent coincidence might indicate a true correlation based on the chosen POIs, or it could be simply a visual artifact caused by the projection of different points onto the same space [93]. Data with more dimensions than the interface can represent always suffer from this; however, overlapping on its own is still an issue.

Because our recommendations should not be merely points or visual symbols, but should convey a representation of the wrapped information, overlapping should be avoided. To overcome this issue, we define a metric that benchmarks the position of a recommendation. The first factor M_1 of the metric measures the overlap of a recommendation with other objects. To this end, the area of a recommendation is set in relation to the total covered area. Since overlap with multiple objects is possible, this ratio can exceed 1; see Algorithm 4.1 for details. The goal is to keep this factor as low as possible, ideally $M_1 = 0$ which means

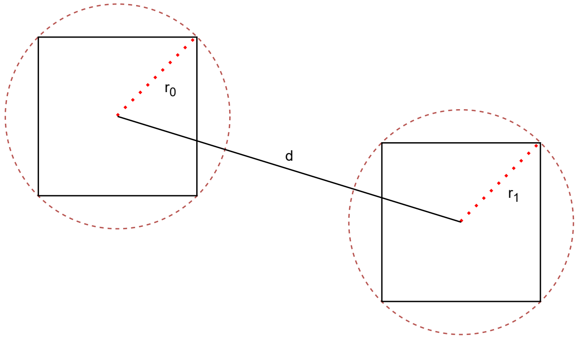


Figure 4.5: Two objects, their radius and the distance between both centroids

that the recommendations do not overlap each other.

Algorithm 4.1 Overlapping metric

Input: recommendation area $recA$**, areas of other** $objs$
Output: overlapping factor M_1

1: $intersectA \leftarrow 0$
2: **for all** aObj $\in$ objs **do**
3: $\quad intersection \leftarrow$ aObj $\cap$ recA
4: $\quad intersectA \leftarrow intersectA + intersection$
5: **return** $intersectA \div recA$

The second factor, M_2, represents the *divergence* of a recommendation, measured by its relative distance to the position of the original composition, as calculated in Equation 4.5. The algorithm calculates these *relative distance values* between a recommendation and the user concepts. These values are called 'relative' because, at this step, we consider the dimensions of the symbols, rather than just their centroids.

Figure 4.5 depicts the variables needed to calculate the relative distance. The radius of the object serves as an indicator of size and is utilized in relation to the distance between the centroids. The relative distance d_{rel} as calculated as follows:

$$d_{rel} = \frac{d}{r_0 + r_1} \tag{4.6}$$

The divergence (M_2) is defined as the mean difference between the relative distance values of the original position (as in Equation 4.5) and any other position that could be used to reduce the problem of overlap. For all n user concepts, there is one corresponding $d_{rel}(i)$ for the original position and a $d^*_{rel}(i)$ reflecting the relative distance from any other position. M_2 is then calculates as follows.

$$M_2 = \frac{\sum_{i=1}^{n} |d_{rel}(i) - d^*_{rel}(i)|}{n} \tag{4.7}$$

Simply put M_2, the divergence, is 0 for the original calculated position. The further a visual symbol is moved away from these original coordinates, the larger M_2 becomes. In practice, the algorithm aims to reduce overlap, measured by M_1 and divergence, measured by M_2. The overall metric M weighs both factors equally, as shown in Equation 4.8.

$$M = (M_1 + 1)(M_2 + 1) \tag{4.8}$$

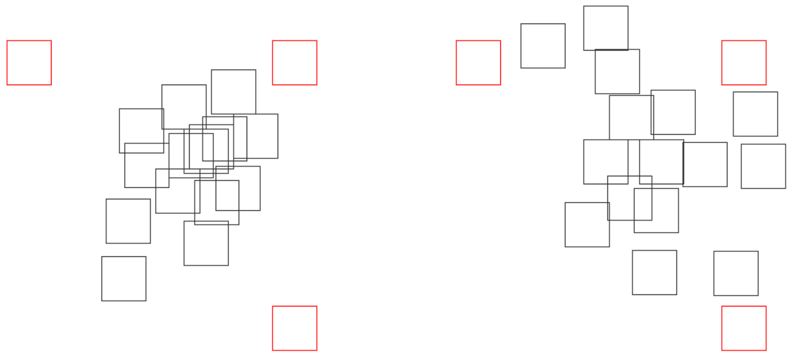
To avoid the product of M_1 and M_2 being zero when one of the factors is zero, both M_1 and M_2 are increased by 1. Therefore, the 'perfect' value of M is 1, indicating that the recommendations reside in their original position and do not overlap with any other objects in the space. Since M_1 is unbounded, M also lacks an upper limit.

But how does the metric M help solve the overlap problem mentioned above? The algorithm sorts the recommendations according to their mean weight to the group of objects to which they belong and positions them one by one. As the number of objects in the space grows, the likelihood of initial overlapping between objects also increases. Whenever this happens, an iterative algorithm *shifts* the position of the recommendation a certain number of times in a certain way to find a (local) minimum of M with the help of the Nelder-Mead method (downhill simplex). The Algorithm 4.2 describes this process in detail. It obtains (and sorts) the recommendation rectangles, determined in Equation 4.5, and is aware of static objects, such as user concepts and already processed recommendations.

Algorithm 4.2 VIBE composition shift

```
    Input: recommendations recs,
     already positioned objects objs
maxIterations ← 10                                   ▷ can be changed
sortedRecs ← SORT(recs)
for all rec ∈ sortedRecs do
    hasIntersections ← (rec ∩ objs) > 0
    if hasIntersections then
        S ← CREATESIMPLEX(rec)                       ▷ 3 coordinates S_1, S_2, S_3
        for n = 1, ..., maxIterations do
            testM ← [...]                            ▷ Yields M_1 and M_2 for S
            for i = 1, ... 3 do
                testM_i ← [M_1(S_i), M_2(S_i)]
            S ← SORT(S)        ▷ Ascending, according to M, see testM
            S_new ← MOVESIMPLEX(S)                   ▷ as in [89]
            if CHECKCONVERGENCE(S, S_new) then
                breakloop
            S ← S_new
        rec_shifted ← CENTEROF(S)
        objs ← objs + rec_shifted
    else
        objs ← objs + rec                            ▷ Add without shift
```

As in the circular composition, the whole process is repeated whenever relevant input data changes; namely the position of user concepts or (the weight of) recommendations. Therefore, there is a need to keep the complexity and total computation time as low as possible. Finding an optimum (minimum in this case) for M cannot be done in a guaranteed amount of time, because a downhill simplex may converge to non-stationary points, or get caught in a local minimum. Ultimately, the algorithm sets an iteration threshold to guarantee consistent runtime, albeit with the potential for suboptimal results. Furthermore,



(a) identical weight (b) random weight

Figure 4.6: VIBE composition with 15 recommendations and 3 user concepts; recommendation shift

a convergence check may interrupt further optimization if defined criteria are met. For this composition, the procedures for convergence check, initial simplex creation, and simplex movement are not rigidly specified. Instead, we employed a heuristic approach. For instance, the initial simplex is built based on the dimensions of the recommendation area. The method for moving the simplex is consistent with that described in the original publication [89]. Additionally, the convergence check examines whether M_1 or M_2 has undergone a change of at least 0.1.

The way this shifting process influences the result becomes apparent when looking at Figure 4.6a. The three red rectangles represent user concepts that provide the context within the recommendations are composed. The recommendations, or more specifically, the rectangles within the recommendations are to be displayed, are colored black, and have a uniform weight in relation to the user concepts. Thus, the initial position of all recommendations would be the same, directly on top of the centroid. As M is going to be optimized within the constraints and convergence settings given, all recommendations except the first are moved away from the center. Because the algorithm considers two factors (M_1 and M_2) and does not guarantee the finding of an optimum, the resulting layout shows some imperfections, e.g. overlapping. A more common situation is shown in Figure 4.6b: Here, recommendations are given random weights[4]. However, some overlap may still occur in the composition if the shifting process does not locate a more optimal position within the prescribed iteration limit.

Conclusion

The circular composition (cf. Section 4.2.2) has been critiqued for considering only the average weight of each recommendation, without accommodating the specific structure. A goal of the VIBE (-inspired) composition is to overcome these shortcomings. It is not necessary anymore to calculate an average weight, as the position is now determined by the weights a recommendation has towards individual user concepts already in space. As Olsen et al. [93] described in their work, an average – as an indicator of general relevance to the given context – could be used to control other visual properties. For instance, a recommendation could have a larger size if it is of high relevance. Additionally, the VIBE composition is adaptable to the user-defined structure in that it utilizes the space anchored by the user's concepts[5], rather than being constrained to a pre-determined space or shape, such as a circle. Yet, there are still issues with this composition when used in an interactive spatial hypertext context.

Relative Positioning The composition lacks a constant (visual) mapping of *relevance* to *distance*, because the distance used in the algorithm is relative to the distance between the centroid of the structure and the user concepts. One can imagine a scenario where there are four user concepts arranged in a square formation and a recommendation has the

[4] Drawn from a uniform distribution
[5] The simplex might leave this area during the shifting process

same weight in relation to all these user nodes. Whatever this weight might be, the resulting composition is the same, without giving the user any clue about the weight. Indeed, with the center c as a reference, an observer can only discern whether a recommendation displays significant variance in weight relative to the user concepts in the space and identify which user concepts are associated with these higher weights.

Space Utilization A major flaw of the VIBE composition is its very limited space utilization, because recommendations are only arranged within the space restricted by the user concepts. Basically, this is very similar to the situation created by the circular composition, where the recommendations are placed *outside* the given context. The inner space may be, depending on the built structure, rather small, only offering the possibility to compose a few recommendations. Additionally, spatial hypertext often prompts users to organize their concepts into formations such as lists or similar structures, leaving limited available space within those arrangements. Ideally, a composition algorithm should make use of both the space inside and outside user-defined structures to augment the structure created by the user in a meaningful way.

Context Updates This issue was already mentioned for the circular composition and applies equally to the VIBE composition. When recommendations are changed, there is no transition strategy to e.g. exchange recommendations not needed with (then) more relevant ones. In the circular composition, minor context updates, such as a slight change in the position of a user node that does not affect the spatial parsing outcome, are handled well. This is because the circle simply adjusts its radius, leading to a smooth transition in both the size and position of the recommendations. The initial layout determined by the first step of the VIBE composition is as stable and smooth, but the optimization of M is not. As a result, recommendations might show a flickering effect, alternating their positions from frame to frame. This

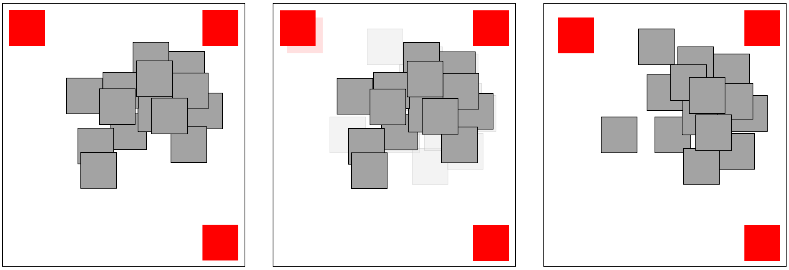


Figure 4.7: VIBE composition: left and right images show slightly different contexts, the middle image does overlay both versions

makes it difficult or impossible for users to trace those changes in the space. This effect can be seen in Figure 4.7. The image on the left shows a situation similar to that in Figure 4.6. The image on the right depicts a situation where the users slightly moved the top left user concept towards the center; this context change triggers a re-run of the composition algorithm, visible by the updated positions of the recommendations. The image in the middle is an overlay of the other two images and shows the rather irregular position changes of the recommendations, followed by a minor interaction.

4.2.4 Heat Map Composition

This composition is a follow-up to the circular and VIBE compositions and aims at improving (almost) all the aforementioned flaws. The name is technical in nature to accurately describe its function. It draws inspiration from heat maps, commonly used in various scenarios to encode the magnitude of a value in a 2-dimensional setting with a color. As an example, this thesis also uses heat maps to visualize data – cf. Figure 5.6. A significant challenge observed in spatial composition is the inefficient use of available space. Limiting the areas where recommendations are laid out prevents optimal display and potentially hinders users from interpreting the composition. Additionally, a relative coding from weights to distances makes it impossible for users to sense the real values. In this composition, the notion is that the user concepts 'radiate' a field, similar to heat or electromagnetic radiation in general, which decreases in intensity as the distance from it increases.

At any given point, the field can be measured with a value between 0 and 1, directly reflecting the notion of weight in the knowledge graph. Simply put, a recommendation of a certain weight toward a user concept can be positioned on any point within the field that matches this weight. Typically, multiple user concepts contribute to the context, each emitting its own field – imagine these as varying in frequency to maintain the metaphor. A recommendation is optimally placed at a location where the divergence between its weight and the combined influence of these overlapping fields is minimized. As a design choice, this composition algorithm operates in a discrete mode. This term reflects the discrete nature of the field because, unlike the VIBE composition, the algorithm does not optimize a continuous function. Instead, it calculates specific values of the field and returns the optimal position based on the given resolution.

Similarly to electromagnetic radiation, the strength of the field at a specific location is influenced not only by its distance from the source of radiation (in this case, a user concept) but also by the medium through which the waves travel. In the context of a spatial hypertext environment, this implies that the strength of the field diminishes when there are obstacles (e.g. other user concepts) between the source and the point of interest. Before going into the details in the next section, Figure 4.8 depicts an exemplary field radiated by a single object (e.g. a user concept). The black square in the upper half of the picture represents an obstacle that absorbs radiation. The degree of attenuation can differ depending on factors such as the type of object involved. Attenuation caused by empty space can also vary. Note that the numbers in this picture may vary from those calculated by the algorithm, depending on

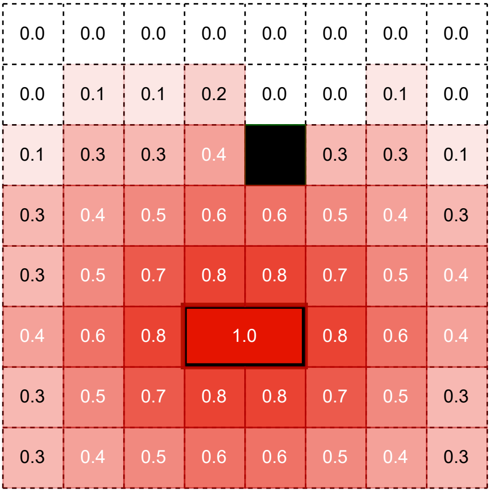


Figure 4.8: Exemplary heat map field radiated by the red, bordered rectangle in the lower half

the settings, such as the resolution.

In addition to simplifying the optimization process, the discrete field also facilitates the prevention of overlap. Once a recommendation is placed within the space, the area it occupies can be subtracted from the remaining available space. This effectively ensures that no other recommendations will be placed in the same location.

Algorithm

As already explained, the algorithm is meant to utilize more space while easing the optimization step. Therefore, the initial steps of the heat map composition define the space in which recommendations can be composed and the resolution that is used to calculate the field. In essence, the algorithm constructs a rectangle with a relative margin around the user concepts, controlled by a factor f_{area}. This rectangular area is then partitioned into smaller squares, similar to the squares on a chessboard. The granularity of these squares corresponds to the resolution: A higher resolution means that the rectangle will be divided into a greater number of smaller squares.

The calculation is detailed in Algorithm 4.3. Note that the resolution is not determined solely by a fixed factor; instead, it is based on the smallest dimension of any user concept, in any direction, and a factor called f_{res} (cf. Algorithm 4.3).

Right after, the algorithm builds one *heat map* for each user concept, filling the cells of the recommendation area with values – as depicted in Figure 4.8. This procedure is carried out separately for each user concept, taking into account both the distance from the cell to the user concept and the 'matter' in the cells that lie between them. In the heat map composition, the terms *air* and *concrete* are used metaphorically to represent the matter of empty cells and the concepts of users that act as obstacles affecting the strength of the radiation of other concepts, respectively.

At first glance, taxicab geometry appears to be a suitable metric for measuring distance, especially since the area of interest has already been divided into discrete cells. But human perception is better described by Euclidean geometry, which measures the distance as a straight line between two points in space. Therefore, the heat map algorithm also draws a straight line to measure the distance and computes all cells which are crossed by the line to calculate how much the matter in-between influenced the radiated field.

All this is done by a *GenerateHeatMap*-procedure, which is called for each map to be generated. First, the algorithm maps the *emitter* – the concept that radiates the field – to the cells designated for recommendations within the area. This step is crucial because the emitter, like any other user concept, could span more than a single cell. To prevent overlaps in the final composition, a cell is allocated to the object as soon as they share even a minimal overlap. Algorithm 4.4 details the process, fed with the output of the initial area setup (cf. Algorithm 4.3) and an emitter for which the heat map should be generated.

Algorithm 4.3 Initial area setup; the result is highlighted

Input: **area factor** f_{area},
resolution factor f_{res},
user concepts $objs$

$minX \leftarrow \infty$
$minY \leftarrow \infty$
$maxX \leftarrow -\infty$
$maxY \leftarrow -\infty$
$minDim \leftarrow \infty$
for all obj $\in$ objs **do**
    if obj.minX $<$ minX **then**
        $minX \leftarrow obj.minX$
    if obj.minY $<$ minY **then**
        $minY \leftarrow obj.minY$
    if obj.maX $>$ maxX **then**
        $maxX \leftarrow obj.maxX$
    if obj.maxY $>$ maxY **then**
        $maxY \leftarrow obj.maxY$
    if obj.width $<$ minDim **then**
        $minDim \leftarrow obj.width$
    if obj.height $<$ minDim **then**
        $minDim \leftarrow obj.minDim$
$areaWidth \leftarrow (maxX - minX) \cdot f_{area}$
$areaHeight \leftarrow (maxY - minY) \cdot f_{area}$
▷ *The area with x and y coordinates + dimensions* ◁
$area \leftarrow$ RECTANGLE(minX - (areaWidth - maxX + minX) / 2, minY - (areaHeight - maxY + minY) / 2, areaWidth, areaHeight)
$f_{cell} \leftarrow minDim \cdot f_{res}$ ▷ *width/height of a cell*
▷ *Next two lines determine the number of rows and columns; the resolution* ◁
$nRows \leftarrow areaHeight / f_{cell}$
$nColumns \leftarrow areaWidth / f_{cell}$

Algorithm 4.4 Object to cell mapping, input from Algorithm 4.3

Input: recommendation area $area$,
cell width/height f_{cell},
user concept $emitter$

function GenerateHeatMap($area, f_{cell}, emitter$)
  $xDistToArea \leftarrow emitter.minX - area.minX$
  $yDistToArea \leftarrow emitter.minY - area.minY$
  $xDistToAreaLow \leftarrow emitter.minX + emitter.width - area.minX$
  $yDistToAreaLow \leftarrow emitter.minY + emitter.height - area.minY$
  ▷ *The following four lines determine which cells represent the corners of the emitter* ◁
  $x_{top} \leftarrow \lfloor xDistToArea/f_{cell} \rfloor$
  $y_{top} \leftarrow \lfloor yDistToArea/f_{cell} \rfloor$
  $x_{bottom} \leftarrow \lceil xDistToAreaLow/f_{cell} \rceil$
  $y_{bottom} \leftarrow \lceil yDistToAreaLow/f_{cell} \rceil$
  $x_{mid} \leftarrow (x_{top} + x_{bottom})/2$
  $y_{mid} \leftarrow (y_{top} + y_{bottom})/2$
  return $x_{top,bottom,mid}, y_{top,bottom,mid}$

Once the composition has assigned all user concepts to their corresponding cells, it populates the remaining cells with values that correspond to the strength of the field. Again, this process has to be repeated for all user concepts. Basically, the algorithm loops all cells that do not refer to the emitter or other user concepts (concrete). Then, a path is constructed linking individual cells to the nearest cell associated with the emitter. This path includes all cells intersected by a direct line drawn from the center of the target cell to the closest emitter-associated cell. The length of the path is determined by the number of cells it crosses, which is further adjusted on the basis of the type of material in the cells (either air or concrete). Algorithm 4.5 details the path generation; parameters are the indices of the cells in a 2D array. For example, the indices $x_{source} = 0$ and $y_{source} = 1$ refer to the cell in the *first row* and *second column*. The algorithm calculates a target angle α and aims to approximate this angle by traversing the grid horizontally, vertically, or diagonally – cell by cell.

The path yields all the information necessary to compute the strength of the field in a certain cell. This is achieved by assigning radiation attenuation coefficients to both *air* and *concrete*. Specifically, *air* allows most of the radiation to pass through, while *concrete* (representing other user concepts) attenuates radiation more significantly. Of course, this is not a simulation and does not aim to mimic reality; nevertheless, the coefficients are named μ_{air} and $\mu_{concrete}$. In the initial version of the heat map composition, we opted for a subexponential decay of radiation, diverging from natural systems where the radiation influence tends to be exponential. This choice was made to provide a distinct behavior and to experiment with how various attenuation models impact the composition. Equation 4.9 shows the calculation, while I is the resulting strength of the field at a certain point, I_0 the strength of the field as it is radiated by the emitter, typically 1.0, a and c represent the number of cells with air and concrete, respectively. Obviously, $a+c$ equals the total distance (cells between) from the emitter to the cell of

Algorithm 4.5 Building the path between two cells

Input: Cell coordinates $x_{source}, y_{source}, x_{target}, y_{target}$

function BUILDPATH($x_{source}, y_{source}, x_{target}, y_{target}$)
  $path \leftarrow [...]$ ▷ *Yet empty path*
  if $x_{target} - x_{source} \geq 0$ **then** ▷ *Determine target quadrant*
    $Q_x \leftarrow 1$
  else
    $Q_x \leftarrow -1$
  if $y_{target} - y_{source} \geq 0$ **then**
    $Q_y \leftarrow 1$
  else
    $Q_y \leftarrow -1$
  if $x_{source} == x_{target}$ **then**
    for $i = y_{source} + Q_y; i \cdot Q_y \leq y_{target} \cdot Q_y; i = i + Q_y$ **do**
      $path \leftarrow path + [x_{source}, i]$ ▷ *index coordinates*
  else if $y_{source} == y_{target}$ **then**
    for $i = x_{source} + Q_x; i \cdot Q_x \leq x_{target} \cdot Q_x; i = i + Q_x$ **do**
      $path \leftarrow path + [i, y_{source}]$ ▷ *index coordinates*
  else
    $x_{tmp} \leftarrow x_{source}$
    $y_{tmp} \leftarrow y_{source}$
    $\alpha \leftarrow \arctan \frac{y_{target} - y_{source}}{x_{target} - x_{source}}$
    while $x_{tmp} \neq x_{target} \wedge y_{tmp} \neq y_{target}$ **do**
      $\beta_{diagonal} \leftarrow \arctan \frac{y_{tmp} + Q_y - y_{source}}{x_{tmp} + Q_x - x_{source}}$
      $\beta_{vertical} \leftarrow \arctan \frac{y_{tmp} - y_{source}}{x_{tmp} + Q_x - x_{source}}$
      $\beta_{horizontal} \leftarrow \arctan \frac{y_{tmp} + Q_y - y_{source}}{x_{tmp} - x_{source}}$
      if $x_{source} - x_{tmp} == 0$ **then**
        $\beta_{horizontal} \leftarrow \frac{\pi}{2}$
      $\Delta_{horizontal} \leftarrow |\alpha - \beta_{horizontal}|$
      $\Delta_{diagonal} \leftarrow |\alpha - \beta_{diagonal}|$
      $\Delta_{vertical} \leftarrow |\alpha - \beta_{vertical}|$
      if $\Delta_{diagonal} \leq \Delta_{horizontal} \wedge \Delta_{diagonal} \leq \Delta_{vertical}$ **then**
        $x_{tmp} \leftarrow x_{tmp} + Q_x$
        $y_{tmp} \leftarrow x_{tmp} + Q_y$
      else if $\Delta_{horizontal} < \Delta_{diagonal} \wedge \Delta_{horizontal} < \Delta_{vertical}$ **then**
        $y_{tmp} \leftarrow y_{tmp} + Q_y$
      else
        $x_{tmp} \leftarrow x_{tmp} + Q_x$
      $path \leftarrow path + [x_{tmp}, y_{tmp}]$ ▷ *index coordinates*
  return path

interest.

$$I = I_0 \cdot (\mu_{air})^a \cdot (\mu_{concrete})^c \tag{4.9}$$

Upon investigation, we found a linear correlation between the distance between two objects and the strength of their relationship, such as the weight of a recommendation (cf. Section 5.2). Given this observation, we modified the original equation, denoted by Equation 4.9, to better reflect this linear relationship. The adapted equation is presented in Equation 4.10. This adjustment aims to more accurately capture the dynamics we observed between object distance and relationship strength in the application. In general, the composition algorithm presents a multitude of opportunities for modification, accommodating various conceptualizations of distance, radiation, and related parameters.

$$I = I_0 - \min\{\mu_{air} \cdot a + \mu_{concrete} \cdot c, I_0\} \tag{4.10}$$

At this point, the algorithm is tasked with combining the separate maps into a single resulting map, denoted R_i, for each recommendation that will ultimately be composed. As in the VIBE composition, a metric called M is calculated for every cell in R_i. It depends on the field strength I_k of the heat map – with user concept k as emitter – in the very same cell, and the weight w_k the recommendation has with respect to that concept k. Simply put, the calculation made in Equation 4.11 determines the mean squared error of the field strength in a cell and the weight of a recommendation. These consolidated maps can also be viewed as heat maps, where cells representing optimal positions for the corresponding recommendation have higher values compared to cells that indicate unsuitable locations.

$$M = 1 - \frac{\sum_{k=1}^{n} (I_k - w_i)^2}{n} \tag{4.11}$$

Figure 4.9 depicts the same context and the radiated field of two different user concepts. The images show a subexponential decay of the field strength as in Equation 4.9, with $\mu_{air} = 0.9$ and $\mu_{concrete} = 0.5$. Such a map is calculated for all four user concepts. Figure 4.10 shows the M metric values of the cells for a particular recommendation. The color scheme employed here is consistent with that used in the preceding image. In this scheme, red cells indicate areas where M is close to 1, making them strong candidates for placing the recommendation. This 'heat map' is superimposed by the context (cf. Figure 4.9); the uppermost black, unfilled rectangle represents the final position chosen for the recommendation. The black boxes, which signify the user concepts, do not precisely align with the original squares because of the grid's discrete rasterization.

In the final stage, all recommendations are assigned their respective positions. As with the VIBE composition, this procedure begins with the recommendation most relevant to the overall context, based on the mean weight. This approach minimizes the number of compromises needed to avoid overlap, at least for the first few recommendations. The size of a recommendation, or more specifically, the area where the recommendation should be displayed, must be determined either by the user or by another algorithm. The heat map composition itself does not address this issue. The required area is then translated into the number of cells needed to accommodate the recommendation, as already described in Algorithm 4.3 (lines 19–22). To prevent overlapping, the composition removes those cells that are already accommodated by other recommendations of higher priority. In the case that there are no more cells left, the composition simply allows overlapping and reuses already taken cells.

Figure 4.11 showcases how the composition reacts to different situations. In Figure 4.11a, all recommendations share the same weight of 0.5 to the user concepts in red. It is easy to recognize the grid pattern and the regular distribution to avoid overlapping between objects. In

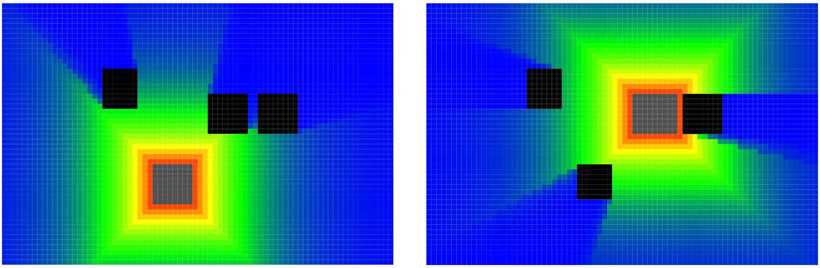


Figure 4.9: Field strength I radiated from the gray object, red is 1.0, yellow 0.66, green 0.33 and blue 0.0

contrast, the situation in Figure 4.11b shows recommendations with random[6] weights. Although the grid pattern is still recognizable to some extent, the composition is less regular, because more cells get the 'chance' to accommodate a recommendation. Minor overlapping appears because the composition did not find any remaining cells without. Both images are based on a composition with a resolution of 91 rows and 84 columns, as $f_{res} = 0.17$ and the width/height of the user concepts is 50. Because the resolution factor f_{res} directly impacts the number of cells that, among other things, are used to measure distance; μ_{air} and $\mu_{concrete}$ have to be adapted accordingly. In case of the shown images, $\mu_{air} = 0.9$ and $\mu_{concrete} = 0.5$; total I is calculated as in Equation 4.9.

Conclusion

In conclusion, the heat map composition serves as a robust and reliable method of arranging elements in a spatial layout. It has improved space utilization and offers flexibility in terms of measuring distances and applying various forms of radiation attenuation. This adaptability makes it a versatile tool for addressing challenges in spatial composition. Compared to circular and VIBE compositions, recommendations

[6] uniform distribution

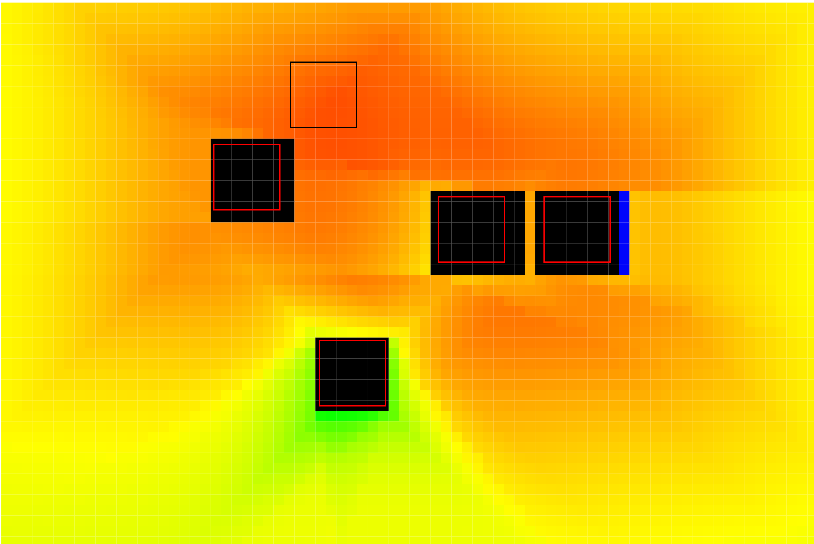
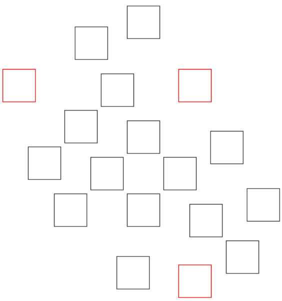
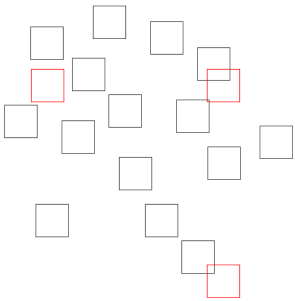


Figure 4.10: Consolidated field metric M for a recommendation with the weights $0.86, 0.04, 0.53, 0.72$ (from left to right user concept)



(a) identical weight



(b) random weight

Figure 4.11: Heat map composition with 15 recommendations and 3 user concepts

are well integrated into the context of user concepts, and overlapping is mostly solved, as long as there are not too many recommendations to be composed. Furthermore, the algorithm supports a direct mapping between weight and distance, aligning well with the insights gathered from our user studies, as discussed in Section 5.2.4. Nevertheless, some issues remain, mainly those which become apparent when the composition is used in an interactive setting and has to react to user input. These are similar to the issues mentioned in Section 4.2.3

Context Updates The issue of a missing transition strategy in case of updated/changed recommendation remains. This happens if e.g. the user adds a new concept to the context or merges two groups of user concepts into one larger context. Yet, what is even more important is the algorithm's stability when the positions of user concepts are altered due to user interactions. The VIBE composition faced challenges due to the somewhat erratic optimization of M which implied position changes, unpredictable in direction and distance. To address this, the heat map composition was designed to eliminate the need for complex calculations in a continuous field. Instead, it adopts a discrete field approach, which is sufficiently small to allow for a straightforward search for the optimal configuration. This leads to two issues.

1. Clearly, a higher resolution is advantageous as it produces cells with more precise field strength values, thereby leaving additional candidate cells available for other recommendations. But it must be considered that the whole algorithm needs to run quite often; in an interactive system it would make sense to run the composition whenever the screen is re-rendered. A standard frame rate is around 60 Hz, theoretically allowing approximately 16 ms for the algorithm to execute. This time constraint does not account for multiple context runs or the computational time needed for other tasks in the application. As computational power diverges

between different devices and the number of cells also depends on the size and number of user concepts, it is not possible to name a value for f_{res} which 'just works'. Yet our experiments show that on an 'average device'[7], a cell size, guaranteeing an acceptable runtime, exceeds by far the size of a pixel. As an example, the composition in Figure 4.11a can be achieved 17 times a second with the settings already mentioned – and a cell size of 8 pixels. However, it becomes apparent that if a recommendation is assigned to another cell, this change will be recognized as a small teleportation, which can occur 60 times a second under bad conditions. In other words: anything but smooth, even if the algorithm becomes optimized, it would reach its limits in a larger context.

2. As mentioned above, if the cell size is large enough to be recognized by a human observer, this will result in unstable compositions during interactive scenarios. Additionally, the composition encounters challenges due to the conversion of continuous weight values into discrete cell locations for the recommendations. This can result in unstable placements where a recommendation oscillates between two adjacent cell boundaries as the context evolves.

Figure 4.12 shows the effect when the user slightly alters the context by moving one user concept. Most of the recommendations are rather stable, which is the goal of the composition. But because of how the algorithm works, those position changes are instantaneous and happen when the algorithm decides that a recommendation belongs to another cell of the field. When closely looking at the image in the middle, the grid structure backing the composition becomes apparent.

[7] This is not an average based on rigorous scientific or statistical methods. The tests were carried out on a Lenovo Thinkpad T470 equipped with an i5-7200U processor with 2 physical cores, each running at 2.50 GHz. This configuration is indicative of consumer-grade hardware available in 2023.

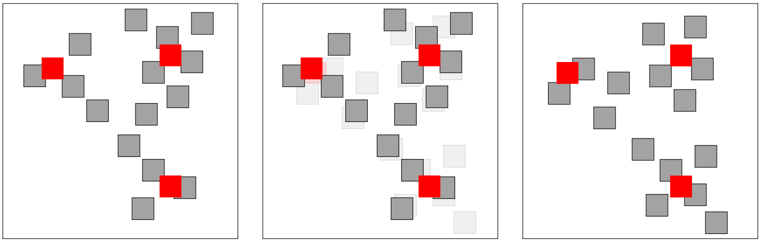


Figure 4.12: Heat map composition: left and right images show slightly different contexts, the middle image does overlay both versions

Definition of Context Until now, we defined the context as the result of user interactions, such as the addition of user concepts, moving and organizing them in space; basically, what is covered by specialized spatial parsers (cf. Section 3.2.2). But what happens with recommendations? If they are, as a result of the composition process, part of the spatial hypertext, they eventually become part of the context. This argument becomes more clear if one is thinking about real-world applications that need to answer the question of how to represent a recommendation in the space. It is not necessarily easy or possible to distinguish between concepts added by the user and recommendations rendered by the system, but other than giving them a different color/shape/size, they are more or less the same. For instance, imagine two user concepts that are spaced far enough apart that neither a spatial parser nor a human observer would consider them part of the same context. Now, introduce recommendations positioned directly in a line between these two concepts. What would be the result? These two previously unrelated concepts would suddenly be interpreted as belonging to the same list, and consequently, the same context.

If recommendations are part of the context, they need to influence each other in their behavior and position. Unlike the circular and VIBE composition, the heat map algorithm could be extended so that the recommendations radiate their own field, which influences the M values of the consolidated field. However, as previously demonstrated, accommodating this more complex context would be computationally intensive. Therefore, alternative solutions that offer efficient runtime performance and ideally operate on a continuous scale must be explored.

4.2.5 Physics-Based Composition

This section builds on work previously published in [107], which has been further extended and adapted for the purposes of this thesis.

Working with continuous values and optimizing multidimensional functions is a mathematically and computationally difficult task. At the same time, a composition algorithm should be able to handle the aforementioned issues of our first experimental implementations; smooth transitions proved to be the most difficult problem to solve while considering the now re-defined, more complex context.

Many algorithms for graph visualization (cf. Section 3.3.3) tackle related challenges employing a physics-based metaphor. Rather than investing in intricate algorithms, these tools create a simplified simulated physical environment. The aim is not to achieve high precision or to cover every nuance of physical mechanics. Instead, visual objects are assigned attributes such as density, velocity, friction, and size, which govern their interactions within the simulation. Repulsive forces can help avoid overlapping, while attractive forces can maintain cohesion between objects. Such physics-inspired metaphors capitalize on the intuitive understanding we have of our physical world to make visualizations more comprehensible and effective for a variety of tasks.

Furthermore, instead of going the error-prone path of implementing solvers for linear and differential (some without known solutions) equations, it is possible to use one of many available physics engines available. While this eases the implementation, the choice to use third-party libraries comes with the price of dependencies. For instance, some of the sample compositions displayed in this section can only be replicated using the same library, the same version, and the same CPU architecture.

Our objective was to refine the concept of heat map composition by mapping the weight of recommendations with specific distances and by identifying optimal compromises when multiple user concepts come into play. A proper metaphor, which is supported by most physics engines, is one with attraction and repulsion, e.g. employed by a spring [124, 69]. For spatial hypertext, composition and parsing happen in a 2D space and objects (e.g. user concepts and recommendations) are usually rigid, which simplifies the process of physics simulation. Possible engines supporting this kind of metaphor are *Brax*[47] (used for research and development of robotics, human perception, materials science, reinforcement learning), *Box2D*[24] (mostly used for games), *Matter.js*[83] (mostly used for games), or *JoltPhysics*[64] (mostly used for games and VR), among many others.

Eventually, Box2D was chosen as physics engine, thanks to its detailed documentation and broad community support, which offers a wealth of questions, answers, tips, and tricks. Furthermore, it is supported on various platforms, most and foremost because of its source code being written in C++, and it is cross-compiled or re-written on platforms that do not support binary code out of the box, like Web browsers. For example *Planck.js* is a re-written version of Box2D for Javascript which maintains the same API and functionality[8].

About Box2D Before going into the details of the physics-based composition algorithm, it is necessary to describe the basic functionality of Box2D, because the composition is built around it. A detailed docu-

[8]Note that floating point arithmetic may vary to some extent, but the effect is too small to influence the results and algorithms explained in this section.

mentation can be found on the Box2D Web page[9] – quoted sentences refer to this documentation.

First, it is important to remember that Box2D does work with so-called *rigid bodies*. “A chunk of matter that is so strong that the distance between any two bits of matter on the chunk is constant. They are hard like a diamond”, which mostly means that a body in Box2D cannot be deformed. The ‘form’ of an object is built upon one or multiple *shapes*, like rectangles, circles, or any other polygon, which are in a fixed relation to each other. Consequently, a *fixture* “binds a shape to a body and adds material properties such as density, friction, and restitution. A fixture puts a shape into the collision system [...] so that it can collide with other shapes”.

Joints are an important concept, necessary to implement the aforementioned metaphor. A joint, in general, describes a constraint between two bodies. This constraint is maintained by the engine and is usually used to hold the two bodies together in a certain way, depending on the joint type. Box2D implements e.g. revolute, prismatic, distance, and other joint types.

Box2D employs an iterative solver that is highly efficient and operates with computational complexity $\mathcal{O}(N)$, where N means the total number of constraints within the simulation. This efficiency ensures that the solver can quickly resolve constraints, making it scalable for a wide range of applications, and promises to solve the original issue of optimizing the spatial composition. To mitigate potential issues such as tunneling that can arise from solving constraints in discrete time steps, Box2D employs motion interpolation techniques. This approach helps to determine the time at which collisions occur, allowing accurate and appropriate collision resolution.

The solver works with floating-point numbers and is optimized to work well with moving shapes with a size in the range of 0.1 to 10. Box2D uses meters-kilograms-seconds (MKS) units and is thus some-

[9] `https://box2d.org/documentation/`

what limited in its capabilities to simulate physics. In the context of our composition algorithm, determining an appropriate scaling factor to convert pixels to meters, along with identifying good values for the physical properties of the fixtures, poses a significant challenge.

Algorithm

The composition is built upon the idea of connecting user concepts and recommendations with springs, similar to what is done by many graph-drawing algorithms (cf. Section 3.3.3). In a relaxed state, a spring does not possess potential energy and therefore exerts no force on the objects at its respective ends, which are effectively connected to those ends. Whenever the spring is extended or compressed from its relaxed state, the potential energy increases, following Hooke's law, proportionally to the extension. Effectively exerting a force on the connected objects such that the spring extension converges to its relaxed state again. When all visual elements in the space are interlinked, the objects self-organize during the simulation. This ultimately leads to a stable condition once the system reaches a damped state.

Box2D offers the so-called *Distance Joint*, a constraint that can be used to keep the distance between two bodies constant. Softness, as with springs, can be achieved by setting the stiffness and damping to desired values (in Newton). The API advertises indirectly setting these values by specifying a *frequency* in Hertz and a dimensionless damping factor generally between 0 and 1. This frequency setting controls the harmonic oscillation of a soft Distance Joint, while the damping factor controls the rate at which the oscillations are damped out. A damping factor represented as $\zeta = 1$ results in a critically damped oscillation. When $\zeta > 1$, the system is overdamped; for $0 < \zeta < 1$, it is underdamped; and with $\zeta = 0$, the system does not experience damping.

Before exploring the effects of these values on the composition algorithm, it is essential to specify the relaxed state of the springs in question. Specifically, it is important to define how the weights affect the length of a spring when it is in this relaxed state. Thus, a mapping function is needed. Influential for this mapping function is the finding of a linear correlation between perceived weight (by the user) and weight (cf. Section 5.2.4, and the perception of proximity depends on the given spatial context (cf. [49]). The 'larger' two objects are, the greater the allowable distance between them while still being considered 'close' to each other. Therefore, the mapping function respects the *size* of objects.

In the context of the composition algorithm, the perception of size is primarily numerical, rather than psychological. Although factors such as *size-distance invariance hypothesis* (SDIH) [84] could add layers of complexity to this understanding, they are beyond the scope of this work. Instead, the algorithm focuses on mapping weights to distances between objects, incorporating the size of objects as an additional parameter. This approach simplifies computational requirements, while still providing a model that represents spatial relationships in a way that is intuitively understandable. Therefore, we specify the size of an object using the longest possible line that can be drawn across either of its dimensions. In the case of non-rotated rectangles, this would refer to either the length or the height, depending on which is greater. This is inspired by *Qill*[10], "a web-based development environment that enables various stakeholders of a web application to collaboratively adopt a model based design of the user interface for cross-platform deployment" [48]. It employs a similar method to measure the size of layout components. In Figure 4.13 all the important variables for the mapping function are illustrated. A factor ϕ is introduced as

$$\phi = -w \cdot (d_{max} - d_{min}) + d_{max} \tag{4.12}$$

where d_{max} refers to the distance assigned a weight of 0 and d_{min}

[10]Online prototype available at: `https://www.w3.org/2012/quill/`

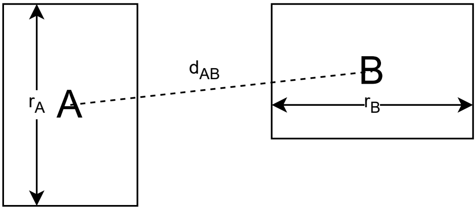


Figure 4.13: The relaxed spring length d_{AB} between A and B is calculated with the help of the height or length of a body, whichever is bigger (r_A and r_B)

a weight of 1; w is the weight of a particular relationship. The result is multiplied by the sum of r_A and r_B, giving us an absolute value for the relaxed length of the underlying spring model. Because this spring implementation connects the centers of both objects, d_{AB} is calculated as follows:

$$d_{AB} = \phi \cdot (r_A + r_B) + \frac{r_A + r_B}{2} \tag{4.13}$$

A sample setup, along with some values chosen for w (weight), is shown in Figure 4.14. It shows how w influences the distance between two given objects in one dimension. The values for d_{min} and d_{max} are not calculated but are chosen to produce a fitting image. Although it was created in a different context, Figure 5.9 shows the result of the applied spring model for the values $d_{min} = 0$ and $d_{max} = 2.8$.

In the next step, the composition algorithm sets up the components of the simulated world, preparing them for execution via Box2D. Initially, the *world* is configured not to exert any gravitational force on its bodies. In addition, user concepts are assigned to a special body type within Box2D, called *static bodies*. Unlike *dynamic bodies*, *static bodies* remain stationary during the simulation, effectively acting as if they possess infinite mass and zero velocity. Although they can be manually moved by the user, they are designed not to engage in colli-

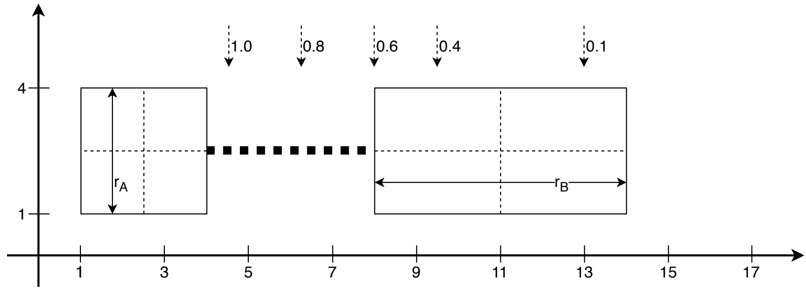


Figure 4.14: Sample mapping for $d_{min} = 0.133$, $d_{max} = 1.2$, and $w = 0.6$

sions with other static bodies within the same simulation environment. This makes them a perfect fit for user concepts, which should only be controlled by user interactions. The recommendation (areas), again modeled as rectangles, are simulated as dynamic bodies, hence do have mass, friction, restitution, velocity, and may collide with all other types of bodies.

This presents a significant obstacle when using physics simulation for composition tasks. Apart from the relaxed states of the springs and the dimensions of the bodies involved, none of the required properties has a clear, tangible equivalent that aligns with the objectives of composition. Quite the contrary, they mostly just do not matter if an observer is only interested in the final result, in sense of the moment, when the simulation damped all oscillations to a stable state. However, our goal is to develop an interactive system that responds to context updates with fluid and easily traceable adjustments. This focus on the dynamic behavior of the composition is something that previous composition experiments did not sufficiently address.

Given the multitude of variables that affect the outcome, it is not feasible to offer universal recommendations for optimal settings. Factors such as the frequency of discrete time-step calculations, solver iteration limits, and others extend beyond the realm of the physical properties previously discussed. Therefore, we give an overview of the properties which can be set via the Box2D API and explain how they influence the composition, how they are related to each other, and what our experience is in setting those values. In the end, we explain which values we used for the example images and user studies carried out later on.

World The environment in which the simulation is happening. Additionally, we consider variables which are outside of the simulation itself as part of this world section.

Gravity A uniform force that acts on all dynamic bodies in the same direction. In the context of a spatial composition, this property should be set to $\vec{0}$, because otherwise the recommendations would tend to float in the direction of the gravity vector.

Other Forces While gravity is a uniform force, it is possible to create custom forces in any direction on the bodies of wish. This can, for example, be used to separate recommendations belonging to different contexts (groups of user concepts spatial parser considers as belonging to the same group), by applying repulsive forces on them. They are completely maintained by the developer which gives a lot of freedom but makes their usage error prone.

Camera Factor This is an important property because it is crucial to convert pixel-sized objects to MKS units of the simulation. First, to obtain accurate results (note that Box2D and similar engines are tuned to work well in certain value ranges[11]) in good runtime. Second, because mass, as explained later, is derived from the density of a shape and its size, which in turn is influenced by the camera factor. In the subsequent phase, mass affects the

[11] Box2D specifies limitations concerning the dimensions of bodies and their distance from the world coordinate system's origin. Users may need to adjust the scale and position of bodies according to their specific objectives. Comparable constraints are common in many other physics engines.

momentum of moving entities; furthermore, it also influences the perception of velocity.

Time Step Box2D solvers work in discrete time steps; how short they are (considering that we want to have a real-time simulation) influences the computational resources needed to calculate the simulation. Apart from that, it is necessary to execute the simulation loop a minimum of 60 times per second. This frequency is generally sufficient to generate smooth object movements that are perceptible to human viewers, and it aligns well with the refresh rates of most modern monitors. The documentation recommends as lower limit "a time step no larger than 1/30 seconds". Additionally, certain properties – or more precisely, the computations that determine their effects – may lead to imprecise results due to phenomena such as aliasing. For example, the frequency of a Distance Joint must be less than half of the time step frequency to avoid such issues. This limitation is consistent with the Nyquist rate, which describes the minimum rate at which a signal can be sampled without introducing aliasing errors [117].

Solver Iterations Every discrete time step calculates the forces acting at a particular instant. The iteration count controls how many times the constraint solver sweeps over all the contacts and joints in the world. In general, Box2D recommends trading short time steps over the iteration count for more accurate physics.

Body These properties are related to the body and its fixture. They mostly determine how bodies react to applied forces.

Shape(s) The shape(s) of a body are necessary to identify collisions with other bodies and to represent the size of a body in the space. A shape in the simulation does not necessarily have to correspond precisely to its 'real-world' counterpart. For example, areas designated for recommendations, which are rectangular in the real-world application, could be modeled as circles or rectangles with extended dimensions (i.e. introducing a margin). Adopting such approximations could achieve specific goals, like maintaining a minimal separation between objects or diminishing the computational burden.

Density Together with the area of its shape(s), Box2D uses the density property to calculate the mass of a body. Mass plays an important role in the simulation, as it significantly influences inertia and momentum. In an interactive system, inertia is perceptible, for instance, through the speed at which bodies respond to changes in velocity (due to applied forces). If the composition uses varying densities e.g. to make more important recommendations 'heavier', this also influences the effects of joints that connect the bodies. Consider, for instance, a spring that is connected to a very heavy body and one very light body. The oscillations will be primarily noticeable on the lighter body. Allowing key recommendations to move at a slower pace could potentially enhance the user's ability to track position changes, which may, in turn, be crucial for user recognition.

Linear Damping When bodies collide, they pass momentum to each other, resulting in a change of velocity in a certain direction. Without linear damping, the body would move forever. For spatial composition, a high damping value would make sense because, otherwise, collisions would lead to a 'ping-pong-like' effect, resulting in a lot of unwanted movement.

Angular Damping Angular damping regulates the reduction of body rotation. In Box2D, bodies can have a fixed rotation, which is often the preferred setting when dealing with non-rotatable concepts and recommendations. Therefore, this feature may not be essential. However, if rotations among bodies are a desired attribute of the composition, it would likely be advisable to select a high angular damping value, similar to the setting used for linear damping.

Friction As linear damping, this property is used to reduce the velocity of bodies when they are in contact with each other. In the case of our desired composition, this situation should not occur (long enough) to have any effect on the resulting behavior of the composition.

Restitution A value between 0 and 1 determining how "bouncy" collisions are. A restitution of 0 simulates non-elastic collision, where the kinetic energy stays the same. A value of 1 consumes all the energy involved in the collision, resulting in no bounce at all. Simply put, a value between is damping collisions. In the context of the composition algorithm, this attribute may not be very significant. The previously established properties aim to minimize the number of collisions; their role primarily serves as a final measure to prevent overlap of recommendations.

As already mentioned, collisions are a crucial part of the simulation, because they have, in addition to preventing overlap, some important pitfalls: Whenever a new recommendation (or user concept) joins the context, a new body has to be created in the simulation. This body needs to have an initial position from which the ongoing simulation can calculate the forces and their direction. Imagine a situation in which a new recommendation is not placed in its desired stable position. As a consequence, the connected joints exert a force in this specific direction, ultimately causing the recommendation to move. While traveling to its stable position, this recommendation could intersect with others. If collision detection is enabled, it is possible that this new recommendation gets obstructed if the applied force is not sufficient to pass. Early tests showed that this is not a theoretical, but a very common situation.

Controlling Collision This implies that in addition to the properties previously mentioned, the simulation requires a strategy for managing collisions. Box2D, for example, allows one to define so-called collision groups, where only bodies belonging to the same group can collide. Additionally, bodies connected by a joint can be forced to (not) collide. Depending on the density of recommendations within the space, collision management must be approached with caution. For instance, applying repulsive forces when two bodies get too close could also eliminate or minimize overlap.

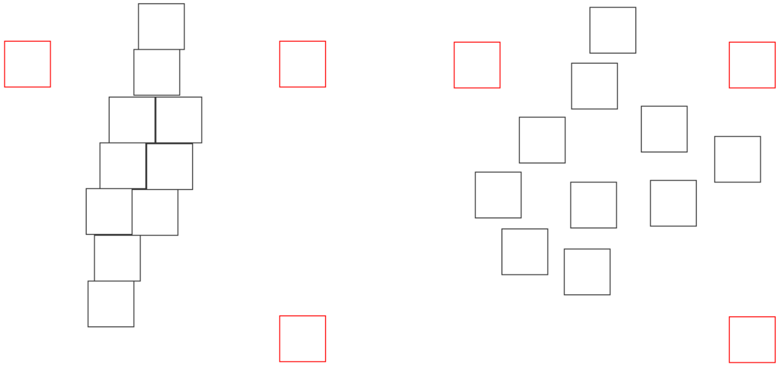
Given that those properties are set to decent values, implementing the composition algorithm itself is quite easy because the engine takes care of computing the resulting velocities and positions. From a developer's perspective, it is important to ensure that the simulation results are accurately reflected in the application and vice versa. Special attention must be paid to interactions that deviate from 'realistic' physics, such as the abrupt appearance of bodies or forces, or user-induced body movements via mouse pointer or touch gestures.

Table 4.1 showcases the values, used in the demonstrator and test software that features the physics-based composition. As previously noted, these values may not be universally applicable and are grounded in personal experience and qualitative feedback from project partners and co-workers. Fundamentally, the entire composition serves as a form of multi-dimensional scaling, which is expected to function regardless of the specific numerical settings. The impact of these settings becomes significant when considering the dynamic behavior of the composition, as explained in Section 4.2.1.

Recommendations as Context As detailed in Section 4.2.4, recommendations should be considered as part of the context. Right now, the composition does only include Distance Joints for relations between user concepts and recommendations and some minor repulsive forces between recommendations; the latter just to reduce the computational

Table 4.1: Parameter settings and descriptions for the physics simulation

	Value	**Comment**
Gravity	$\vec{0}$	No gravity
Other Forces	$\frac{3000\,\mathrm{N}}{d_{AB}^{1.5}}$	Computed on all recommendation pairs (AB) with repulsive direction (cf. Equation 4.13)
Camera Factor	50	Aim at having bodies with no dimension longer than 10, or less than 1 meter (converting from CSS units)
Time Step	$\frac{1}{60}$ s	Matching most screen refresh rates
Solver Iterations	1	High accuracy not important; runtime efficiency
Shape	x	Exact match of "real-world" objects
Density	0	Results in a fixed mass of $1kg$
Linear Damping	0.9	High value to quickly settle in a stable state
Angular Damping	x	Bodies are set to have a fixed rotation
Friction	0	
Restitution	0.9	
Collision	yes/no	Recommendations collide with each other when they belong to the same context
Joint Damping	0.7	Prevent too much oscillation
Joint Frequency	$2w + f_{min}$	Low frequency to prevent fast movement; make springs "stronger" for high weights
Frequency f_{min}	0.1	Minimum frequency for Distance Joints



(a) without Distance Joints (b) with Distance Joints

Figure 4.15: Physics-based composition with 10 recommendations (identical weights) and 3 user concepts

burden of resolving too many collisions. Figure 4.15a displays the composition, already in a damped state, with ten recommendations. Each recommendation has an identical weight of 0.5, both in relation to the concepts of the user and to each other. The consequence of this setup is that collision becomes the only mechanism that prevents the overlap of all recommendations at the exact same location. Additionally, the Distance Joints function much like tensioned springs, exerting significant forces on the recommendations. A minor alteration in the context could result in these recommendations 'snapping' to new positions due to these high forces. If the original claim is taken at face value, one might interpret the composition as suggesting that all recommendations are closely related to each other, which is not necessarily the case. Therefore, we incorporated Distance Joints between the recommendations. While they are similar to user concepts in contributing to the context, the key difference lies in their nonstationary nature.

Figure 4.15b shows the same situation, but with Distance Joints between all the recommendations. For convenience, we assigned a weight of 0.5 to these Distance Joints. Naturally, in a real-world scenario,

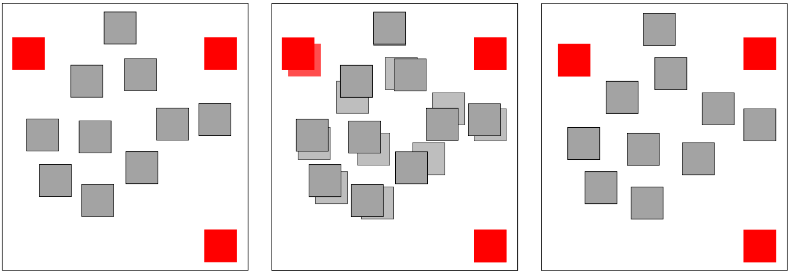


Figure 4.16: Physics-based composition: left and right images show slightly different contexts, the middle image does overlay both versions

these weights would cover a wider range, as they are retrieved from the knowledge base. The properties of the joint are identical to those used for user concepts.

Smooth behavior in case of context updates is a major goal of the physics-based composition. Figure 4.16 showcases two slightly different setups of the same user concepts. For convenience, the relevant weights are set to 0.5. It becomes apparent that the minor movement of the upper left user concept towards the center results in similar minor movements of the recommendations. They occur smooth and steady as the damped-physics simulation reacts to the updated context. This contrasts with the responsiveness of e.g. the heat map composition, where the discrete field leads to teleporting recommendations. It is important that the physics simulation (and the rendered context, obviously) is refreshed/recalculated at a suitable rate per second (cf. Table 4.1).

Meta Context The context is defined by the user concepts, which build a separate, visual group; such that it is clearly distinguishable from other contexts by the human user and the spatial parsers. Recommendations belong to exactly one context to ease the composition and the knowledge base query. But, as already argued, recommendations augment the context, consequently altering it to something new. In practical terms, this could result in scenarios where previously distinguishable contexts become indistinguishable. For example, the recommendations from both contexts might fill in the visual gap that used to separate them. Therefore, the composition needs a strategy to reflect this effect on the *meta context*.

Simply put, a composition must ensure that the context remains the same, so that a user is able to identify the same user concepts belonging to a context, with or without rendered recommendations. This does not imply that the e.g. visual gaps should remain unfilled with recommendations. For example, in the case of a circular composition (cf. Section 4.2.2), recommendations create an easily identifiable circle around the user concepts provided by the context, preserving clarity for the user. A similar effect is applicable by the VIBE composition (cf. Section 4.2.3). The heat map composition, much like the physics-based composition described here, lacks this kind of clarity.

Therefore, the visual appearance of the recommendations must be strictly fit to the context. As we focus on proximity as the main indicator (e.g. colors could also be used), visual gaps must be maintained. Fortunately, general repulsive forces, as detailed in Table 4.1 almost do the trick. Given that the repulsive force between recommendations is the only interaction they have if they belong to different contexts, these recommendations tend to move apart from each other. A simplified example is shown in Figure 4.17. It features two user concepts in red, the visual gap between which results in two separate and easily distinguishable groups, representing two distinct contexts. Each context is augmented with five recommendations, weighted with 0.5 with respect to everything else. Without repulsive forces, the recommendations of both contexts would build a pack, visually interfering with each other. With these forces, as shown, the recommendations move away from each other, such that the visual gap which would be there without recommendations is still clearly visible. It should be noted that

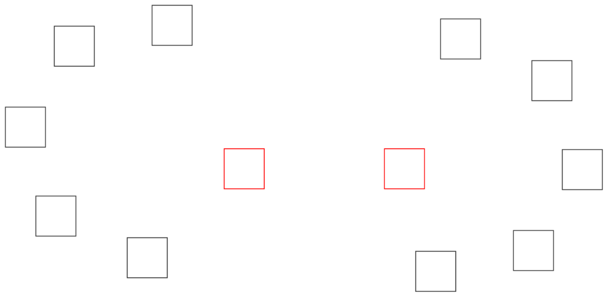


Figure 4.17: Red concepts build two different contexts with five recommendations each; repulsive forces are applied upon the recommendations

there may indeed be scenarios in which these repulsive forces alone are not sufficient to maintain clarity. In such instances, one may need to employ additional visual properties within the composition, although exploring these alternatives falls outside the scope of this thesis.

Conclusion

The physics-based composition picks up the issues we discovered in our early experiments and proves to be an applicable composition method to control recommendations in a spatial hypertext environment (cf. Section 5.3). The primary improvement is the ability to offer dynamic, easily traceable responses to changes in context, while ensuring a good runtime performance – both relative and absolute – on contemporary hardware for the scale typically used in a recommender system. In fact, this section does not outline a detailed algorithm, but aims to offer guidelines for leveraging rigid body physics engines, with a particular focus on Box2D. Recommendations become part of the context and influence the composition such that they are not an abstract additional layer upon the spatial hypertext but are an inherent part of the user's workspace.

In addition, the composition comes with a strategy to maintain the meta context. This is important because real-world applications usually come with many user concepts organized in a potentially complex structure, and hence a multitude of different contexts.

Although the objectives are met within the defined parameters, employing physics simulation comes with its own set of challenges. Often, the focus shifts to devising workarounds tailored to the specialties of the chosen engine, rather than offering a universally applicable solution.

Collisions and Initializing As already mentioned, the simulation may experience problems when, from a 'physics perspective', unusual things occur. This e.g. is the case when the composition introduces new recommendations, because the bodies are 'magically' spawned within the simulation. Forces act abruptly, leading to potentially unexpected changes in the composition from a user perspective. Another issue is the initial placement of new bodies within the simulation. A simplistic approach might be to generate these bodies at the world's origin, requiring them to subsequently move to their stable positions under the influence of the damped springs. This movement may be obstructed or completely prevented by collisions along the way. Sometimes, engines are not able to resolve collisions perfectly; Box2D, for example, has issues with aligned rectangles, which may lead to unexpected and infinite bouncing[12]. Additionally, Box2D struggles with scenarios where two or more bodies overlap perfectly, a situation that could arise if the composition algorithm creates multiple bodies in a single simulation step.

A possible workaround would be to use some heuristic when calculating the initial position. In one of our project prototypes, we initiated new recommendations from the center of a given context. We complemented this by adding a fading animation. This created the illusion that recommendations were emerging away from the center, making it

[12] See `https://youtu.be/BwhVAYSRe8Y?si=Nd4XTU7B4gFjWBoW&t=46` (starting at 46 seconds) as example. The recommended 'Michael Rooker' and 'Laura Haddock' encounter a high-frequency, infinite physics glitch.

evident that they could not collide with recommendations belonging to other contexts. Although this reduces the likelihood of these events, a negative impact on the composition cannot be completely mitigated.

Number of Recommendations Obviously, any composition finds its limit in the number of objects it has to control and lay out meaningfully in a restricted space. Distance/proximity needs free space if it is to be interpreted as a means of weighting between objects. In the physics-based composition, each constraint within a given context exerts a force on the objects, without any compensatory mechanism to account for the quantity of objects involved. This absence of normalization of forces can result in a cumulative attractive force when numerous objects are present. Consequently, these objects are drawn together more strongly, often coalescing into a densely packed cluster or sphere. This clustering effect undermines the model's ability to clearly distinguish between individual objects and their relation to each other, thereby compromising the integrity and interpretability of the simulation.

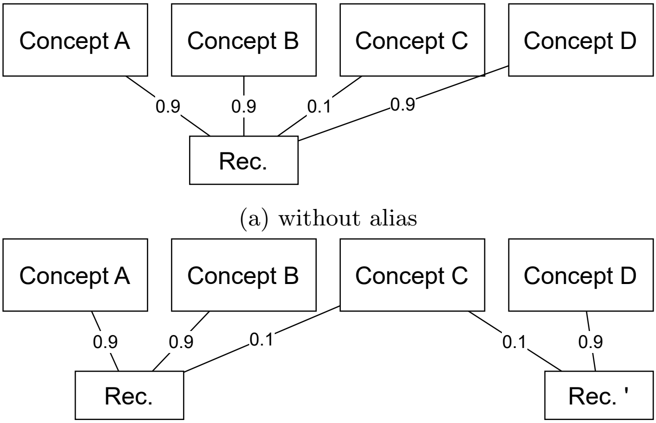
A solution would be to reduce the number of recommendations or the available area, such that stacking forces are prevented. This consequently leads to the question of which number or area is suitable for this composition and compositions in general. Later in this thesis, a detailed user study sheds some light on this question: Users appear to have a cognitive threshold for the number of information items that they process before proceeding with their next action. As such, it may be advantageous to limit the number of recommendations, especially since an overabundance of information is likely to reduce the impact of additional recommendations due to a ceiling effect. In the context of the user study we conducted, this optimal number was identified to be approximately seven, in the specific study setting (cf. Section 5.3). This quantity is sufficiently small to avoid the cumulative attractive forces that otherwise result in densely packed clusters of recommendations.

As the study is limited in its scope, there is room for further research on this topic. There are good reasons to suggest that, considering the graph structure (i.e., the weights that the recommendations hold relative to the rest of the context), there exists an upper limit of the recommendations that should be included in the composition. This limitation takes into account the expressive power of the visual cues being used, considering the given input parameters.

MDS Limitations The graph of recommendations is, in fact, a vector space that is translated into 2D positional information. When multidimensional information is displayed in a space with fewer dimensions, information gets lost. In practice, dimensions are determined by the sum of the number of user concepts and the number of recommendations, minus one to account for the absence of self-reference. Typically, most use cases yield a substantially higher number of dimensions. As is the case with any MDS technique, the objective is to arrive at a composition that is 'good enough'.

In some cases, this does not work because the result cannot be interpreted correctly or, worse, may be misleading. Figure 4.18a shows such a situation. The recommendation has a height weight with respect to concepts A, B, and D. Therefore, the physics composition would employ short springs, transmitting strong attractive forces on the recommendation, although the low weight towards concept C would suggest a higher distance.

One could argue that, in the context of MDS, a less-than-ideal positioning relative to concept C is a trade-off made to maintain meaningful positions in relation to other concepts. However, there are viable solutions to address this type of issue. Refer to Figure 4.18b. In this example, the composition produced an alias representation of the recommendation, which is disassociated from concepts A and B, while the original recommendation remains visible but is disconnected from concept D. In this way, there is simply more room for the original and the alias to adapt their position to the given recommendation graph.



(b) with alias

Figure 4.18: Two images side by side with subfigure

Aliasing is a technique that can be adapted for all composition algorithms. However, it is beyond the scope of this thesis. The implications are discussed in Section 6.3.

4.2.6 Pearl Necklace Composition

Although the physics-based composition satisfied the requirements we set initially, we thought about picking up the remaining pitfalls to implement a different approach. Some of the issues, like collisions and effects of the physics solver, originate in the simulation approach, which prioritizes runtime, because typical engines aim at simulating thousands of bodies and constraints. Furthermore, the entire composition is dependent on complex engine algorithms, which complicates the task of thoroughly understanding and researching the inner mechanics of the composition. This complexity can pose challenges for in-depth analysis and may require specialized expertise to fully analyze the behavior of the composition.

The number of recommendations a composition can – or better, should – control is an overarching issue, affecting not only the physics-based composition, but the former experimental ones as well. The issue arises because compositions lack a metric to gauge the availability of space for positioning recommendations, while also preserving enough white space. This white space is essential to allow users to meaningfully interpret the distances between objects. Other than relying on experimental numbers (which may depend on many factors, probably making it very hard to come up with a general applicable formula), it may be a viable approach to build-in such a metric.

To make this work, the composition process must be simplified. The limitations of MDS show that the composition has to be broken down. Obviously, there are many ways to do that. Our proposed composition here can be aptly likened to a 'pearl necklace metaphor'. In this analogy, the pearls symbolize the recommendations, and the necklace arranges itself around the context. The challenges here are the arrangement of the necklace and the sequence/behavior of the threaded pearls. The available space can be readily understood as the length of the necklace, which is comprised of the diameters of the pearls and the distances between them; this simplifies the calculation of the metric to an effectively one-dimensional problem.

The key takeaway is to streamline the recommendation graph into a circular structure. In this arrangement, each recommendation retains connections only to two close neighbors[13], effectively forming the sequence of pearls on the necklace. The relationships to user concepts can serve to adjust the orientation of the necklace. Picture how the pearls can glide along the string, circulating around the context. The details and considerations are detailed in the next section.

[13] In the best case its two nearest.

Algorithm

Circular List The first challenge is to transform the recommendation graph into a circular structure with the goal of keeping the relations of each recommendation to its two nearest neighbors. Obviously, this only works in very few cases; therefore, we define the stress function s that consumes two arbitrary recommendations R_i and R_j as parameters and calculates a value that indicates to what extent the relationship between both recommendations should be kept.

In theory, many factors (properties of the recommendations) could influence the result of s; our implementation is based solely on the weight $w_{i,j}$ both recommendations have towards each other. As the resulting values of $s : R \times R \to [0,1]$, we define the function as follows.

$$s(R_i, R_j) = 1 - w_{i,j} \tag{4.14}$$

In simpler terms, if two recommendations are closely related to each other, the stress between them is minimal, making it more probable that their relationship will remain in the transformed graph. The algorithm builds this graph with two constraints.

1. Each recommendation has two relations to other recommendations.

2. The mean squared stress $M(R)$ for all recommendation pairs in a circular sequence is kept to the minimum (cf. Equation 4.15).

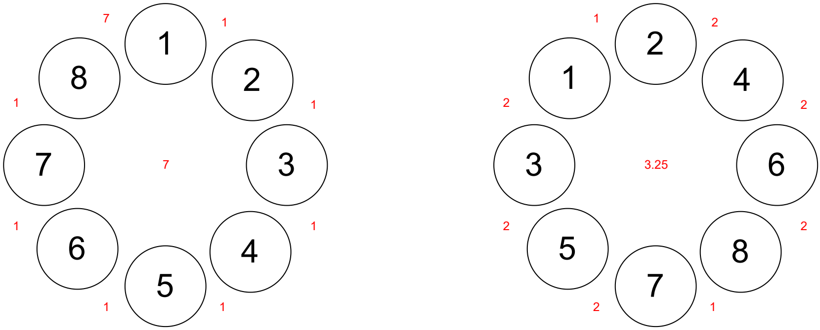
$$M(R) = \frac{1}{n}\left(s(R_1, R_n)^2 + \sum_{i=1}^{n-1} s(R_i, R_{i+1})^2\right) \tag{4.15}$$

Building a circular structure of all recommendations provided would not be sufficient as there may not be enough space on the necklace to compose them all. Therefore, the algorithm must determine the number of recommendations to draw. In alignment with the composition algorithms described before, the general relevance to the whole context, e.g. measured by the mean weight to user concepts, can be used to select the best n recommendations. This task must be carried out by a component capable of calculating the length of the necklace, which then forwards this parameter to the circular list structure.

The pseudocode presented below offers a sample implementation for the required circular list. It is important to note that this code is designed for illustrative purposes to aid the reader in understanding and has a quadratic runtime complexity with respect to n. In a production setting, the actual implementation should incorporate various performance enhancements, such as result caching or maintaining queues for recommendations that need to be added or removed due to changes in necklace length.

Algorithm 4.6 Building an optimized circular list out of a given, sorted list of recommendations

```
Input: recommendation array sorted by relevance recs
       number of selected recommendations n
function CircularList(recs, n)
    result ← [...]                         ▷ initialize empty resulting circular linked list
    for i = 1, ..., n do
        if n ≤ 2 then                      ▷ order does not matter in this case
            result ← result + recs[i]
        else
            s_best ← ∞                     ▷ stress s to beat
            i_best ← −1                    ▷ index of best candidate configuration
            for option = 1, ... result.length do
                result.add(option, rec[i])  ▷ probe temporarily
                s_tmp ← M(result)
                i_tmp ← option
                if s_tmp < s_best then
                    s_best ← s_tmp
                    i_best ← i_tmp
                result.remove(option)       ▷ remove probe
            result.add(i_best, recs[i])
    return result                          ▷ Return the final circular list
```



(a) not optimized: $M(R) = 7$ (b) optimized: $M(R) = 3.25$

Figure 4.19: Two versions of a circular shape with 8 recommendations; weight is determined by the absolute difference between both numbers

The simplified examples in Figure 4.19 show what the circular structure looks like. Both images are based on the same recommendations, labeled with a number $x \in \{1, 2, \ldots, 8\}$; the stress between two recommendations i and j is determined by $|x_i - x_j|$. To clarify the illustration, these numerical labels do not adhere to the prescribed interval for the weight or stress variables s and $M(R)$. However, the underlying principle remains unchanged: the algorithm identifies a configuration that minimizes the mean squared stress, as demonstrated in Figure 4.19b.

It becomes clear that most of the recommendations are not adjacent to their nearest neighbors in the original graph structure. However, they reside in proximity to each other. The theoretical graph of this example exhibits significant "weighted transitivity" [cf. 96]. Transitivity is a property often discussed in the context of social network theory as "triadic closure". In such cases, the existence of relationships (edges) between two individuals (nodes) within a group of three (triad) increases the likelihood of forming the remaining relationship. In the case of weighted graphs, the concept of transitivity can be adapted to account for the varying influence of partially weighted edges. Although they contribute less than fully weighted edges, they still play a role in shaping the graph's overall transitivity. Although there is no assurance that knowledge graphs, as employed in our recommendation system, will demonstrate high transitivity, they are designed to encapsulate semantic affiliations between entities. Such relationships are naturally conducive to higher levels of (weighted) transitivity.

As a key requirement, the circular list structure presented here is most effective when applied to graphs with high transitivity. In the absence of such transitivity, the outcomes could be misleading or nonsensical, as recommendations that lack a meaningful relationship might still be composed in close proximity.

The Necklace In order to compose the recommendations in relation to the context, the algorithm must generate a closed polygon that not only encompasses the user concepts, but also provides sufficient space to thread the 'pearls'. Multiple approaches would work to solve this problem; in fact, the circular composition already did this by computing a bounding rectangle and drawing a circle around it. For the implementation of spatial parsing, Schedel et al. [115], utilized so-called *K-Discrete Oriented Polytopes* (K-DOP) to encapsulate the visual symbols of a spatial hypertext. In a 2D application context, a 2-polytope is essentially a polygon. The term "k-discrete oriented" indicates that, unlike a basic bounding box where $K = 2$, this method can incorporate more orientations, resulting in a more precise representation of the original symbol(s). The concept of a K-DOP is generally designed to create a convex bounding volume. A K-DOP is formed by taking the dot product of a set of vectors with a set of K predefined directions (normals to the faces of the bounding polytope). Due to the mathematical properties of convex sets and the use of these operations, K-DOPs are convex.

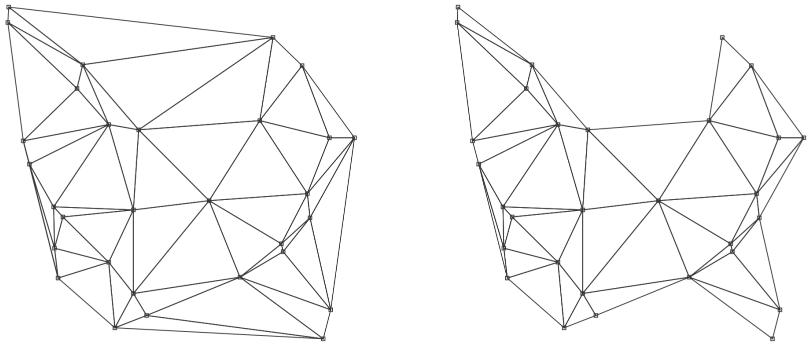
However, convex hulls are inadequate for encapsulating more complex human-designed structures. For example, consider an L-shaped configuration made up of two linear lists that meet at a single point. A convex hull would leave a significant amount of white space within the L-shape. In the end, the same critique would apply as for the circular composition and its poor space utilization. Furthermore, the necklace would, in parts, be ‘far’ away from the context, undermining the idea of proximity to express relation.

Instead, we opted to implement a concave hull, which smoothly encapsulates all kinds of structure and does not have the same limitations as e.g. K-DOPs. There are multiple methods to compute concave hulls, usually employing a value to influence the grade of ‘concavity’. One of the methods is to create a so-called *alpha shape* [40]. Typically, this is done starting with a *Delaunay triangulation* (DT). A DT for a specific set of discrete points p_i, assuming that they are in general position, constitutes a triangulation where no point p_i falls within the circumcircle of any triangle that is part of the DT. In simpler terms, all the points are connected by edges to form non-overlapping triangles, each meeting the condition that no other points fall within its circumcircle. DTs optimize the geometry of a triangulation by maximizing the smallest angle in each triangle [88].

To generate an alpha shape, one needs to determine a specific value for α. The algorithm then takes all triangles from the DT and checks if the radius of their circumcircle ($R = \frac{abc}{4\triangle}$) is less than $\frac{1}{\alpha}$. If this is the case, this triangle will be kept; otherwise, it will be removed. Eventually, the union of all remaining triangles forms the alpha shape. An example is shown in Figure 4.20. It shows a DT of 30 random points on the left and the same DT with triangles removed, which had a circumradius $R > \frac{1}{0.007}$ on the right.

The larger the value α is, the more triangles are removed, resulting in a potentially more concave hull. Therefore, α serves as an indicator of the degree of concavity of the hull. Unfortunately, this somewhat naive approach needs some extensions.

First, if α becomes too small, the union of the remaining triangles



(a) Delaunay triangulation (b) alpha shape with $\alpha = 0.007$

Figure 4.20: Thirty random points displayed along with the corresponding alpha shape and underlying Delaunay triangulation. For reference, the dimensions of the image are 787×755 (bounding box).

might not result in one but many polygons. As we want to use the exterior ring of the polygon as the blueprint for the necklace, it has to be a single polygon. A similar consequence of a large α is that the resulting polygon might have holes. The most straightforward approach is to select an appropriate value for α. Given that the definition of 'appropriate' depends on numerous variables, we chose values that empirically proved effective in our experiments, without offering a universally applicable formula.

Second, we specified that spatial hypertext not only includes points, but also visual symbols, typically in the form of geometric primitives such as rectangles or circles. This is a problem in the context of the DT. If one uses the corners of a rectangle as points to construct the DT, there is no guarantee that the original edges of the rectangle will be preserved as edges in the resulting DT. This can be solved by constructing a *constraint Delaunay triangulation* (CDT) [95] or a *conforming Delaunay triangulation* (RDT) [97]. Both methods are very similar, leading to a conforming triangulation with respect to a given set of edges. A

RDT does this by adding so-called *Steiner points*, points that are not part of the original input but whose addition 'improves' the result of a geometrical problem. In this context, a RDT constrained by polygonal edges that represent user concepts would introduce additional points along these edges to ensure their inclusion in the triangulation. A CDT instead alters the original triangulation until the same constraints are met. Obviously, this method does not work if e.g. rectangles overlap. Here, it is necessary to introduce Steiner points; hence, we opt to utilize RDTs. Both methods show to have a similar runtime of $\mathcal{O}(n \log n)$, with respect to n points.

Third, following the arguments in the previous point, the removal of triangles has to be modified as well. As the necklace is meant to wrap the user concepts, the algorithm must not remove rectangles whose edges are part of a representing polygon. To be more specific, all edges of the triangle should be eliminated, unless they are part of or contained by a user concept. It is important to note that the edges of the triangles may differ from the original polygonal edges, due to the insertion of Steiner points in the RDT. To determine which edges should be retained, the algorithm must verify if a given edge is a component of any original polygon that represents a user concept. All other edges of a triangle are safe to remove, since the circumradius is less than $\frac{1}{\alpha}$.

If all those extensions are applied, we end up with a polygon built from the union of the remaining triangles. The outer ring of that polygon is the first step toward the necklace that we need to thread the 'recommendation pearls'. The recommendations themselves have an area that occupies space. If they were composed directly on the hull, they would overlap with the user concepts. Consequently, it is essential to offset the hull by a specific margin, at least enough to ensure that the recommendations have sufficient space or distance from the user concepts to avoid any overlap.

Offsetting[14] a concave polygon is a non-trivial task; the result de-

[14] Sometimes called *buffer* of a geometry

pends on the definition of an offset and the algorithm which is used. In the context of composition, our primary requirement is runtime efficiency. We do not impose any particular conditions on the offset polygon, except that it should maintain the general shape of the original polygon. Although this is a vague definition, it is an important one. Otherwise, any circle large enough would be a valid offset. A secondary requirement is ease of implementation, as the code should be readily portable to various environments and sufficiently transparent to allow for potential custom modifications, if needed.

Possible solutions include the utilization of *Minkowski addition* or *Voronoi diagrams*[15] [30]. For the pearl-necklace composition, we chose the algorithm put forth by Martinez et al. [82]. This is because it outperforms the widely used algorithms by Greiner and Hormann [51] as well as Vatti [128] in terms of runtime and already has a solid implementation by Milevski et al. and comes with a permissive MIT license [85]. A sample is shown in Figure 4.21e.

The idea behind the necklace is to provide a flexible framework in which recommendations can adaptively align themselves according to the current context. This implies that, much like pearls sliding along a thread, the recommendations glide smoothly along the offset hull. However, the offset hull has corners, some of which form acute angles. These acute angles introduce multiple challenges. First, they result in unnatural behaviors in the movement of recommendations; specifically, recommendations make abrupt directional changes when navigating along such corners. Second, these acute angle corners present another problem related to the perception of distance. Consider two edges that are connected by a right angle (90 degrees), where each edge is a component of the offset hull and hosts a single recommendation. The algorithm calculates the distance on the basis of the actual

[15] Voronoi diagrams are the dual graphs of Delaunay triangulations. Although the latter is already computed, using Voronoi diagrams for offsetting has proven to be unreliable in certain scenarios (e.g. holes in the polygon).

lengths of these invisible edges. In contrast, a human observer perceives the distance more directly, typically interpreting the two recommendations as being closer together than the algorithm would suggest. This discrepancy between algorithmic and human perception can lead to a misinterpretation of the composition. Therefore, the final step aims at rounding off the corners, eventually having a hull that more closely resembles the metaphorical pearl necklace.

Again, multiple solutions would fit the requirement of providing round corners. For example, *quadratic Bézier curves* [cf. 129, p. 238] are a very common parametric curve used in computer graphics to draw a simple curve from one point to another. Instead of calculating points of a polygon that create a curve, Bézier curves are based on a function and parameters. If one has the function and the parameters, it is easy to calculate any point on that curve. A quadratic Bézier curve is described by a common function $B(t)$ with $t \in [0, 1]$ and three points p_0, p_1, p_2. Two points (p_0, p_2) describe from where the curve has to start or where it ends, respectively; the third point indicates how the curve looks like. Parameter t essentially represents the proportion or fraction along the curve where the point is calculated. The nature of Bézier curves presents some challenges with respect to our composition.

The algorithm needs a suitable strategy to pick the necessary points. For the start and end point, a straightforward solution is to compute points on the lines that meet in a corner, e.g. at $\frac{2}{3}$ of the total length. Choosing a suitable p_1 proves to be more challenging and could be calculated using a heuristic that depends on the angle between both lines and the desired 'roundness' of the curves.

Later on, the recommendations on the necklace will have a certain distance from each other. This distance will be measured by the length of the line segments of the hull in between. For Bézier curves, there is no non-trivial solution to calculate (proportional) arc lengths. Additionally, the algorithm would need to employ a special treatment for curves.

Another approach is described by Chaikin – "an algorithm for high-speed curve generation" [26], nowadays often referred to as *Chaikin's refinement* or *Chaikin's corner cutting.* Simply put, the idea is to take all corners and "cut" them by a certain amount, effectively exchanging each point by two new points that describe two less acute angles instead of one more acute angle. Chaikin's refinement can be repeated many times, each iteration leads to more points, more and shorter line segments in between, and thus smoother curves. The necessary computations can be done with integers[16] [26], utilizing simple addition and division (in case the divisor equals 2, a bit shift is sufficient) operations. The runtime scales linearly with the number of points n and exponentially with the number of rounds r, resulting in $\mathcal{O}(2^r n)$. Given that our experiments show suitable results for $2 < r < 5$, the algorithm is fast enough to satisfy our needs. It consumes more memory compared to working with Bézier curves, because it has to handle 8 to 16 times more points than the original hull instead of taking care of n additional control points. However, the distance calculation in the next steps will be eased, because there is no need for a special treatment for curves.

The entire process of computing an offset, well-shaped concave necklace for a given input of user concepts (visual symbols in general) is depicted in Figure 4.21. To illustrate the earlier example of L-shaped user concepts, the figure features seven rectangles (for reference, their size is 60×60) arranged in that pattern (Figure 4.21a). Since this sample is computer generated, a jitter has been introduced to the rectangle positions to ensure that they are not perfectly aligned. In the next step, a conforming DT (RDT) is generated (Figure 4.21b) and serves as the basis for an alpha shape ($\alpha = 0.009$, Figures 4.21c and 4.21d). Since the recommendations require space and should not overlap with user concepts, the polygon that describes the alpha shape is offset, as shown in Figure 4.21e. Finally, Chaikin's refinement is done in four

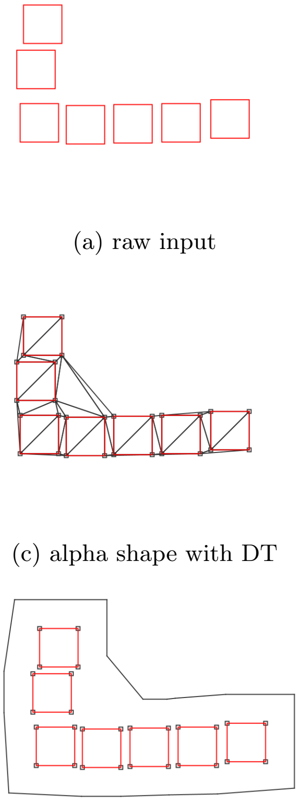
[16] Sub-pixel calculations are not needed, minor rounding-errors do not matter in this context, as the differences are not visible (the necklace will not be drawn).

rounds, resulting in smooth and round corners.

Threading the Pearls In this step of the algorithm, the recommendations will be threaded onto the necklace, using the circular list data structure described above. But beforehand, the composition needs to determine a few things:

1. The necklace represents a (circular) one-dimensional space for the recommendations which can host a certain amount of recommendations. The number of recommendations displayed depends on the length of the necklace, the intended size of the recommendations when rendered, and the amount of space desired between them.
2. The sequence of recommendations is already determined by the circular list, but the orientation of the necklace is not yet defined. Simply put, the composition needs to decide whether, from the perspective of singular recommendation, faces e.g. north-side or somewhere else.
3. As already discussed in the physics-based composition, the dynamic behavior when the context changes is important. For the pearl-necklace composition, logic is necessary to e.g. handle a growing or shrinking necklace. This is discussed in the next section.

The length of the necklace can be readily determined since, due to Chaikin's refinement, it is made up of a sequence of coordinates that form a closed polygon. Consequently, the total length is the sum of all the line segments, calculated as the Euclidean distances between consecutive points in the polygon. The size each recommendation consumes can be determined by the diameter of its circumcircle. For the distance between recommendations, things become more complicated. In principle, a good approach is to utilize Equation 4.13, already defined for the physics-based composition. It computes a distance, based



(a) raw input

(b) conforming DT (RDT)

(c) alpha shape with DT

(d) alpha shape

(e) offset alpha shape

(f) Chaikin's refinement ×4

Figure 4.21: Steps (a—f) to compute the final necklace, used to thread the recommendations

on the size of the recommendations and the strength of the relation they have to each other.

But there are two things to consider. First, the necklace is a) circular and b) will grow and shrink quite often; each time, the user is moving a user concept. Therefore, it might be challenging to maintain the precise desired distance. It could be beneficial to allow for some flexibility, ensuring that a recommendation is not removed merely because the necklace length has decreased slightly.

Second, the composition does not measure distance as in the Euclidean space (which is the intention of the above-mentioned equation). It calculates distance along the necklace. Although the number of acute angles is minimized, the necklace is not purely linear and exhibits various curves. In short, the perceived (Euclidean) distance is usually shorter than what the algorithm determines.

Given these two considerations and to keep the implementation simple, we decided to use a general heuristic distance, assuming that what we previously defined as ideal is not directly applicable. Instead, we opt to standardize the size of all recommendations and use the median weight of all pair of recommendations to determine the heuristic distance. This distance then helps to calculate how many recommendations can be accommodated on the necklace. Eventually, the deviation between the heuristic and the actual can be evenly distributed across the recommendation pairs.

In Equation 4.16, the formula specifies how to determine the number of recommendations that can be accommodated on the necklace P. Here, L represents the length of the necklace. It is divided by the estimated space required for each recommendation, which is a sum of the circumcircle diameter D and the heuristic distance h_{AB} as discussed earlier.

$$P = \lfloor \frac{L}{D + h_{AB}} \rfloor \tag{4.16}$$

Until now, the relationship between user concepts and recommendations has not been deeply elaborated. Currently, they are composed on a concave hull, roughly maintaining a uniform distance from the alpha shape that encompasses the user concepts. The distance between recommendations has already been determined; thus, there are limited possibilities to encode the weight of recommendations by proximity. The only dimension left is the necklace itself: By rotating the recommendations along the path that forms the necklace, the composition can search for a configuration that is suitable for the given context.

Figure 4.22a shows a simplified situation to clarify the effect of rotating the recommendations on the necklace. The context is built by four user concepts A, B, C and D. The dashed circle around represents the necklace that hosts one recommendation in the upper right. The lines between that recommendation and the user concepts denote the weight the recommendation has towards them. Considering that distance is utilized to represent this weight, there exist optimal positions for the recommendations on the necklace that minimize a benchmark value. In other words, these positions should closely resemble the definition of an ideal distance, as described in Equation 4.13. The figure itself features a second dashed position for the recommendation on the left. Given the weights, it becomes apparent that the dashed version of the recommendation is in a poorer position than the other, because it appears closer to the user concept D than A and B which are connected with a higher weight.

In summary, the goal is to move the recommendations along the necklace to best approximate the relationship between weight and distance. As the necklace restricts movement, it does not make sense to reuse the aforementioned Equation 4.13, originally used by the physics-based composition. Although an absolute mapping from weight to distance is the only way to encode these values without any loss to the user, this restriction requires the use of a relative measure, as done by the VIBE composition (cf. Section 4.2.3). The relative positioning there

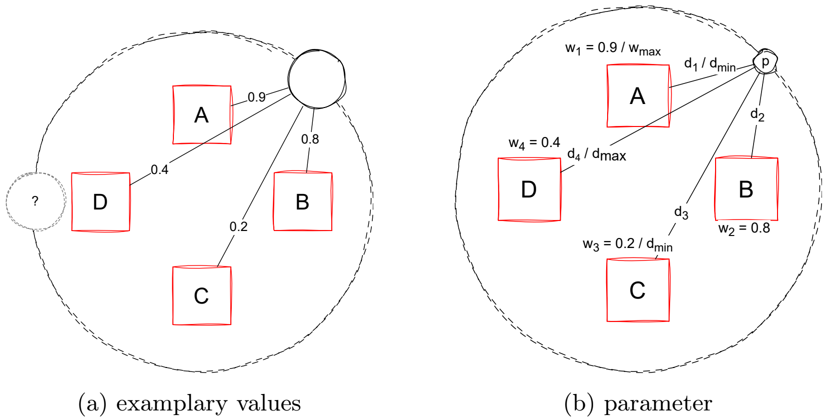


Figure 4.22: Simplified necklace and context to showcase the effect of rotation

cannot be transferred one by one, as the VIBE composition does not face any constraints on the resulting positions. Therefore, we redefine the benchmark value as follows.

Consider a recommendation r and n user concepts within the context. The weights of r towards the user concepts are indicated as $w_1, w_2, \ldots, w_n \in [0, 1]$; we also have $w_{min} = min\{w_1, \ldots, w_n\}$ and $w_{max} = max\{w_1, \ldots, w_n\}$. Then, for any given point p on the necklace[17], we can calculate a value $M(p, r)$ that we aim to minimize. For p, we calculate the (Euclidean) distance to the centers of all user concepts $d_1, d_2, \ldots, d_n$, normalized with $d_{min} = min\{d_1, \ldots, d_n\}$ and $d_{max} = max\{d_1, \ldots, d_n\}$ such that $d \in [0, 1]$. Equation 4.17 details the calculation of $M(p, r)$. Essentially, the function calculates the mean squared deviation by comparing the locally normalized distance with the normalized weight. Figure 4.22b displays the parameters with an example.

[17]The necklace has infinitely many points, representing a continuous shape.

$$M(p,r) = \sum_{i=1}^{n} \left(d_i - (1 - \frac{w_i - w_{min}}{w_{max} - w_{min}}) \right)^2 \tag{4.17}$$

For the heat map composition, we contend that a simplified search for a minimum is ‘good enough’. Instead of employing a more complex method like the downhill simplex approach, we opt to evaluate the function at selected sample points. The heat map was split into a grid, and the necklace can be divided into several segments. As the resolution of the grid is a factor to influence accuracy (and runtime), the same applies for the necklace. The more sample points are evaluated, the better the result; the worse the runtime. Based on this assumption, there are many ways to iterate these points. For the sake of simplicity, we chose to evaluate every point of the hull polygon, as well as an additional midpoint for each line segment. Although this does not offer a uniform distribution based on the necklace’s length between the sample points, it streamlines the implementation and obviates the need for additional calculations.

However, this presents a solution for only one recommendation/pearl. Usually, the composition has to handle more recommendations; and the order is set by the circular list implementation; their distances are controlled in certain boundaries as well. A naive approach would be to calculate $M(p,r)$ for all r on the necklace, and the algorithm determines some sort of average rotation. The concept of “weighted transitivity” now re-enters the stage. If the knowledge graph exhibits low transitivity, an average rotation could likely produce nonsense results, given the limited or nonexistent transitive relationships between neighboring recommendations on the necklace and their corresponding relation to the user concepts. Simply put, the pearl necklace composition takes advantage of and needs knowledge graphs with high weighted transitivity.

Given that the graph exhibits high transitivity, there is no need to perform the aforementioned computation for every recommendation.

Instead, it is feasible to choose specific recommendations and compute the rotation only for those. We propose selecting the top k recommendations, prioritized by their mean weight to user concepts, in line with the other composition algorithms. Next, the algorithm iterates through the sample points and stores the clockwise rotation length from the start of the iteration to the minimum value of $M(p, r)$. The average of these recorded rotation lengths determines the rotation length applied to all recommendations. The implementation used in the SPORE (cf. Section 2.5) research demonstration uses $k = 1$, which results in mostly meaning- and useful compositions.

Responsiveness As in the other compositions, the pearl necklace composition has to run at a certain frequency to provide an interactive user experience. It is important that when the context undergoes minor adjustments, the composed recommendations exhibit minor changes as well, ensuring smooth transitions. Several things may happen in the meantime of a composition step. The necklace may change its shape due to a moved user concept; it may comprise more or less recommendations, due to a change of the necklace's length. We aim to obtain a rendering frequency of 60 Hz, similar to the physics-based composition approach. This means that recommendations that have an established position will immediately change to their new position as determined by the most recent composition step. When this positional change is minor, the update may appear as a seamless transition. But there are cases where this positional change is not minor, e.g. the recommendations have to rotate to the other side of the necklace. The physics simulation addressed two main aspects: First, extended transitions occur gradually because the bodies are influenced by forces rather than being placed directly. Second, as the recommendations move, mechanisms such as collision avoidance ensure that they do not overlap with each other, leading to a relatively orderly transition.

The pearl necklace composition needs a similar transition strategy to guarantee a pleasant dynamic behavior. To achieve this, the composition has to mimic the two aspects mentioned above.

1. Limit/control the speed recommendations are allowed to move
2. Control the movement by forcing the movement on the necklace

To control the speed, the composition has to be divided into two components that run independently of each other. One component is the algorithm as described until now, which builds the necklace, sorts the recommendations, and denotes where they have to be. A second component receives this determined future position of the recommendation. The primary function of this component is to ensure a smooth transition for the recommendation's movement towards its intended destination. To achieve this seamless movement, the component strategically calculates intermediate positions for the recommendation, distributing them across given time frames, effectively limiting the speed of the transition. The result is a graceful and continuous progression from the recommendation's current location to its future position.

These intermediate positions do not have to be in a straight line. To regulate the movement, these intermediate points can be pinned to the latest version of the necklace. In this way, the recommendations move gracefully and in order on the necklace. But the algorithm needs to take care of some special cases, e.g. if a recommendation is not on the necklace (any more), due to an updated necklace; or if rotation is necessary, determine in which direction (the insertion of recommendations may lead to different rotation directions). Additionally, the target position could undergo updates since the context could change before the transition process is complete for all recommendations. Such cases must also be covered.

The following algorithms describe the core functions of both components. The Algorithm 4.8 *UpdatePath* is called by the first component for each recommendation whenever a composition step is complete. It determines the path the recommendation must follow to reach its target. It takes the target position and the hull to which it belongs (as an array of points) as parameters. The initial step involves identifying the nearest point on the necklace polygon, especially if the recommendation is not located on the most recent version of the necklace. If so, the initial point on the path corresponds to this point, indexed as i_{best} in the necklace point array. Next, the algorithm determines the direction to move the pearl representing the recommendation on the necklace to reach its destination. The goal is to take the shorter path. However, computing this path can be challenging, especially when speed is a priority. To address this, a heuristic is employed, suggesting that a greater number of array entries (points) in one direction likely indicates a longer path. While this is not always accurate, it is effective in many cases, and a slightly longer rotation for recommendations is not detrimental.

Algorithm 4.9 is simply called *Step* and is called by the rendering loop of the UI toolkit in use. It knows the path the recommendation should travel (the output of the prior function) and takes the elapsed time since the last call as a parameter. In simple terms, it controls the animation. First, the algorithm computes an acceleration factor rooted in the desired animation path's length. A longer path induces greater acceleration, making the pearl move more quickly. On the other hand, shorter distances are covered at a slower pace, ensuring that the animation remains perceivable to users. This acceleration is bounded by property $pearl.a$, serving as a 'speed limit'. During the subsequent step, the algorithm employs these velocity data to identify the pearl's rendering position once a time span of Δt elapses. The base speed and the acceleration factor can be fine-tuned using the two parameters $locality$ and v.

Both functions are associated with one recommendation/pearl on the necklace. Several properties accessed by both functions are encapsulated within a 'pearl property', as detailed below. Consequently, there are as many pearl properties as recommendations and as many function calls are necessary.

Algorithm 4.7 Description of the pearl property, accessed by Algorithms 4.8 and 4.9

▷ *All these variables can be accessed by prefixing them with pearl* ◁
$pos \leftarrow \vec{0}$ ▷ *Current position*
$a \leftarrow 8$ ▷ *Acceleration, defaults to 8*
$path \leftarrow [\dots]$ ▷ *Sequence of points*
$pathLength \leftarrow 0$ ▷ *Length of the desired path*
$pathIndex \leftarrow -1$ ▷ *Index of point on the path, visited recently*
$timePassed \leftarrow 0$ ▷ *How long the animation has already run*
$animationDone \leftarrow False$

Figure 4.23 visualizes how the composition reacts to user input. In 4.23a, one of the four user concepts is slightly moved, resulting in a slightly different context. The recommendations on the necklace adapt to this update by rotating counterclockwise (determined by Algorithm 4.8), as the calculation/optimization of $M(p, r)$ suggests another configuration. The animation component ensures a seamless transition following the predefined path of the necklace. The length of the necklace remains relatively consistent, so no recommendations are removed or added due to changes in the available space.

Figure 4.23b illustrates an example of this. The initial condition corresponds to the final state depicted in 4.23a, with subsequent changes resulting from the move of the upper left user concept further to the left, thereby extending the length of the necklace from its previous state. As Equation 4.16 calculates space for one more pearl, the composition adds the next recommendation from the 'mean weight stack'; it is marked with a '+' symbol in the drawing. The other recommendations no longer rotate uniformly in a single direction, as they did previously, but instead adjust their positions to accommodate the necklace's new configuration, ensuring that they provide adequate space for the newly positioned recommendation and its adjacent neighbors. The position of the new recommendation is based on its position within the circular

Algorithm 4.8 Updating the animation path of a single *pearl*

```
Input: point of target, point array for necklace
function UpdatePath(target, necklace)
    pearl.path ← [pearl.pos, target]                ▷ Default linear path
    ▷ First, move pearl back to the necklace ◁
    d_best ← ∞
    i_best ← −1
    ▷ Use the same loop to find index of the target point ◁
    i_target ← −1
    for index ← 1, index ≤ necklace.size, index ← index + 1 do
        P_tmp ← necklace[index]
        d_tmp ← calcDistance(P_tmp, pearl.pos)
        if d_tmp < d_best then
            d_best ← d_tmp
            i_best ← index
        if target == P_tmp then
            i_target ← index
    pearl.path ← [pearl.pos, necklace[i_best]]      ▷ Move to necklace
    direction ← 0                ▷ We need to determine rotation direction
    ▷ Heuristic: more points might imply greater length ◁
    distanceDirect ← |i_target − i_best|
    distanceWrapAround ← necklace.size − distanceDirect
    if i_best > i_target then
        if distanceDirect > distanceWrapAround then
            direction ← 1
        else
            direction ← −1
    else if i_best < i_target then
        if distanceDirect > distanceWrapAround then
            direction ← −1
        else
            direction ← 1
    index ← i_best
    while index ≠ i_target ∧ direction ≠ 0 do
        index ← (index + direction)(mod(necklace.size + 1))
        pearl.path ← pearl.path + necklace[index]
    pearl.animationDone ← False
    pearl.timePassed ← 0
    pearl.pathIndex ← 1
    pearl.pathLength ← calcLength(pearl.path)
```

Algorithm 4.9 Animation step for a pearl

```
Input: passed time in ms Δt
function STEP(Δt)
    locality ← 200                  ▷ Can be changed to adapt acceleration
    v ← 300                         ▷ Can be changed to adapt base velocity
    a_local ← min(pearl.pathLength / locality, pearl.a)
    expectedAnimationTime ← ((pearl.pathLength / v) · 1000) / a_local
    d_completed ← pearl.pathLength · (pearl.timePassed / expectedAnimationTime)
    d_planned ← pearl.pathLength · min((pearl.timePassed + Δt) / expectedAnimationTime, 1)
    pos_tmp ← pearl.pos
    loop
        if pearl.pathIndex == pearl.path.size ∨
        d_completed ≥ pearl.pathLength then
            pearl.pos ← pearl.path[pearl.pathIndex]
            pearl.animationDone ← True
            break
        pos_next ← pearl.path[pearl.pathIndex + 1]
        d_next ← CALCDISTANCE(pos_tmp, pos_next)
        if d_next > d_planned then          ▷ Next point too far away
            portion ← (pos_next − pos_tmp) / d_next
            pearl.pos ← pos_tmp + portion · d_planned
            break
        else
            pearl.pathIndex ← pearl.pathIndex + 1
            pos_tmp ← pos_next
            d_planned ← d_planned − d_next
    pearl.timePassed ← pearl.timePassed + Δt
```

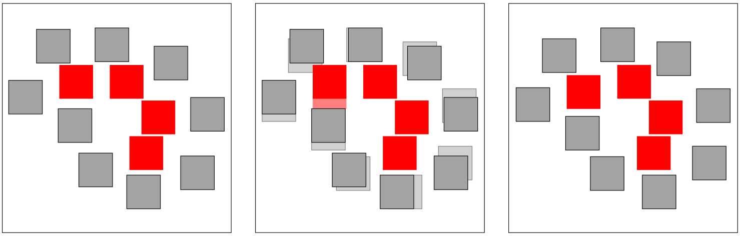
list structure. Unlike physics-based composition, where a stable state must be reached through simulation before displaying a recommendation, this method simplifies the process by allowing for immediate determination of a recommendation's position.

Although not depicted, a comparable response occurs when an update in context leads to a shorter necklace. In these circumstances, the composition removes one or more recommendations. The remaining recommendations then adjust their rotation to preserve their spacing with adjacent pearls, adhering to the predefined bounds and adjusting to the necklace's new, reduced length.

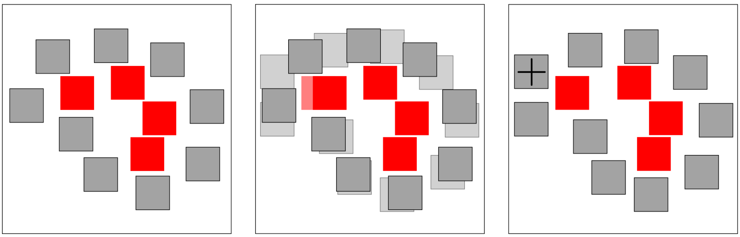
Since the animation and *UpdatePath* components are independent from each other, it is possible to experiment with various call strategies. We have adopted the version where both functions are invoked frequently, approximately 60 times per second, as previously mentioned. Consequently, when a user relocates a concept, the entire composition responds immediately. Alternatively, one could design a mode in which the path update function is called only after a concept's movement ceases, for instance, at the conclusion of a drag and drop action. A mixed mode could also offer advantages. Normally, if a user moves a concept to a significantly different context, the necklace would expand continuously until the move concludes; however, if the update is triggered only after reaching a certain degree of expansion, this could prevent the necklace from excessively stretching across the entire screen in extreme cases. In essence, the outlined composition method presents opportunities for additional refinement, enabling more precise handling of particular scenarios.

Conclusion

The pearl necklace composition offers structured and orderly arrangement of the recommendations. By constraining the recommendations to adhere to a necklace, their movement and interactive dynamics be-



(a) minor rotation due to context update



(b) longer necklace allows one more recommendation

Figure 4.23: Pearl necklace composition: left and right images show slightly different contexts, the middle image does overlay both versions

come predictable, eliminating concerns about collisions, overlaps, or artifacts that might arise from simulations or optimization processes. This streamlined approach has its trade-offs, including the loss of numerous nuanced relationships that are not explicitly represented in the final composition: Recommendations are limited to maintaining connections with only their immediate neighbors, and only the highest-ranked k recommendations are guaranteed to preserve an appropriate distance to the user concepts.

Transitivity As already mentioned in various sections of the algorithm's explanation, the entire composition relies significantly on a concept that could be termed weighted transitivity. The relationships that are neglected are inconsequential if they are offset by transitive links. The circular list structure, or more precisely, the sequence within it, yields satisfactory outcomes only when the underlying graph is of a strong transitive nature.

Transitivity plays an important role not just in one aspect of the composition but in several. For example, the rotation of the entire necklace is influenced by a few k recommendations. These recommendations are expected to align closely with the concepts of the users with which they share the strongest connections. However, this intended arrangement can be disrupted if the user organizes their concepts in a manner that lacks transitive relationships among the nearest concepts. In such cases, the composition's underlying logic could be thrown off, leading to a misalignment between recommendations and user concepts, and thus failing to accurately reflect the intended relationships. Simply put, the algorithm works best if the user follows a logic similar to that of the circular list order.

In Section 5.2, we engaged participants in a user study in which they were asked to arrange five concepts within a simplified spatial hypertext workspace. The primary aim of the study diverged from this aspect, but it provided an incidental observation: Users appear to instinctively organize concepts with transitivity in mind. Nevertheless, the study's scope was too narrow to lend substantial support to this observation, due to the limited number of participants and, crucially, because it only involved the organization of five concepts. More extensive research is required to confirm this tendency with greater confidence.

Interpretation The composition's design facilitates intuitive user comprehension. Naturally, it guides users to perceive the relation of a recommendation to a specific group of user concepts. This is because each recommendation, along with others that align with it, forms a discernible chain resembling a necklace. If a user is drawn to a particular recommendation, they can conveniently navigate along the necklace to discover similar recommendations. On the contrary, should a user seek contrasting options, they can simply explore the opposite side of the necklace. The arrangement of the user concepts themselves suggests a starting point for this exploratory search on the necklace.

Meta Context This relation was handled by the physics-based composition, where general repulsive forces made groups of objects – building the context – distinguishable from each other, even if they are augmented with recommendations. The algorithm outlined in this section does not incorporate any mechanism to facilitate such segregation; in fact, necklaces encapsulating different groups might overlap with each other, just as the individual recommendations on the necklace can. Therefore, some thoughts on this shortcoming.

Figure 4.24 shows an screenshot of SPORE, one of our demonstration clients. There are currently six (visible) user concepts in dark gray that are arranged into two groups with three recommendations per concept. The recommendations are rendered with help of the pearl necklace composition; for debugging purposes, the necklace is visible as a red line. Right in the middle, it shows that both necklaces intersect, consequently the recommendations on them interfere. Even with

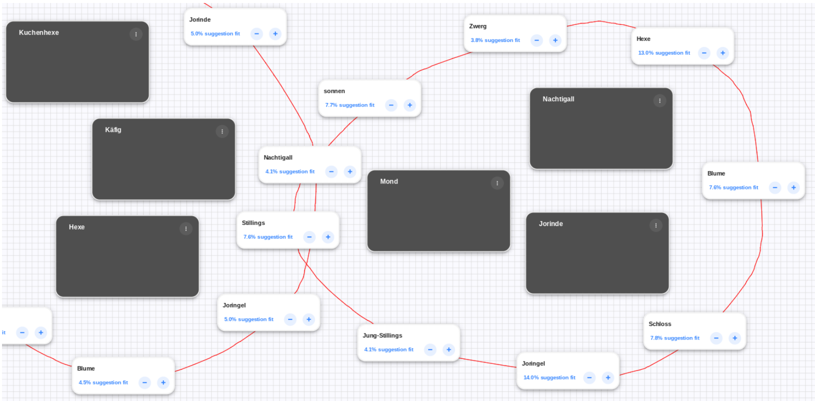


Figure 4.24: SPORE client with pearl necklace composition in debug mode: meta context

this debug view, it becomes hard to distinguish where individual recommendations belong to. This situation is crafted with the intent of showing the worst case, but it helps to visualize the general issue.

There are possible solutions, such as employing repulsive forces as in the physics-based composition, 'blocking' certain parts of the necklace for recommendations, or merging all intersecting necklaces into a larger one. All these ideas come with the downside of interfering with the base of the composition, the circular list structure. In the end, this is a matter of trade-off that needs more research to examine how these interfering contexts influence the perception of recommendations and the spatial hypertext in general.

4.2.7 Summary

This section has established the algorithmic framework for spatial composition, outlining the challenges encountered and the algorithmic solutions devised to address them. We introduced five distinct strategies for spatially composing recommendations within a spatial hypertext context, detailing the implementation of each and conducting a comparative analysis to evaluate their effectiveness and identify their strengths and weaknesses.

The subsequent narrative mirrors our learning journey and aims to contribute to future advancements in the field. The algorithms introduced are not in their final state and offer numerous opportunities for enhancement. They serve as a solid foundation for investigating their impact on aspects such as usability and user experience in different application scenarios, as explored in Chapter 5.

The key takeaways are as follows.

1. Clarity vs. Completeness. With a growing number of concepts, potentially including recommendations as part of the context, and considering the meta context, it is easy to reach a state where it is not possible to visually encode all the relations. It is necessary to have a strategy in place that helps limit the number of relationships to maintain clarity for users. The pearl necklace composition does this by transforming the graph into a circular structure to get rid of most of the relationships. The circular composition ignores that recommendations add to the context and, as the VIBE and heat map composition, ignores the meta context. Whichever strategy is chosen, it imposes requirements on the underlying graph structure.

2. Responsiveness is challenging. The reason for implementing an interactive composition is to help users trace and understand changes in composition. To do so, algorithms need to be time efficient to keep up with the rendering loop of the UI and they need to produce stable results, which means that small context changes must not lead to significant changes within the composition – something we experienced with the VIBE and heat map composition.

3. What is the context? For this thesis, context is defined as the result of a spatial parsing process that considers the color, shape,

size, position of visual symbols and uses this information to infer groups of visually related symbols. This is a rather limited view, even if one considers the earlier introduced meta context. First, there are more factors of interest, e.g. the perspective/viewport of a user. Second, the inference process builds distinct groups; a symbol cannot be part of more than one group, hence the presented composition algorithms do not need to handle such situations. The definition of context is – again – a trade-off between completeness and clarity that has to be made.

4.3 Asgard – Structure Service

Asgard is the name of the structure layer of the *Mother* architecture, as already detailed in Section 3.2.3. In the overall system and in the context of this work, this layer encompasses components that are related to structure services; components, interfaces, applications, and algorithms that interpret, validate, or make use of hypertext structure. This term can also be contextualized in so-called structure domains (e.g. spatial [120, 29], navigational, or taxonomic structure [86]), all with their own specialties and challenges. While visual representation of knowledge is done in the user interface and is, therefore, described in Section 4.2; this section discusses aspects for deriving knowledge from user activities and the application of spatial parsing to generate queries for knowledge bases.

The conceptional boundaries are already laid out. Spatial parsing enables the machine to gain a fundamental understanding of the visual structure. Furthermore, the findings of a user study in the following chapter indicate a relationship between the proximity of visual symbols and knowledge. It suggests a linear correlation between relative distances and the weight that users assign when presented with a pair of concepts. Details of the study are given in Section 5.2. However, the user test creates an artificial setting, driven by the scientific necessity to determine the nature of the relationship. Therefore, the implementation within Asgard is approaching the question of how to translate human-generated structures to infer knowledge with a certain use-case in mind.

4.3.1 Knowledge from Structure

The aforementioned synthetic design of the study implies that the workspace for users (participants) is simplified. In addition to limitations with respect to user interactions, spaces are (1) immutable, (2) created by one person only, and (3) limited in complexity. How does this impact the goal of inferring knowledge when applied on a larger scale? Immutability means that users do not revisit the space to alter the content. At the moment of choosing to move forward, they imply that their intended expression has already occurred. Simply put, an algorithm can use the content and structure to infer the expressed knowledge. However, in practical applications, particularly in the domain of exploratory search, a state of completion is often absent. To function effectively, an algorithm must be designed to identify either a state of (relative) completion or facilitate a progressive inference process. In this process, knowledge should be capable of being updated or discarded by the system as changes occur in the space.

That only one person is working in a space is also a limitation that is rather unusual in practice. Especially in an industrial setting, workspaces can be shared with others as they are a valuable resource on their own. This complicates the interpretation of the distance, as its utilization might differ among the users. There is no single valid way to express ones thoughts and knowledge. Therefore, it may be not suitable to directly convert from distance to a weight, as it was done in the study. Instead, an algorithm must focus on the structure itself, how it emerges, which visual relations are added and which are destroyed.

Limitation (3), the lack of complexity in the test setup, refers to the fixed number of five concepts imposed on the participants. Although this suggests issues with multidimensional scaling, the potential structures involved are relatively simple, with minimal contradictions and uncertainty. Similarly to point (2), the increased complexity in real-world applications makes the direct use of distances problematic. Concentrating on structure may offer greater robustness in knowledge inference.

In conclusion, the setup of the test helps answer the raised questions, yet the approach needs some enhancements to be ready for real-world application. Therefore, further development is based on two research projects, both of which are already described in Chapter 2. The first project focused on providing domain experts with easy access to industrial machinery maintenance data. The goal was to enable them to benefit not only from a readily available data set but, more importantly, from the collective knowledge of their colleagues. The second project's goal was to develop a story generator (fairy tale). This system is designed to use the spatial arrangement of concepts to create stories for children. It aims to enhance their creativity by offering recommendations and learning from their input to suggest ideas from one user to another. This project provided a great opportunity to observe how these recommendations affect children using the system. Both projects are affected by all three points raised and can serve as the blueprint for the following thoughts.

Pre-Conditions

Both projects feature a knowledge base that is already initialized with some data; concepts (e.g. machine types, fairy tale characters), and weighted relations. In this way, the system is able to provide recommendations when users add concepts to their space that are already available within the knowledge base. Users in both application domains interact with *workspaces*, which are initially blank areas that store content and its structure, allowing them to be revisited at any time. Additionally, these workspaces can accommodate multiple users at the same time.

Users interact with the system by adding concepts, either by accepting recommended information or by manually adding them – potentially information that the knowledge base is not (yet) aware of. Furthermore, since this is the essence of the spatial hypertext method, they can organize their concepts in multiple contexts, receive spatially composed recommendations, and utilize this to explore the knowledge base in an iterative and interactive manner. It is possible to create and remove multiple workspaces and change them at any time.

To introduce some formalism, we define knowledge as a graph $K = (C, E)$, where C represents the set of concepts, $C = \{c_1, c_2, \ldots, c_n\}$, and E represents the set of edges or weighted relations between these concepts. Each edge in E can be denoted as $e_{i,j}$ with a corresponding weight $w_{i,j}$, implying an undirected relationship between concepts c_i and c_j with weight $w_{i,j}$. Thus $E = \{(e_{i,j}, w_{i,j}) | e_{i,j} \text{ connects } c_i \text{ and } c_j\}$.

The same approach can be applied to the range of possible interactions. It is important to note that modification of the appearance of a concept encompasses all types of alterations, including changes in position, color, or shape. They are grasped by spatial parsing.

Workspaces Actions

Let S be the set of all workspaces.

- Create workspace: Represented as $S = S \cup \{s\}$, where s is the new workspace created.
- Delete workspace: Represented as $S = S \setminus \{s\}$, where s is the workspace to be deleted.
- Open (enter) workspace: Let $s \in S$ be the workspace to be opened.
- Close (leave) workspace: Let $s \in S$ be the workspace to be closed.

Actions Within a Workspace

- Let C_s represent the set of concepts in workspace s.
- Add concept: $C_s = C_s \cup \{c\}$, where c is the new concept added to workspace s.
- Remove concept: $C_s = C_s \setminus \{c\}$, where c is the concept to be removed from workspace s.
- Modify concept appearance: Let $c \in C_s$ and $appearance(c)$ be a function that modifies the appearance of concept c. This can be represented as an update to the attributes of c in C_s.

Algorithm

In the context of those user interactions, the algorithm is designed to account for each specific interaction, with each having a distinct impact on the inference of knowledge based on user activities. It is crucial to continuously consider the three key points outlined in the introduction of this section. The processes of adding, removing, and modifying concepts within a workspace will be examined first.

Addition When a concept c is added, it is already known in the knowledge base or introduces new information. In the latter case, c is added not only to C_s, but also to C and therefore becomes part of the knowledge graph K. However, the algorithm cannot infer any new weighted edges to E, because it cannot determine if the user settled his or her session to a stable state such that it is sufficient to be used to infer knowledge. In case c is already known and part of K, there is nothing to infer.

Removal A removed c from C_s does not necessarily reflect in C, because concepts can appear in more than one workspace s. For the *addition* process, it has been argued that the algorithm should not instantly modify E, as the workspace may be in a state of flux where the user plans to make additional changes. As a result, immediate removal of a concept does not lead to direct deletion or updating of the corresponding entries in E.

Modification This covers all interactions, except the two aforementioned ones, that lead to a changed context and trigger a different parser result, as already detailed in Section 4.2.1. Most of these modifications contribute to the interpreted structure but are also preliminary and cannot be used directly.

In essence, although these actions form the foundation for knowledge inference, it is important for the algorithm to differentiate when a workspace is in an appropriate state for interpretation. This involves assessing the stability and completeness of user interactions within the workspace. The algorithm must be capable of distinguishing between transient, exploratory changes and more deliberate, finalized modifications. This discernment is essential to accurately interpret the spatial structure and modifications as meaningful reflections of knowledge, ensuring that premature or incomplete changes do not lead to erroneous inferences.

Therefore, our approach takes workspace activity as an indicator of stability. As long as any user has an active session, we assume that the workspace is not ready to be used to infer knowledge. An active session ends when a user leaves the space by closing the client application. Once a workspace s is ready, spatial parsers (cf. Section 4.2.1) are used to analyze the space context and create an interpretation of the visual structure. This interpretation is compared to the previous interpretation of the very same space, to identify structure changes that should reflect on the knowledge graph. If there is no previous interpretation because it is a new workspace, we assume an empty graph as the previous interpretation.

Essentially, the interpretation of a workspace s resembles that of a knowledge graph and is represented by a graph $K_s = (C_s, E_s)$, which comprises concepts and edges. In this model, the edges are deduced for concepts that have spatial/visual relationships; unlike in the knowledge graph K, these edges are unweighted. As explained above, each C_s in the workspace has a corresponding C in K, therefore, if a user introduces a new concept, it must also be added to K as well. Of greater interest are the changes within E_s:

- If there is a edge E_s that was not present in the previous run, the weight of the corresponding E in K should increase (or E should be created if not yet existent)
- If there is a edge E_s missing that was present int the previous run, the weight of the corresponding E in K should decrease (or E should be removed if the weight drops to 0 or below)
- An edge E_s that is present in the most recent and the previous interpretation run does not reflect on K.

Momentum The process of increasing or decreasing weights can occur either relatively or absolutely, yet neither approach fully captures the dynamic nature of knowledge. Knowledge often evolves rapidly and may become obsolete over time. To address this, we propose introducing a *momentum* factor that moderates weight changes based on recent trends. The underlying principle is as follows: If an edge in K has frequently experienced an increase in weight in the recent past, subsequent increases should be amplified, suggesting that the weight is currently lower than it should be. In contrast, if many workspaces experience the removal of an edge E_s, this indicates that the weight of the corresponding edge in K should be reduced more significantly. Consequently, an edge that experiences a similar amount of in- and decreases seems to be in a stable state, and hence the weight should not change dramatically in such cases.

To achieve this, the algorithm takes into account the sign of the N

last changes in weight and uses them to calculate the momentum factor p – as shown in Algorithm 4.10.

Algorithm 4.10 Momentum

```
Input: history of N most recent weight changes
p ← 1
h ← 1
l ← 1
for all change ∈ changes do
    if change < 0 then                ▷ A change of 0 is not a change
        l ← l + 1
    else if change > 1 then
        h ← h + 1
return h ÷ l
```

Calculating the new weight w_n of an edge becomes straightforward once the direction of the change (increase or decrease) and a fixed base value w_Δ are established. The calculation can be described as follows:

- For an increase: $w_n = min\{w + w_\Delta \cdot p, 1\}$
  This formula adjusts the weight by adding a product of the base value w_Δ and a factor p, ensuring that the new weight does not exceed 1.

- For a decrease: $w_n = max(w - w_\Delta / p, 0)$
  Here, the weight is reduced by subtracting the base value w_Δ divided by the factor p, with a lower limit set at 0 to prevent the weight from becoming negative.

Summary

The method detailed in this section focuses on extracting knowledge from organizational structures created within a spatial hypertext interface. This approach is particularly relevant given the fluid and evolving nature of knowledge, especially in the context of its application at an industrial scale. In such environments, spatial hypertext applications are not just tools for individual use but become collaborative platforms where numerous users interact within the same virtual space over extended periods. This collective interaction contributes to the complexity and richness of the data, as the spatial arrangements and connections made by users reflect collective intelligence and diverse perspectives. By analyzing these structures, the system can adapt and learn, offering insight to its users.

4.3.2 Query Derivation

Query derivation in context of the system implementation refers to the process of building a knowledge base query, based on user actions and content in the workspace. This is related to *visual querying*, a term that is already discussed in the related work Section 3.3.2. There is little research on how a spatial hypertext context can be utilized to build queries. Schedel [114] proposed an "intelligent browser interface" as a practical application for spatial parsing. In this context, "interpretation graphs are treated as networks of search terms".

This thesis primarily concentrates on the representational aspect and, consequently, implements a straightforward query derivation algorithm that operates in two distinct modes. The first mode refers to the group detection that is utilized by the composition algorithms: As previously discussed in Section 4.2.1, the context within the system is formed by groups, which are identified by spatial parsers. As a result, the generated recommendations are based on these identified groupings. This necessitates the derivation of queries in a manner similar to how groups are identified by the spatial parsers. Essentially, the process of query formation should mirror the method used for group identification to ensure relevance of recommendations provided.

After identifying groups, a query is built for each group to fetch all entities that have a relationship with at least one entity within the group. Essentially, this process forms an OR query, where the retrieval of related entities is based on the presence of a relationship with any single entity in the group. This approach ensures comprehensive coverage of related entities, broadening the scope of information gathered for each identified group.

Of course, the method for identifying groups can be altered; another simple approach that was used in some applications is to interpret the entire workspace as a single group. This makes spatial parsing obsolete, but may lead to less accurate results, as it is not possible to focus on different aspects in the workspace. Spatial composition can help, in either mode, in making means of the provided result.

4.4 Summary

This chapter explained the algorithmic details of the system, with a strong focus on methods to compose recommendations in a spatial hypertext application. Hel (cf. Section 4.1), the "knowledge world" [13] is the fundamental layer, defining what knowledge is in the context of the proposed system and how it is organized. Asgard (cf. Section 4.3), the "structure world" [13] is the layer that defines the context of the system. Spatial parsers are the key in setting the required limits for the context, making spatial composition work. They also allow the system to learn from the visual information provided by users. In simpler terms, the way users arrange things visually helps the system understand and respond better to what the users are doing.

Midgard (cf. Section 4.2), the "user world" [13] represents the spatial hypertext interface. The composition algorithms outlined in this section detail the identified requirements, challenges, and solutions encountered in the context of spatial composition. Further, this section defines the setting of spatial composition, which is also relevant for the tests described in the next chapter.

The physics-based and pearl necklace composition approaches represent significant contributions to the research community and lay a foundation for future investigations in this field. These methodologies represent some of the initial fully documented algorithms for spatial composition used in real-world industry research projects, and their principles have been supported by user testing.

5 User Studies

This chapter delves into analyses and user studies designed to probe and assess the user experience associated with spatial composition. The first study investigates the link between proximity and knowledge acquisition – a vital aspect for the effective deployment of spatial composition and for leveraging user-generated structures to deduce knowledge. Following this, a second study, complemented by a thorough analysis of the collected data, examines how spatial composition alters the user experience compared to traditional recommendation displays, such as ranked lists.

It is important to note that both studies were conducted entirely or partially during the global COVID-19 pandemic. This situation presented challenges in setting up traditional laboratory environments, pushing us to find innovative ways to conduct research remotely and in more diverse settings.

5.1 COVID-19

The conduct of in-person user tests during the COVID-19 pandemic posed significant challenges and limitations to the global research community in general, hindering the practicality of such tests. The subsequent disclaimer seeks to clarify the challenges faced during in-person user testing and examine the impact on research procedures.

1. Health and safety concerns: The main focus during the COVID-19 outbreak was to safeguard the health and safety of partici-

pants, researchers, and all individuals involved. Due to the extremely contagious character of the virus, it was essential to follow public health recommendations and procedures, such as reducing in-person contact and upholding social distancing measures.

2. Travel and mobility restrictions: Widespread travel restrictions, lockdowns, and limitations in mobility were imposed to mitigate the spread of the virus. These restrictions made it challenging to gather participants from various locations, preventing the ability to conduct in person user tests, especially when the target user base was geographically dispersed.

3. Ethical considerations: In-person user testing demands close proximity between researchers and participants, often involving physical interactions and shared surfaces. This raised ethical concerns about possible virus transmission and compromised the duty of care owed to participants.

4. Time constraints: The COVID-19 pandemic introduced unexpected disruptions, shifting priorities, and increasing demands on researchers' time and resources. The immediate challenges and adaptation of the research methodologies to comply with health guidelines required significant effort and reevaluation of the study designs, which could lead to delays and compromises in data collection and analysis.

5. Uncertainty and volatility: The evolving nature of the pandemic introduced an unprecedented level of uncertainty and volatility. Government guidelines and regulations varied between regions and changed frequently, making it difficult to plan and execute user tests in person consistently and reliably.

In light of the significant obstacles and restrictions brought about by the COVID-19 pandemic, alternative research methods like remote user testing, online surveys, and virtual simulations were considered as feasible options to address the challenges faced during the research process and maintain the progression of knowledge acquisition.

5.2 From Spatial Structure to Knowledge

This section builds on recently published work [110]. It has been further extended and adapted for the purposes of this thesis.

It is an often stressed argument in this thesis that spatially structured entities are a visual (emerging) representation of (emerging) knowledge. Based on this conception, we use the interpretation of spatial parsers to construct a query context, leverage human-generated structures to develop a knowledge base, and utilize the relation between 'weight' of recommendations and proximity for spatial composition. However, there is lacking evidence and understanding of how visual structure is related to knowledge (cf. Section 1.2).

Therefore, we implemented a user test to shed some light on this question. First, we need to narrow down the terms "spatial structure" and "knowledge" for the intended setup.

The technical description of how knowledge is modeled in our system can be found in Section 4.1 – but more important for this test is how this concept maps to what human-beings experience as knowledge. *Mental models* are a way to explain how individuals process and interact with the world around them. They are internal representations of external realities, which serve as a framework for interpreting reality based on our experiences and internalized knowledge [34, 63]. Our simple framework, where knowledge is modeled as an associative network of related concepts without directed links but weighted connections to signify the strength of relationships, aligns with the abstract concept of mental models.

To keep it simple, spatial structure in this setup means a simplistic version of what is described in Section 3.2: A fixed number of concepts, each represented as a rectangle of the same size, containing a word describing the concept. They can be freely arranged in space, but cannot be deleted or altered in size, shape, color, or any other visual property. The space itself is more simplistic as well and does not allow one to change the viewport by panning or zooming.

5.2.1 First Thoughts and Preliminary Test

As a first step, it was important to design and implement the test. Understanding knowledge as the nuanced relationship between two concepts is challenging to extract in such experiments. This is largely because it is deeply influenced by the participants' prior knowledge and is difficult for users to articulate as well as for the system to accurately measure. To reduce the impact of prior knowledge, we aimed at identifying *simple* concepts, which are most likely part of everyone's daily life. For the pre-test, we chose 30 terms describing a concept, inspired by a similar test carried out by Schedel [114]. Although most of our prospective participants are native German speakers, we opted for English nouns. This decision not only welcomes international participants, but also simplifies the dissemination of our findings. Some sample terms are tree, trunk, leaf, school, teacher, infrastructure, water, gas, hospital, roof, building, park, and street. They should be easy to understand and convey various types of relationships (e.g. hierarchy: tree to leaf, or building to school; logical: school and teacher). While many are obviously (not) related, some other concepts share a more controversial relation, e.g. street and tree.

This led us to the idea of letting participants *rate* the relation of two concepts on a given numerical scale, and hence the whole setup was called *Pairwise* from this point on. The pairwise ratings build the baseline, the ground truth of knowledge, which can be compared with other forms of knowledge generation – e.g. a simplistic spatial hypertext interface where users can arrange concepts. As it should be simple, without the possibility to move and zoom the viewport, it becomes clear that the interface cannot display all (30 in our case) concepts. Users would probably struggle to find a good structure, or even worse, become frustrated and end the task too early. Therefore, we chose to reduce the number of concepts shown in the interface and instead let users solve multiple instances. The created spatial structure of each instance should be analyzed by a spatial parser to retrieve the weight between two concepts. Spatial parsing is a rather complex task that can be solved by various approaches (cf. Section 3.2.2). For this test, we decided to implement a very simple parser which uses the distance between objects in the space to measure the grade of their visual relation, relative to the largest distance in an instance. Consider Figure 5.1 as an example: Four (n) objects ($O_1, O_2, .., O_n$) are arranged by a human, and the distances between are measured by the parser. The longest measured distance is between objects D and B; for the sake of simplicity within the illustration, it is denoted by the number of dashes the line has, ten. Objects C and D overlap, and hence have a distance of 0 between each other. In the next step, all distances are normalized with the longest distance d_{max} as the base and a rating $R(O_i, O_k)$ is calculated:

$$R(O_i, O_k) = 1 - \frac{d(O_i, O_k)}{d_{max}} \tag{5.1}$$

The resulting ratings for this sample are shown in Table 5.1; note that a rating of 1 is only reachable if two objects touch or overlap. This simple approach is not as capable as other parsers and lacks the ability to detect larger visually connected groups of objects. However, as we limit the number of objects in the space to a reasonable number, this flaw should not influence the result; furthermore, better parsers

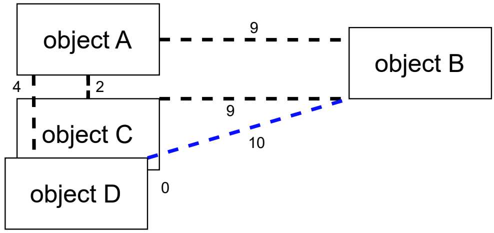


Figure 5.1: Four objects and their distance measured in line dashed; longest distance is highlighted in blue (10)

would generate an even more sophisticated interpretation. If this simple implementation shows to be able to infer knowledge, as defined above, other parsers will be as well.

Table 5.1: Interpreted ratings, based on the visual structure depicted in Figure 5.1

O_i	O_k	R
A	B	0.1
A	C	0.8
A	D	0.6
B	C	0.1
B	D	0.0
C	D	1.0

For the pre-test, we wanted to find good candidates for the following numbers; the numbers actually used in the pre-test are in parentheses:

1. Define a suitable number of concepts (30), participants (34), and concepts per spatial interface instance (5)
2. Define the number of pairwise comparisons (8) and spatial interface instances (4) which have to be solved per participant

The test system is implemented as a Web application, based on

Vue.js[1] to ease participation and allow participants to use their own familiar device. Figure 5.2 and Figure 5.3 are exemplary screenshots of the application. In the pre-test, every participant undertook both versions, and we recorded both the time taken to complete the tasks and the positions of the concepts within the spatial interface. When looking at the chosen numbers, it becomes apparent that individuals rate only a portion of the entire data set. If n denotes the number of concepts, then there are Δ_{n-1} edges.

$$\Delta_{n-1} = \binom{n}{2} = \frac{n \cdot (n-1)}{2} \tag{5.2}$$

Given the 30 concepts selected for the pre-test, there are 435 distinct pairs. However, only eight of these pairs are evaluated. With the 34 participants we managed to recruit for the pre-test, a maximum of 272 pairs can be assessed. This implies that an analysis of the ratings might not yield significant insights; however, this is not a primary concern for the preliminary test.

Discussion, Results and Conclusion

For the pairwise comparison, the 34 participants (mostly undergraduates) took around seven seconds to rate one pair (mean) and, therefore, about one minute to complete the whole session. An instance of the spatial variant was solved in 28 seconds on average; hence a session takes about two minutes, given the four instances that needed to be solved. In particular, with the spatial interface, there are more rated pairs, since one instance yields (cf. Equation 5.2) ten unique pairs, given the five concepts displayed in one instance.

As noted previously, if there are too many concepts and/or an insufficient number of participants, the representation of edges in the knowledge graph becomes inadequate for deriving meaningful conclusions. Furthermore, we discovered that many students, even though participation was optional, lost interest during the test. For example,

[1] `https://vuejs.org/`

some participants skipped some spatial instances and did not interact with them at all. To avoid these issues in the next test run, we decided to reduce the number of concepts and increase the number of pairwise comparisons so that each participant has to rate *all* edges. Such pairwise evaluations can be quite monotonous, so we aim for a test session to be no longer than five minutes. This translates to approximately 43 comparisons. Given that ten vertices result in 45 edges, this seems to be a suitable number of concepts for our objective.

For spatial instances, we found that five concepts worked quite well, especially as we measured a wide range of screen sizes. Increasing this number would make it difficult to complete the task on smaller screens; lowering it would lead to less complex and less ambiguous structures. To have a rating for all pairs, at least five instances would be required in each session. As each pair is rated in context of other pairs and the spatial session should have a duration similar to the pairwise comparison, we decided to let each participant solve 11 instances. In this way, each pair appears in at least two different contexts, which should lead to more accurate results.

The question of the desired number of participants lasts. Since each participant evaluates the entire knowledge graph, our goal is to enhance the quality of the test sessions by providing a personalized introduction to the system and observing the task. By doing so, we can gather additional feedback on the users' thought processes and intentions as they arrange the concepts.

5.2.2 Test

Experimental Design

As in the pre-test, each participant is asked to solve two tasks; first rate all 45 unique pairs of ten defined concepts within the user interface described above and depicted in Figure 5.2. Afterward, they completed 11 instances of the spatial variant (cf. Figure 5.3) by arranging five concepts in a 2D space. For the initial task, the concept pairs are randomly mixed to prevent users from finding the pairwise ratings too monotonous. The second task employs a spatial interface, with five concepts per instance, which are *stacked* in the initial view, so that participants can only see the topmost concept. An example is shown in Figure 5.4.

The procedure that decides which concepts are presented in each spatial instance is randomized as well. First, a shuffled list of all 45 pairs is created. Then, all instances are created in the same iterative manner. The first pair, not yet evaluated, is picked from the shuffled list and assigned to the instance. Depending on how many pairs or, more specifically, concepts are already included in the instance, each pair introduces a minimum of one and a maximum of two new concepts. If an instance already comprises four concepts, and thus only has space for one more concept, the first unevaluated pair that has precisely one concept already in the instance is selected. If no such pair remains, the set is reshuffled, and pairs can be reused. As already discussed, this algorithm ensures that each participant has each possible pair at least twice in the task, considering the 11 generated instances.

Arranging concepts in a meaningful layout happens by moving the objects within the space. The system must be used with a mouse, trackpad, or similar devices; touch interactions are not supported by the interface. To move an object, the cursor must be positioned within its boundaries. Once the primary (mouse) button is pressed, the object will track the cursor's movement until the button is released. This is the only possible way to interact with the spatial hypertext, users cannot alter the viewport or anything like that.

Between the objects, the interface draws light gray lines, as can be seen in Figure 5.3. Their stroke ('thickness') is set according to length and therefore becomes thinner when two objects are moved apart from each other. This is intended to prompt users to view proximity as a significant element and to illustrate the pairwise relationships in this spatial representation.

The pre-test was done with students, some of them with an international background, most of them were German native speakers; but all were able to handle the rather simple English terms. For this run, we decided to use German terms because this makes it possible to recruit groups of participants who otherwise would not be able to solve the task due to lack of understanding of English terms. We chose ten concepts that were already used in the pre-test and translated them into German. As before, we aim at providing concepts that are obviously (not) related and those whose relation is not obvious. In Table 5.2 the concepts are displayed along with their English translations; it is worth noting that some concepts carry a wide range of meanings, so a translation might include several terms.

Table 5.2: Ten concepts and their English translation

Concept	**Translation**	**Concept**	**Translation**
Laub	(rotten) leaves,	Baum	tree
Lehrer	teacher	Schule	school
Infrastruktur	infrastructure	Ast	branch
Fahrrad	bicycle	Physik	pyhsics
Photosynthese	photosynthesis	Erde	earth, soil, ground

Measures

During the test, any interaction with the system is recorded and stored for later analyzes. For pairwise evaluations, we measure the duration from the moment a pair is first displayed to the moment the “Next” or “Finish” button is pressed. In addition, order and ratings are saved. Within the spatial instances, the same timekeeping is used and stored, along with the following data.

- concepts of an instance and their initial order when stacked
- resolution and dimensions of the browser window during the test
- the movement of objects, the updated positions and the order of those event, such that a detailed replay of an instance and/or session is possible; a movement is triggered when an object is pressed, the mouse (together with the object) moves and the button is released
- the number of clicks each concept registers; determined by a mouse button press followed by its release, irrespective of any mouse movement in the interim

Participants

We recruited thirteen native German-speaking participants using convenience sampling, with the goal of having a demographically diverse group. The mean age was 38 years and ranged from 15 to 65 years. Of the participants, five were women and the rest were men. Three participants had prior knowledge of (spatial) hypertext, while the others lacked any IT or research-related background. Before collecting any test data, informed consent was obtained from all participants. Participation was optional, and no compensation was provided.

Procedure

Although the pre-test was conducted during the active phase of the COVID-19 pandemic, this test took place in 2023, when all regulations and restrictions ended. All participants were personally approached, either in person or remote via Zoom[2], and asked if they would like to volunteer in the user study, which will take approximately 15 minutes to complete. The environments were diverse, mostly at home, and thus tasks were solved on familiar devices which must meet the requirements described above (mouse-like input device). After the candidates agreed

[2]Video conference application: `https://zoom.us/`

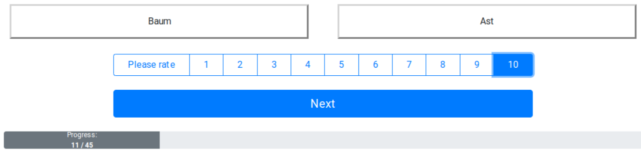


Figure 5.2: Pairwise comparison between Baum (tree) and Ast (branch); rated with a ten

to participate, they were asked to provide their age, gender, and if they had prior knowledge of (spatial) hypertext; answering the first two questions was optional.

Then they were introduced to the tasks they needed to solve: Rating 45 pairs of ten concepts on a scale from 1 to 10, where 1 means “both concepts are barely or not related to each other” and 10 describes “a close relationship”. The participants were informed that there were no incorrect responses and encouraged to trust their intuition. If they encountered an “interesting” pair, they were advised to remember it or make a note.

When done with the first task, they were introduced to the second task with explaining that they should arrange five concepts in a 2D space such that it makes sense to them; it was emphasized that proximity and structure are the important outcome of this task. Again, it was highlighted that there is no wrong result or interpretation.

Finally, a short interview followed, either in-person or via Zoom (remote). Participants were asked to explain one randomly chosen instance of the spatial task and what their intention was to structure the concepts in this specific way. When participants placed objects in overlapping positions, they were asked if this was done deliberately. If not, then they were asked why they used overlap as an indicator of strong relationships. In the end, an open discussion followed, in which more questions could be asked from both sides.

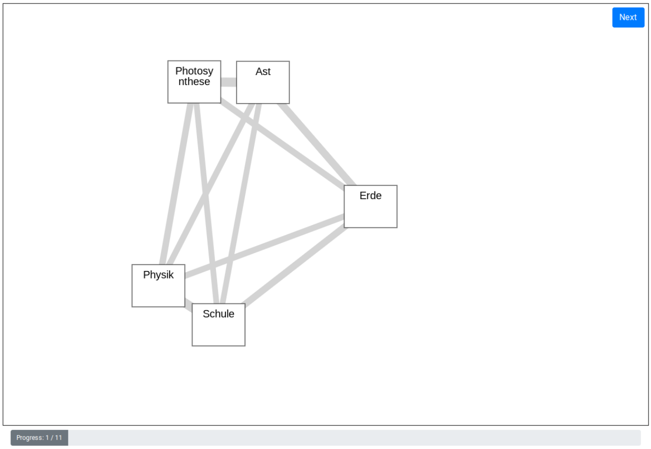


Figure 5.3: Spatial test instance with five concepts; thickness of lines is an additional indicator of distance

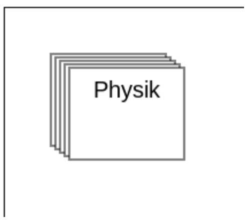


Figure 5.4: Spatial test instance with five stacked concepts; initial view

5.2.3 Results

All participants completed both tasks, although one participant experienced a bug in one spatial instance: It was possible to move objects outside the desired frame, without the possibility of reversing this action. As a result, this particular instance had one concept that could not be positioned as intended by the participant. Since only this concept was affected, it was excluded from that session and did not influence the following results. Table 5.3 shows a summary of the measured task durations and the inferred ratings per participant. Ratings for spatial sessions are calculated as described in Equation 5.1, and the values of the pairwise comparison are adjusted to a normalized range of 0 to 1 to ensure consistency.

Table 5.3: Summary of rating and duration measurement for both tasks

	Duration (in s)			**Rating**		
	min	mean	max	min	mean	max
Pairwise Comparison	2.7	6.3	11.7	0.09	0.49	1.00
Spatial Session	17.4	30.5	51.9	0.24	0.51	0.88

One of the objectives of this test is to demonstrate that the inferred ratings of the spatial variant align with those of the baseline. To do this, we compared the 45 pairwise values against each other. The mean absolute error (MAE) between both variants is 0.10 and the overall distribution is represented in two box plot charts, cf. Figure 5.5. Before employing tests to check for correlation, it is crucial to confirm that the participants had a similar opinion on how strongly the concepts are related to each other. Therefore, the single score *intraclass correlation*[3] for all participants in the pairwise comparison task is calculated. The $ICC(A,1)$ resides in the 95% confidence interval of $0.42 < 0.533 < 0.68$.

[3] $ICC(A,1)$: Two-way random, single measures, absolute agreement

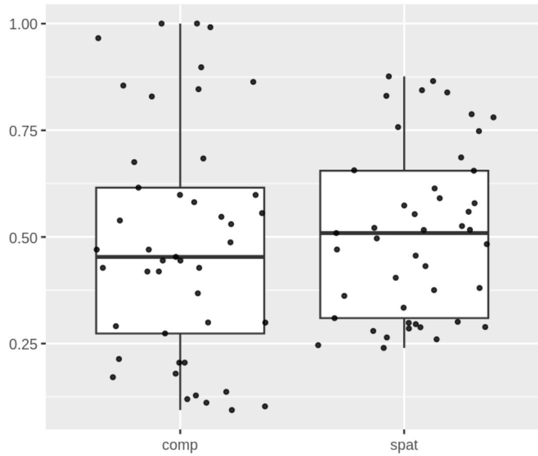
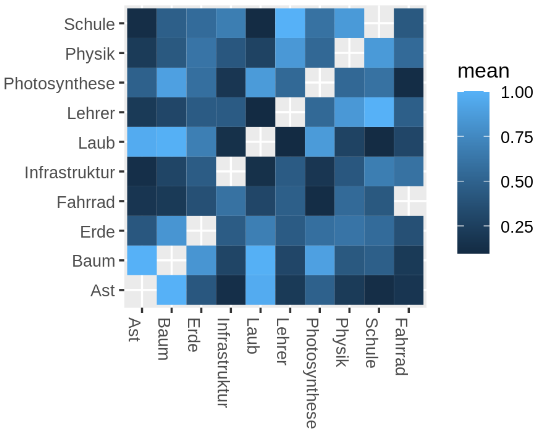


Figure 5.5: Inferred ratings for pairwise *comp*arison and the *spat*ial task

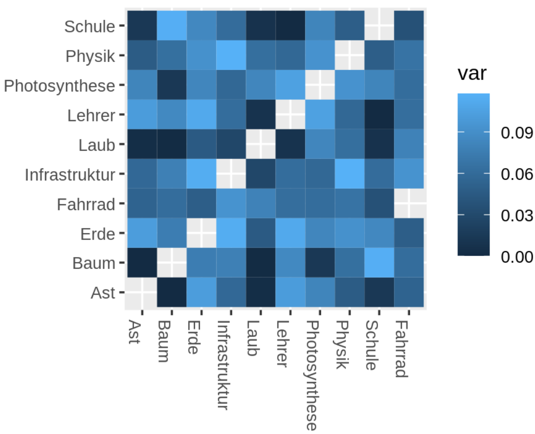
A F-Test, comparing both distributions, confirms the null hypothesis of equal variance with $p < 0.001$. However, normality cannot be confirmed by the aforementioned box plots and a Shapiro-Wilk-test, which null hypothesis got accepted with $p < 0.005$ for both distributions. The ICC-value indicates moderate [71] to good [31] inter-rater agreement. Cohen's weighted kappa (squared distance off diagonal) is identical to $ICC(A,1)$ [125], following the suggested interpretation of Landis and Koch [73], this means a "substantial" inter-rater agreement.

To gain understanding into participants' rating decisions for each pair, we examine the values and their associated variances. Figure 5.6 shows the mean rating for each pair and how 'controversial' they are. The pairs with the highest mean rating (1.0) and lowest variance (0.0) are Baum (tree)/Ast (branch) and Schule (school)/Lehrer (teacher). The lowest rating (0.012) is derived for Photosynthese (photosynthesis)/Fahrrad (bicycle); the most controversial pair is Laub (leaves)/ Photosynthese.

To assess the correlation between the ratings of both variants, we



(a) mean values



(b) variance values

Figure 5.6: Mean ratings and their variance derived from pairwise comparison

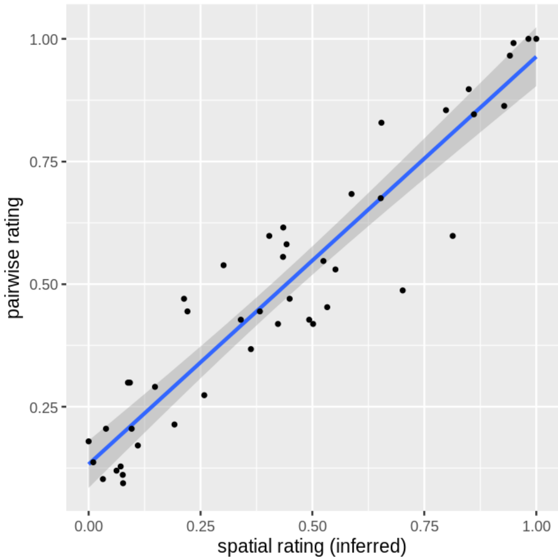


Figure 5.7: Ratings for all 45 concept pairs, derived from pairwise comparison and inferred from spatial instances

plotted the average values on a scatter plot and fitted a linear model (lm) with a confidence interval 95%, as shown in Figure 5.7. Testing for correlation can be achieved with calculating Kendall's rank correlation tau, as the ratings do not follow a normal distribution. The resulting Tau-b value is $\tau = 0.754$ with $p < 0.001$, hence we can accept the null hypothesis of a Tau not equal to 0. Botsch [21] and Cohen [32] give some advice on interpretation, stating that $\tau > 0.5$ indicates a strong relationship.

While assessing the task duration and the click count for the spatial variation, we discovered an interesting relation between both variables, which is depicted in Figure 5.8, together with a linear model (lm) and the 95% confidence interval. When participants allocate more time to a spatial instance, the click count (along with movement) within that instance rises. This finding is reinforced by Kendall's correlation test,

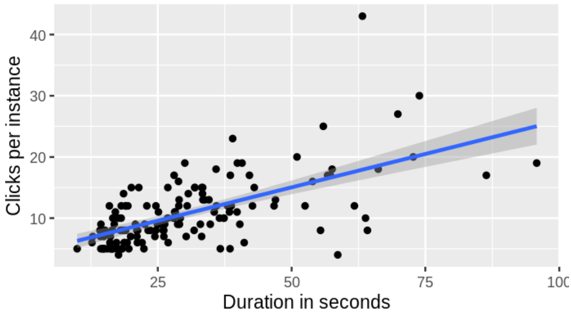


Figure 5.8: Clicks per spatial instance by duration in seconds

yielding $\tau = 0.480$ and $p < 0.001$. As mentioned above, this signifies a robust association between the two variables.

The interviews were conducted in an informal manner, so the subsequent findings are presented descriptively. A notable observation from all interviews was that some concept pairs seemed particularly controversial in their weighting. The discussions revealed two primary reasons for this. First, some terms are ambiguous. For instance, Erde (earth) was interpreted by most of the participants as "planet", not as "ground" or "soil". In the case of Laub (leaves), one participant thought of leaves lying on the ground and not those still attached to a tree. Consequently, the relationship with photosynthesis varies. Another reason is the different range of interpretations: The majority of participants believed that Physik (physics) and photosynthesis were unrelated, since photosynthesis is considered more of a biochemical process. However, some participants pointed out the physical foundations of the same process. During the testing phase, it became evident that even seemingly unrelated concepts like Lehrer (teacher) and Fahrrad (bicycle) were perceived as having a strong connection. Thus, participants were asked for their rationale. Interestingly, mental imagery was decisive: The combination of teacher and bicycle consistently evoked an image of a teacher riding a bicycle to school among most of the participants.

Overlapping was used by three of the participants; others avoid overlapping explicitly. When asked about their choice to use (or not use) overlapping as an indicator of a strong relation between two objects, participants mentioned finding one method more "appealing" than the other. Furthermore, they noted that their choice was not a conscious decision.

5.2.4 Discussion and Conclusion

The task duration of the test pretty much equals the estimated time from the pre-test, hence the test worked as intended, apart from a bug that occurred once. Feedback indicated that the duration of the test was good and that participants did not feel any signs of fatigue, although the pairwise comparison was reported to be quite repetitive. The ratings itself, derived from both tasks have a similar average, top-rated pairs have a rating of (almost) 1, yet there is a significant difference for the minimum values (0.09 for pairwise- vs. 0.24 for spatial comparison). This could be explained by the algorithm which is used to infer weights in the spatial task: A zero rating is assigned only to the concept pair with the greatest distance. However, given that the participants dealt with five concepts in each instance, their primary focus was on establishing a structure that included all these concepts rather than strictly evaluating distances. In reality, they were unaware that distance was a factor in determining the ratings.

The inter-rater agreement was better than expected, because even pairs expected to be more controversial were rated quite consistent. Ultimately, the ICC and κ values may be context-specific, given their dependence on the proportion of genuinely controversial pairs to those with uniform ratings. Our choice of concepts aimed to balance this proportion, and, as evidenced by the broad range of variance values, the approach appears to have achieved its purpose. This experiment results in a consistent, weighted knowledge graph of ten concepts, whereas the weights are derived from participants rating the relationship of each pair. It can be used as a valid baseline to be compared with weights derived from other methods, like our spatial instances.

The primary question this experiment aims to address is whether spatial structure can be correlated with this simplified definition of knowledge. We developed a basic approach for spatial parsing that also generates a weighted graph, allowing us to compare the results with the baseline. It is crucial to acknowledge the limitations of this experiment:

1. Given the limited number of five concepts per instance, it is yet an open question how this scales up to more concepts, which lead to more complex, ambiguous structures
2. The parser results are not used as is, but are gathered to calculate an average over multiple instances of (potentially) multiple users
3. Knowledge is not static but changes over time, as spatial hypertext in general is emergent – this experiment is short-lived as users rate and structure concepts without the possibility to revisit or refine their output later in time

We are seeking to find that the ratings of both variants do correlate, ideally, with equal values. The MAE is higher than expected, but this can be explained by diverging minimum values, as already discussed, and the influence of controversial pairs. Addressing the latter could be achieved by recruiting more participants to stabilize the average ratings. Linear correlation of the ratings is evident, when solely examining Figure 5.7. Kendall's τ supports this impression and provides an estimate of how strong the relationship between the two values is. This can be explained because proximity is a major factor in spatial hypertext applications and information visualization in general (cf. Section 3.3.1). An interesting observation is the relationship between (relative) distance and weight. If we fix the linear model that describes the relationship between baseline rating and spatial rating, by normalizing both diverging minimums to zero, the gradient is almost exactly

1 (1.06). The spatial rating is computed on the basis of the (relative) distance, whereas we assumed a linear relationship between distance and weight. In simple terms, users tend to bring two concepts closer together in a linear manner as the weight of their relationship increases, and the same applies in the opposite direction.

This is a very important finding because it helps to provide visualizations as well: The linear relation can be used to translate a weighted knowledge graph to symbols with certain distances in a 2D space. To provide an example of this, we created a simple visualization for the knowledge graph comprising the ten concepts. This visualization employs weights derived from pairwise comparisons and integrates them into the physics-based composition described in Section 4.2.5. The composition is configured to map weights to the 'length' of the underlying spring model in a certain way. This particular image is generated with $d_{min} = 0$ and $d_{max} = 400$, while the diameter of an object is 60. The calculation is formalized in Equation 5.3 and for $d_{min} = 0$ it matches the rating calculation of the simple parser, but in the opposite direction (cf. Equation 5.1).

$$d(O_i, O_j) = (1 - R(O_i, O_j)) \cdot (d_{max} - d_{min}) + d_{min} \qquad (5.3)$$

Furthermore, Figure 5.9 shows a color gradient from light green to dark red to represent deviation from the predefined distances. For example, Baum (tree) and Photosynthese (photosynthesis) would be closer to each other if other objects in the space would not exert opposing pulls on them.

Although it was not an original goal to discover a relationship between the number of clicks and the time spent in a spatial instance, our data tend to confirm the emerging nature of spatial hypertext. A possible explanation, which is supported by the interviews, goes as follows: If users are unsure how to structure a set of objects, they follow an iterative approach of refining this structure. This refinement is not

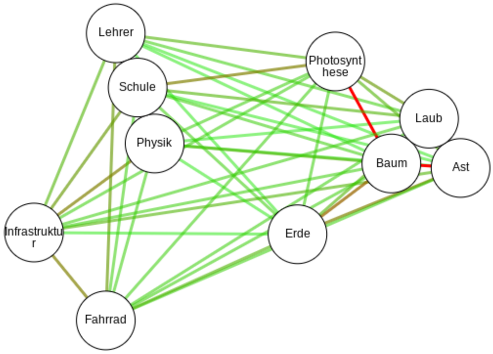


Figure 5.9: Simulated, force-directed layout of the gathered weight data

just conceptual, but is performed 'by hand', with users actually moving objects multiple times within the space. The system logs these modifications, providing an opportunity to observe and study the evolution of structures over time and potentially identify recurring patterns in how users make these adjustments. Furthermore, this process of refinement – if understood by the system – may be used to improve spatial parsing and other features of the system, Schedel's *temporal parser* [116] (cf. Section 3.2.2) is already hitting the same mark.

5.2.5 Summary

In summary, the test revealed a positive correlation of weighted/rated relationships derived by a pairwise comparison of concepts and by parsing spatial hypertext instances with five concepts each. This answers the question of whether spatial hypertext structures can be used to generate knowledge from, and furthermore, provides a possible approach to how this can be achieved. The linear correlation between object distance and baseline rating suggests that users tend to use distance to encode their interpretation visually. This finding can be used to implement algorithms that enrich a spatial hypertext instance with additional information (e.g. suggestions from a recommender system), because this linear relationship may work in both directions: parsing structure to compute weights and translating given weights to distances, as is done by the presented spatial composition algorithms.

Subsequent research could emphasize enlarging the test, by enlisting more participants, expanding the knowledge graph, and modifying the number of concepts displayed in a spatial instance. By analyzing the refinements users make to spatial structure in spatial hypertext interfaces, a more profound understanding of their interaction patterns might be attained.

5.3 Effects of Spatial Visualization versus Ranked Lists

> The user study discussed in this section has been previously published and initially presented as a poster paper with preliminary results [105], and subsequently as a full paper in conference proceedings [106]. The content from these papers has been updated and expanded with recent findings to be incorporated into this thesis.

The thesis posits that the composition of recommendations within spatial hypertext improves perception and aids sense-making. This is because organizing recommendations in relation to user concepts, coupled with their responsive behavior to user interactions, simplifies the comprehension of the knowledge structure. This arrangement not only visually contextualizes the recommendations, but also makes the underlying connections and relationships more intuitive and easier to grasp for users. Information retrieval and recommender systems often utilize spatial interfaces to visualize results and allow various interactions, e.g. to refine the shown result. User studies have already shown improved performance (efficiency, effectiveness, and satisfaction) of spatial interfaces compared to conventional one-dimensional list representations [65, 68]. However, little research has been done on describing how users interact with these interfaces compared to each other.

List representations serve as an effective baseline for comparisons because of their widespread use and familiarity among most users. Undoubtedly, for non-exploratory searches, they excel. This is because lists are typically organized by priority, placing the most relevant item at the top where it is easily visible and accessible to users. For search tasks with a strong focus on exploration, where it is difficult to formulate a query right at the beginning, we expect that a ranked list representation falls behind because of its lack of structural insight. At present, there is limited research exploring user interactions with spatial search interfaces, particularly in studies that compare these interfaces against non-spatial baselines for exploratory search tasks.

To gain insight into this matter, we carried out a between-subject user study involving 43 computer science students. The goal was to evaluate a specific spatial composition algorithm for recommendations. Additionally, our objective was to comprehensively document all user interactions, creating a detailed database that would facilitate a detailed analysis of the advantages, disadvantages and general differences. The spatial system works in a 2D space with a simplified physics-based composition (cf. Section 4.2.5). For comparative analysis, we also developed a second interface that presents results in a ranked list format. Both interfaces share a similar look and feel, functionality, and technical framework. This design choice was intentional, as our main focus was to understand how spatial composition influences user interaction with the system.

5.3.1 First Thoughts

Comparing two interfaces with inherently different interaction philosophies is difficult and probably not useful unless done thoughtfully. Although we are still interested in how users engage with spatially composed recommendations, our main objective was to understand how these interactions differ from those with a traditional list-based interface. To align our methodology and study design with contemporary research and to verify whether our results are consistent with others, we modeled our approach on studies with comparable objectives and interfaces, conducted by Klouche et al. [68, 67].

In [68], the authors compare a baseline search interface, based solely on a text input field for search and a ranked list for the results. In simple terms, the interface resembles the look and feel of most modern Web search engines. The interface of interest allows one to arrange query terms in a 2D interface, similar to what we called user concepts in the context of spatial hypertext. In fact, the interface provided by the authors can be called a spatial hypertext. It also shows the same ranked list as the baseline interface along the spatial interface. Within the spatial part of the interface, the items in the ranked list are represented as circles of varying size, indicating the general relevance with respect to the query terms; the position of those circles follows a map-based approach, in principle similar to what was suggested in VIBE [93] or the VIBE-based composition, detailed in Section 4.2.3.

These circles do not actually show the content of items they represent, but hint at how they are related to the query terms. By clicking in the space, the list is re-ranked from the 'perspective' of the click; thus, the approach is called "visual re-rank" for "multi-aspect" search. We can take two things out of this research. First, our exploratory search methodology shares similarities, though it is not confined to aspect-based queries only. The concept of aspects can be seamlessly integrated into our approach. Essentially, by employing an aspect-based knowledge base and restricting user interactions to these aspects, we can facilitate a direct comparison with the referenced study. This comparison aims to determine whether we observe a similar boost in efficiency as observed in their findings.

5.3.2 Knowledge Base

The knowledge base plays an important role in the technical setup of the study, as the quality of recommendations, whether queried spatially or in a baseline interface, heavily influences the user experience. If the system generates irrelevant or non-sensical recommendations, users end up struggling with the recommendation algorithm or data, rather than engaging in productive interactions with the search interface. Additional requirements include that the knowledge base should utilize aspects and straightforward query terms, rather than potentially intricate conceptual entities, and that it should offer content that is both valuable and helpful to explorational search tasks.

From the beginning, this thesis has been accompanied by the idea of integrating a knowledge base on sustainable and energy-sufficient techniques. This topic is not only interesting but also presents a good opportunity for exploration. Furthermore, it is a topic that most users already have some fundamental knowledge and associations, making it an ideal choice for an in-depth study of interaction.

For a meaningful and easy-to-replicate knowledge base, we utilize a method based on Wikipedia articles and the co-occurrence of nouns (used as aspects). In addition, this approach is easy to adapt to various other topics, thus simplifying the process of repeating the study with an alternative focus if required. The knowledge base itself is in line with the requirements mentioned in Section 4.1; a weighted, undirected graph structure.

In this test scenario, the knowledge base is built around the topic of sustainability, renewable energy and energy-efficient buildings. All aspects are extracted nouns from nine German Wikipedia articles, since the study will be conducted with German-speaking students only. The choice of the articles is important because they serve as a seed for the method described hereafter. They greatly influence the content/ topic of the created knowledge base and, as important, the size of the knowledge base in terms of number of aspects. The articles of choice were as follows:

1. Windenergie (wind power)
2. Passivhaus (passive house)
3. Fernwärmespeicher (steam accumulator)
4. Nullenergiehaus (zero-energy building)
5. Sonnenenergie (solar energy)
6. Methanisierung (methanation)
7. Power-To-Gas
8. Wärmedämmung (thermal insulation)
9. Energieautarkie (self-sufficiency)

Next, the seed-articles are used to compute a list of nouns that will then be used as entities in the knowledge graph and a set of related (linked) articles. To detect nouns, the part-of-speech tagging module (POS tagging) of the NLP library spaCy[4] is used. In addition to stemming the nouns, the algorithm keeps track of their *inverse document frquency* (idf) and *term frequency–inverse document frequency* (tf-idf). The resulting relations are determined by looking on (structural) co-occurrences. All articles, the initial seeds, together with the set of related articles, are segmented into the article itself, paragraphs, and individual sentences. Whenever two nouns appear together in any of these, a counter is increased with the following rules:

- Same article: Increase by 1
- Same paragraph: Increase by 2 (+ same article) = 3

[4] `https://spacy.io`

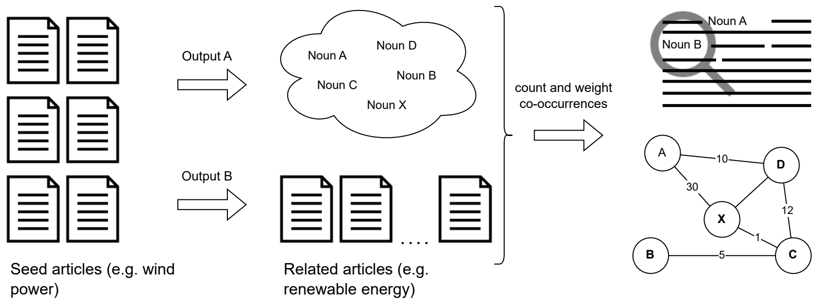


Figure 5.10: Stages of knowledge base generation with example seed and output graph; adapted from [104]

- Same sentence: Increase by 3 + same article + same paragraph = 6

In a final step, the counted values are log-scaled and normalized to fit the desired range from 0 to 1; nouns with a tf-idf below 0.2 are deleted to remove nouns without much meaning. The determination of the threshold in this context is derived from a series of qualitative experiments. However, given the nature of this study, it should be noted that the threshold is subjectively established and requires careful consideration. This methodology, although applied to a different topic, is also described in [104, 6]. Figure 5.10 gives an overview of the process. In total, 1046 articles were analyzed for this test's knowledge base, resulting in 2,179 aspects and 130,050 edges.

To generate relevant recommendations, the knowledge base is designed to accept any number of aspects. These aspects must be present within the knowledge base itself. Upon receiving these input aspects, an algorithm performs the steps in Algorithm 5.1[5]. In essence, the algorithm computes the relevance weight of potential recommendations based on the aspects in the query. This relevance weight is calculated as follows: for each recommendation, the algorithm sums up the precal-

[5]In line with the description in Section 4.1.

culated weights between each aspect in the query and the recommendation, and then multiplies this sum by the IDF of the recommendation. This approach helps to prioritize recommendations that are not only relevant to the query aspects but also significant within the overall dataset. This relevance weight is used to rank the list of recommendations, whereas the original edge weights between 0 and 1 are used to feed the composition.

Algorithm 5.1 Recommendation query

```
Input: query aspects q
function QUERY(q)
    map ← {}  ▷ Recommendation to weight map; defaults to 0 if key not set
    for all aspect ∈ q do
        recommendations ← GETNEIGHBORS(aspect)
        for all rec ∈ recommendations do
            map[rec] ← map[rec] + WEIGHT(aspect, rec) · IDF(rec)
    return SORT(map)
```

5.3.3 Test

We opted for an experimental design featuring both a spatial variant and a list variant. Due to the ongoing COVID-19 pandemic, these were tested remotely, using the same knowledge base, as already described, to ensure that the same recommendations were available in both setups. Our primary aim was to develop both variants with as much similarity as possible, ensuring that the only difference is in the composition and interaction mechanisms. Next, we will outline the system design and explore the details of both variants.

Two interfaces were used in the experiment, one based on an adapted physics-based composition (cf. Section 4.2.5), the other (i.e., the reference interface) that does not show recommendations in a 2D space, but in a ranked list. The participants were randomly assigned to one of the interfaces. By adding or removing recommendations, users can control the context; thus, alter what is recommended by the system.

Spatial parsing was deactivated for this study. Consequently, query generation is not based on visual structure, but instead involves searching for all recommendations related to any of the added aspects (cf. Section 4.3.2). This adjustment was made because we restricted users from modifying the positions of visual entities in the space, making spatial parsing superfluous. Thus, the emphasis was shifted to the representation of recommendations rather than user interaction with previously added aspects. Additionally, this approach simplified the comparison between the two variants. The implications for the physics-based composition and the adaptations that are made are detailed in the next section.

Spatial Variant

The spatial variant, as shown in Figure 5.11 spans a viewport that fills the user's screen. Some space on the left remains empty because this room is used by the list variant to show the ranked list of results. To ensure that users of neither variant do have any advantages due to different large viewports, we chose to leave this space empty in this variant. Note that the list variant utilizes a similar 2D space, but without composed recommendations. All aspects of the test, including interactions, composition, and browsing, happen within this specified frame. Colored rectangles represent aspects that were added by the user, with

- *green:* initial (given) aspect, which cannot be removed,
- *light blue:* added aspect, that can be removed and
- *dark blue:* as light blue; the most recent added aspect.

The recommended aspects are not colored and are composed with the algorithm described in Section 4.2.5.

As mentioned previously, possible interactions are restricted, particularly with dynamic behavior and responsiveness being disabled in this

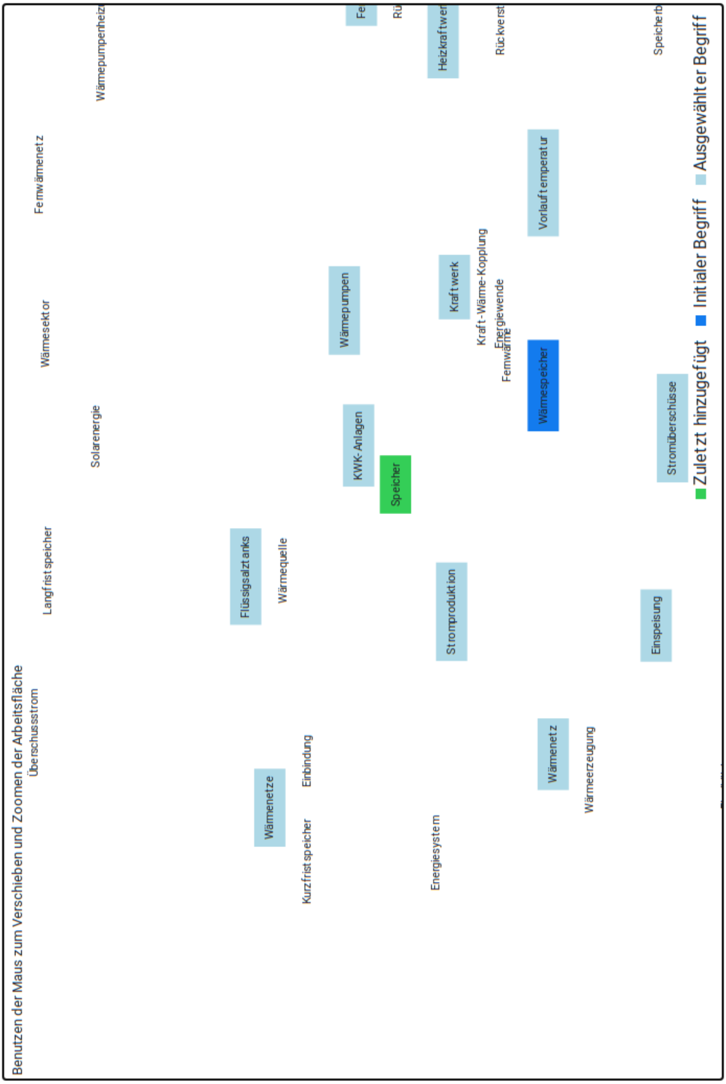


Figure 5.11: Spatial variant, shows recommendations within a 2D space

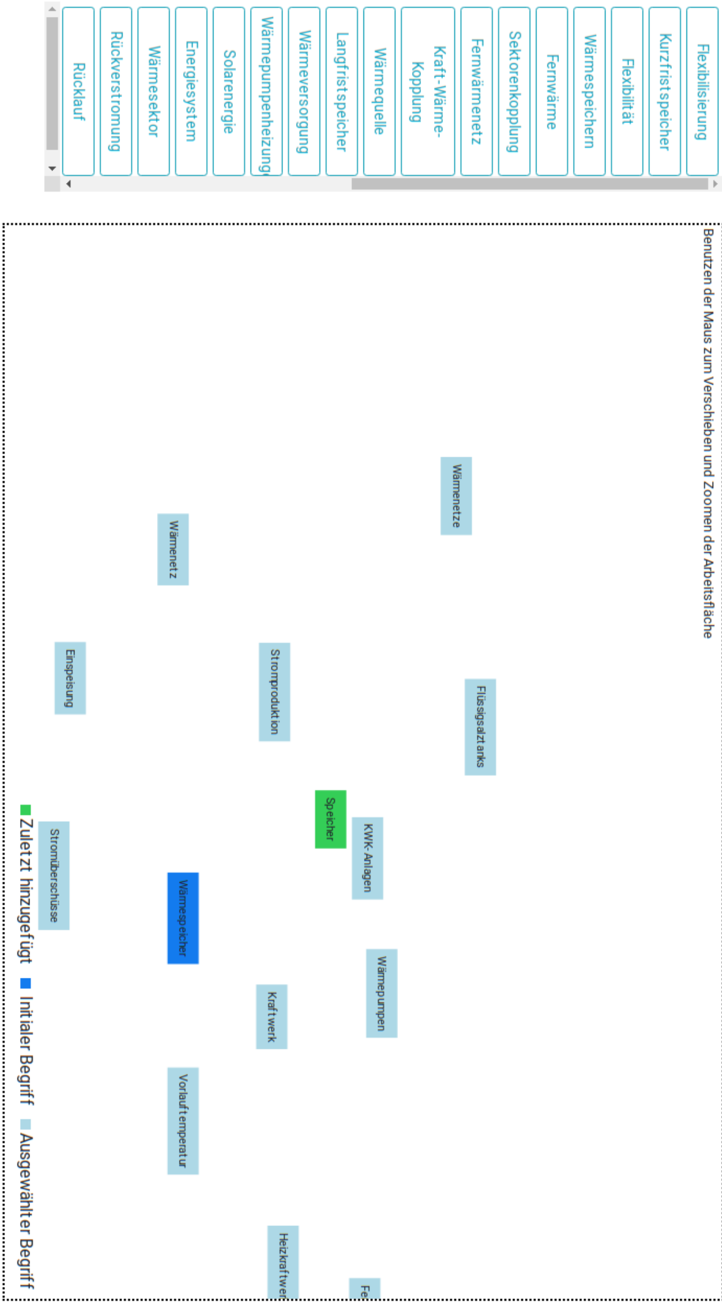


Figure 5.12: List variant, shows recommendations within a ranked list

test setup. Users are only allowed to add aspects to the workspace; this happens by clicking on a recommendation, and to delete already added aspects, again by clicking on them. This means that they cannot move their aspects to structure them according to their specific needs. We decided to simplify the proposed system for several reasons. In fact, the test omits one of the key features we identified as innovative and potentially beneficial for enhancing knowledge exploration.

First, the test should have a clear focus on the composed positions of the recommendations. Allowing the context to be altered would shift this focus. Second, our goal was to make both interfaces as similar as possible, in order to attribute any observed differences in interaction directly to the layout's composition or the ranked list format. Obviously, a ranked list would not benefit from a workspace where aspects could be rearranged. And, last but not least, due to the ongoing pandemic, we had to conduct the test online; as detailed later, on the hardware of the participants. Although factors such as screen size are readily measurable, allowing us to exclude participants with screens that are too small, hardware performance can also vary, potentially leading to inconsistent simulation outcomes. Conducting the test becomes much more repeatable and comparable if the composition results are computed on a server. In this setup, the test interface would simply display the static and damped outcome of the physics simulation, ensuring uniformity across different hardware configurations.

The composition on the server speeds up the simulation, because it is not necessary to run it in real-time. Still, the simulation runs with 60 steps per simulated second, but terminates after ten simulated seconds, as after this time, the composition mostly shows itself in a damped state, without any further movement. Aspects added to the system are placed at the position where the recommendations were composed. Both the addition and deletion of an aspect trigger an update of the context, and thus a new query. In response, the knowledge base generates a fresh set of suitable recommendations which are then composed according to the method previously described.

The phyiscs-based composition has no means of determining the number of recommendations that fit to a given context. Therefore, we set up a heuristic to limit the maximum number n_{max} of recommendations shown. This heuristic applies both for the spatial variant and for the list variant and depends on the number of query aspects n_q; see Equation 5.4 for details.

$$n_{max} = n_q \cdot 3 + 10 \tag{5.4}$$

The equation is derived from experiences already made with such interfaces and reflects that a growing number of spatially represented keywords leads to more space, which can be occupied by recommendations.

The viewport of the space can be altered by panning it in the direction of choice (it is unlimited) or by zooming in and out. Both actions can help to focus on certain aspects or to gain a better overview of the recommendations and query aspects. They are activated using a computer mouse (or trackpad), through a gesture of moving the pointer while keeping the primary (mouse) button pressed, and by using the mouse wheel or equivalent trackpad gesture.

List Variant

The list variant presents the baseline system for comparison and is very similar to the spatial variant, as can be seen in Figure 5.12. As noted above, the idea of using a ranked list is inspired by the baseline interface of [68], where the queried information is also shown in such a list. The following overview should help to differentiate both variants.

Recommendations are not rendered in the 2D space, yet the composition is calculated as in the spatial variant. First, this is done to have comparable reaction times of the server system: the composition, which needs to simulate ten seconds of physics, usually took around a second on the available hardware, thus long enough to (negatively) impact the test results, if not compensated. Second, we decided to leave the 2D space as part of the list variant, Although it does not show recommendations, it displays the very same map of aspects; therefore the composition is needed anyway.

Instead, recommendations are displayed in a ranked list, with the most relevant recommendation (cf. Algorithm 5.1) at the top of the list. As in the spatial variant, the set of displayed recommendations is updated when the context changes, i.e. a recommendation is added, or an aspect is removed. The number of recommendations displayed is also identical, as calculated in Equation 5.4. The ranked list occupies the space adjacent to the 2D area and introduces a scrollbar when the number of recommendations exceeds the viewport capacity.

The list variant offers the same 2D space as the spatial variant. It shows the aspects that contribute to the query. In both cases, the positions of the aspects follow the composition algorithm to help the user get the same meaningful overview. However, there is a notable difference in the way the two variants are used. In the spatial variant, users can immediately discern the location where a new aspect will appear, as it is the same as the recommendation's position just a moment earlier. Conversely, when a user adds an aspect by clicking on an entry in the list, they would not intuitively know where this aspect will be positioned, for instance, if they wanted to remove it right away.

To compensate for this disadvantage, the 2D space is panned automatically when an aspect is added, such that the recently added aspect is in the middle of the spatial view; without altering the zoom. It is important to note that users of the list variant can ignore the spatial view at all if they do not want to remove keywords or keep track of their query aspects.

Participants

We recruited 43 computer science students (11 female, 30 male, and two not specified) with a mean age of 22.6 (ranging from 19 to 30) and asked them to participate in a voluntary user study at the beginning of a beginner-level lecture. In total, we repeated this setup three times with three different lectures and different students. We ensured that no student could participate more than once. Before collecting any data from the participants, informed consent was obtained. Participation was optional, but motivated by the opportunity to get to know an interesting application domain. Students who opted not to participate faced no negative consequences. However, they were requested to avoid disrupting the test. This request was also extended to participants who had completed the test or discontinued it before finishing.

Upon starting the test session, all participants received a unique code to delete or request their collected data from the authors' record. No compensation was given. Subsequently, the results of the study were analyzed and discussed in another lecture.

Procedure

The ongoing COVID-19 pandemic required a departure from a traditional laboratory setup for the study. In response, we developed a Web-based test client, which was introduced and demonstrated to the three groups of participants using Zoom[6].

The study was integrated into the first 30 minutes of a lecture. Students were informed about the opportunity to participate in the user study and were reassured about its voluntary nature. Once the details were clarified, a link to the test was shared in the Zoom chat. Clicking on this link led the students to a page where they received their test instructions. This step was important as the subsequent page provided an explanation of how the interface functioned, tailoring the informa-

[6] A video conference tool used by the university to handle remote lectures.

tion to the specific interface variant to which each participant was randomly assigned. The written instructions were kept as consistent as possible between both variants, differing only in aspects specific to the differences highlighted in Section 5.3.3.

Then a verbal introduction to the task was provided. The test started once the participants felt confident in their understanding of the interface, had given their consent, and clicked the "next" button.

At the end of the task, participants were invited to answer questions about their demographics (such as gender, age, daily computer usage, and domain knowledge) and to offer feedback on the interface they used. Participation in this feedback section was, as the whole test, entirely optional. This approach ensured that, despite the lack of a physical laboratory environment, the study could proceed effectively, utilizing digital tools to interact with and guide participants remotely.

Task

After the participants gave their consent, the system randomly assigned them to either variant of the interface. Independent of the interface, participants were asked to perform the same exploratory task. The task had a strong focus on wind turbines and their environmental influence; thus, we ensured that the prior knowledge about that topic among the participants is comparable. Similarly, we asked participants to estimate the time they spend on the screen per day in front of a PC or laptop. This was based on the assumption that participants who are used to working with a variety of user interfaces might adapt more efficiently to an unfamiliar interface. Given that only a brief introduction was provided and no extended period was allocated to 'learning' how to use the interfaces, familiarity with various interfaces was considered a potentially influential factor.

Regarding these two variables, we did not discover significant differences. Among the 21 participants assigned to the spatial variant and the 22 participants assigned to the baseline interface, most self-rated their previous knowledge as average ($mean = 0.16$, $SD = 0.8$). This assessment was based on a scale ranging from -2 to 2, where a score of 0 denotes an “average” level of prior knowledge. The mean screen time was around 9 hours. Obviously, the selection of computer science students as participants, coupled with the prevalence of online lectures during the pandemic period, likely contributed to this high number. In conclusion, both their study field and the prevailing circumstances suggest that they are more used to interacting with a range of user interfaces than the average individual.

The task was provided within the introduction of the interface, written in German. For this thesis, the task is translated into English. Participants were asked to imagine the following situation:

> There are plans to build several new wind turbines (German: “Windräder”) near your residence. Find as many potential advantages, disadvantages, or other personally relevant topics that you want to ask questions about at a soon-to-be-held citizens’ meeting.

It is important to note that the task does not contain any explicit time limit, other than the 30 minutes that were contributed to the test. In a pre-test, this amount of time proves to be far more than needed by any pre-run. This decision was intentional, as the nature of an open exploratory search task is better suited to be concluded when the user feels satisfied with his or her findings. As a result, the task was deliberately designed with a broad objective, allowing users to explore a wide range of topics at the surface level or to dive deeper into fewer topics according to their preference. The task is completed when participants have the impression that they have gathered enough information and topics for the citizens’ meeting. Then they should write down the topics about which they plan to ask questions.

There are several reasons for suggesting a task like this. The construction of new wind turbines represents a prevalent real-world situation. In the context of the regional setting of Hof University, numerous wind plants have already been established, with plans for more in the pipeline. These developments frequently involve citizen collectives, who often hold strong opposing or supportive opinions on the matter. In fact, within the Hof regional area, there is the highest installed capacity per square meter in Bavaria (2023), as reported by Salvati [112]. Therefore, students should find it relatively easy to envision themselves in this scenario. Furthermore, the topic is approachable, raising obvious questions about e.g. (construction) noise to the point of more (obscure) concerns about subsonic noise and its influence on wildlife. This open nature allows participants to set their own focus, supported by recommendations of the system.

Like the task itself, the knowledge base responsible for the recommendations provides them in German, since the participants were German native speakers. Regardless of the interfaces, the system initiates the search task with one given aspect that cannot be removed. This was done to show recommendations right from the beginning.

Measures

During the test, any interaction with the interface is recorded and stored on the server, so a detailed replay of each user session can be created. In addition to a detailed analysis of interactions, we wanted to analyze the effectiveness, efficiency, and satisfaction of the users. In the context of this study, effectiveness is measured based on the number of topics written down by the participants after the search session. Therefore, the effect is subjective and varies from the point of view of each individual participant. On average, this subjective influence is considered negligible. Efficiency is defined as the task duration in seconds, starting with the first interaction in the search interface and ending with the last interaction. Satisfaction is the subjective assessment of how helpful the system was, on a scale from 0 to 10 ("not helpful at all" to "very helpful"). Additionally, the objective was to characterize how participants interacted with each variant of the interface. To achieve this, we tracked two types of actions: *navigation* and *interaction*. Navigation involved counting the number of times participants panned and zoomed in or out. A pan is initiated when the primary mouse button is pressed and is finished upon its release. Zooming begins with the use of the mouse wheel (or trackpad gesture) and ends after its last usage, provided the wheel is not used again within one second. Interaction was measured by the number of times that participants added or removed aspects in the interface. This approach allowed us to quantify and compare user engagement with both interface variants.

To be able to analyze the data in relation to the current state of the exploration (context), the system records the following:

1. All events, along with a timestamp
2. List variant: The position in the ranked list of an added recommendation
3. For any removed aspect, in both variants: Number of other aspects that were added before this one got deleted, to determine if users tend to remove aspects they just added (revert) or those which are in the space for a longer time (progress)

(1) and (2) are later called "position" and refer to the position in the list and the position on the stack of added aspects.

5.3.4 Results

The presentation of the results is divided into two sections. The first shows the evaluation of satisfaction, effectiveness, and efficiency. This involves compiling the collected data and, for example, calculating the average values. We then examine the data for statistically significant differences between the two interfaces. The second section focuses on a detailed evaluation of user sessions over time. This includes e.g. examining user behavior according to the progress of the task. In addition, we discuss whether the discovered effects may be the reason for the observations outlined in the first section.

Aggregated Evaluation

The results for both interface variants are summarized in Table 5.4 and Figure 5.13. When interpreting the data, it is important to keep the effectiveness measure in mind. It is crucial for this evaluation as it makes the other values comparable to each other. For both variants, the participants chose a very similar number of topics, around three on average. Obviously, there are various topics, and it is not completely sufficient to quantify them solely on their number. Therefore, we evaluated the chosen topics that showed the expected range from simple concerns about noise to the danger of interfering with wind flow in the larger area. The quality of the topics is hard to quantify; but the estimation of the author suggests that the chosen topics are similar, regardless of the assigned interface.

Examining task duration, which reflects our efficiency metric, we noticed a notable improvement with the spatial interface compared to the baseline variant. The mean task duration decreased from around 4 minutes (list) to just over 2.5 minutes. The fastest user in the spatial variant finished in just 16 seconds (compared to 66 seconds for the list), while the longest completion time recorded was 577 seconds (slightly less than the list's maximum of 593 seconds). It is important to note that this measured duration begins with the user's first interaction, meaning that users could familiarize themselves with the interface first (e.g. by panning or zooming) before actively engaging with it.

Navigation actions – panning and zooming the viewport – were used more frequently by those assigned to the spatial variant. This makes sense as it was necessary for them, in order to interact with the recommendations. Users of the baseline interface, instead, had no necessity to navigate the space at all, because they could interact with the recommendations from the ranked list; still, it was used to some extent. Indeed, there was a list user who did not use the pan or zoom functions at all during the session; he or she spent 260 seconds, adding 17 keywords and deleting one.

In general, users assigned to the baseline interface had more *interaction* actions, which means that they added more recommendations as aspects and removed more of them, respectively. However, the recorded data do not show statistically significant differences (Mann-Whitney: $p = 0.089$), with respect to a predefined significance level of $p < 0.05$. A possible explanation could be that due to spatial composition, recommendations of interest are easier to identify, leading to fewer false positive aspects.

The mean satisfaction (“How helpful was the interface?”) was slightly higher for users of the spatial interface. Again, this is neither statistically significant nor did we expect huge differences here. The functionality and recommendations were identical in both variants, while the design of the study conditioned users to only know their assigned interfaces. Thus, they were unable to judge based on a comparison, but only on their assigned variant.

Feedback received at the end of the test sessions was invaluable in gaining a clearer understanding of the participants’ experiences with the interfaces, the task, and the data. The following quotations are translations from the original German responses provided by the participants.

Two participants suggested improving the introduction to the interfaces, possibly with a “video tutorial or a tutorial button”, or by “using images, less text”. This need was especially felt by those using the list interface. One user mentioned “the number of suggestions got overwhelming over time, making it difficult to make correct associations”, while another found “the list became too extensive and confusing at

Table 5.4: Summarized results for list and spatial variants; bold values indicate significance: * $p < 0.05$, ** $p < 0.01$ (Mann-Whitney)

	List (L)		Spatial (S)		L vs. S
	M	SD	M	SD	M-W (p-values)
Chosen topics	2.5	1.5	3.3	2.8	0.5662
Task Duration (s)	251.6	147.2	162.5	133.4	**0.02265***
Helpful rating (0–10)	5.7	2.0	6.2	1.9	0.2504
Navigation (pan+zoom)	21.8	27.1	50.2	37.6	**0.000741****
Interactions (add/remove)	25.3	12.4	20.3	14.8	0.08811

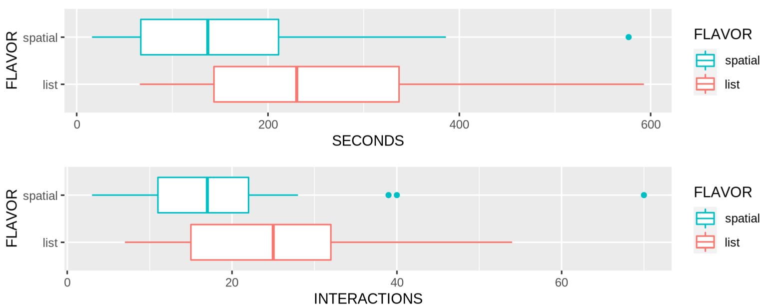


Figure 5.13: Task completion time in seconds and interactions for spatial and list variant.

the end". One user of the list interface wrote that "[a]fter some time, the number of recommendations increased, which could not be associated correctly". Another list user wrote: "After selecting multiple topics, I lost track of which additional keywords I could add. Maybe it is possible to work with umbrella terms" and someone else: "Building recommendations upon selected aspects works quite good, but it is difficult to come back to a certain topic, e.g. if current focus is on noise pollution and you decide to dig further into animal welfare".

In contrast, users of the spatial interface did not report such confusion. Instead, they requested more feedback on the changes in the recommendations composed, such as "providing a connection between selected aspects [...], to trace how you got to the current state", "the user could be guided to not yet shown recommendations, especially if they are outside of the current viewport" and "because selecting new aspects refreshes the interface and recommendations, you lose aspects you wanted to select next". However, they found "topics are evident by their proximity to each other".

Regarding color usage, users of the spatial interface appreciated the current color scheme ("the color coding was good", "I liked the color design") but suggested the interface "[...] should bring in a little more color because otherwise it becomes monotonous".

The participants' opinions on the relevance of recommendations varied widely, regardless of the interface they used. Responses ranged from "some suggestions were not very helpful" to "I would use [this application] for personal use".

Session Evaluation

An exploratory search is a dynamic process: There is a continual change in the development of a user's thoughts and associations during the individual progress in their search journey. In the previous section, we looked at the aggregated data to look for differences with respect to the defined measures. This analysis reveals e.g. a better efficiency for the spatial interface variant. However, this leaves unanswered whether this was beneficial across all phases of the search or merely an boost that yielded an overall advantage, for example, at the start. To shed some light on this question, we had a detailed look at the measured time between interaction and navigation events and set this into relation with the number of aspects in the space – an indicator of the exploration progress.

To clarify how the following figure must be interpreted, consider the following example. If a user has five query aspects and needs 20 seconds to add the next aspect or to remove one of the five in the space, we add a data point with $N(5) = DURATION(20)$. If this is done for all participants, grouped by their assigned interface variant, the result looks as in Figure 5.14. As individual data points would clutter the plot, it only shows box plots without outliers for each N and loess curves to ease the readability.

Some important notes for interpretation: For each N, there can be more $DURATION$ values than participants, because the participants can remove and add aspects as often as they want; thus, each participant can contribute one or more data points. To assess the influence of deleting aspects, we will later look at how often aspects were removed from the context. Furthermore, the more aspects were in the space (N) the less participants influenced the result, because many users finished without having as many as 44 (max N) aspects added.

Indeed, the first issue is mitigated by the fact that deletion was used sparingly by the participants of both interface variants. The numbers are shown in Figure 5.15. Approximately 13.4% of all interactions are deletions that spread evenly among the N values. The issue of missing data for high values of N cannot be neglected. To honor this shortcoming, the plot features an interval of confidence (0.95) for the loess curves. Approximately half of the participants with the baseline interface finished their session with 16 added aspects, while half of the

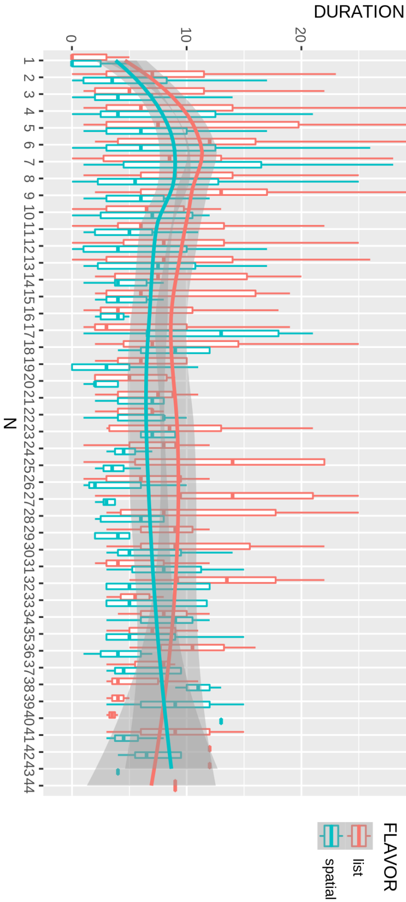


Figure 5.14: Duration between two interactions (adding/deleting), by number of already added aspects (N)

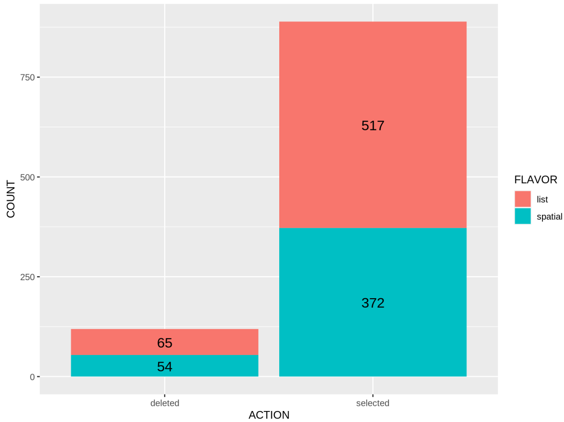


Figure 5.15: Number of ‘deleted’ and ‘added’ interactions by test variant

users of the spatial variant finished with 12. Only some outliers reached a total of more than 40 aspects. Details are depicted in Figure 5.16

When looking at the collected data, the interface with spatial composition constantly (mostly) yields shorter duration times between interaction events. This difference stays quite constant as well, with an exception at the very beginning, where users seem to have their first interactions at the same time; and the end, where missing data makes the interpretation difficult anyway. This parallel development of the duration time for both interface variants is very notable, because the loess curves do not follow a (simple) monotonic function, but show a very specific pattern: They start almost at the same point and show a rising duration time until there are about 7 aspects in the space. Right after, when more aspects and recommendations populate the interface, the curve flattens and stays at an almost constant level.

We believe that these curves demonstrate a dual effect: a learning or training effect, indicated by a flattening trend, alongside an effect of escalating complexity, marked by an initial rise, occurring as more aspects are added, and consequently, more recommendations are received. It is

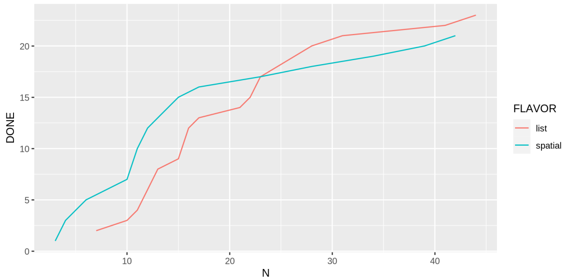


Figure 5.16: Number of participants, which completed the task (done) by the number of keywords they had added by completion (N)

not easy to determine which part of the system contributes to these observations. Our expectation was that users would predominantly skim the recommendations and determine their subsequent actions in the process. Similarly to Figure 5.14, we examined the frequency of navigational interactions (such as panning and zooming within the space) in relation to N. However, this analysis is subject to the same limitations as previously mentioned.

Figure 5.17 shows the result of this analysis. To ease the interpretation, the figure features two loess curves only, one for the spatial (blue), the other for the list (red) variant of the interface. The blue curve follows the same pattern as the duration time between interactions in Figure 5.14. It rises from the start and reaches its peak when around seven aspects are added to the space. Thereafter, the curve flattens to a constant level. This indicates that the participants spent their time browsing/exploring the results by navigating the space. However, it also suggests that they did not invest additional time or effort when presented with more recommendations. On the contrary, there is less exploration beyond seven aspects, or rather 31 recommendations (cf. Equation 5.4).

For users of the list interface, the chart presents a different picture,

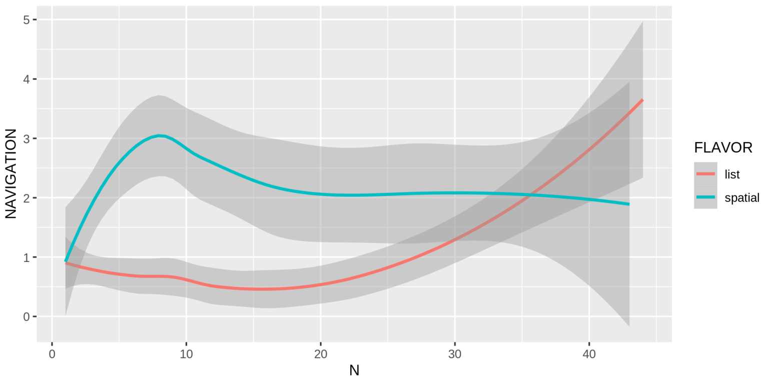


Figure 5.17: Number of navigation events (zoom/pan) between two interactions, by number of already added keywords (N).

but this distinction is primarily superficial. The activity of 'browsing/exploring the results' occurs within the ranked list, not within the spatial overview of the interface. In numerous instances, navigation within the space was not utilized at all. It is probable that these users allocated their time to scrolling and reading the list, similar to how spatial users navigated within the space. Unfortunately, we did not capture data related to these scrolling events.

This leads to the conclusion that only a certain amount of the recommendations, may they be spatially composed or ranked in a list, are perceived and processed by the user before he or she is going on with changing the context. In our very specific setup with its application domain and interface design, this threshold appears to be reached with about ten aspects (cf. Figure 5.14) and 40 recommendations, respectively. The fact that this value is consistent across both interface variants could simply be a coincidence. To draw a clearer distinction, it would be advisable to consider other spatial composition algorithms. It would also be logical to modify the correlation between the quantity of aspects and the number of recommendations displayed.

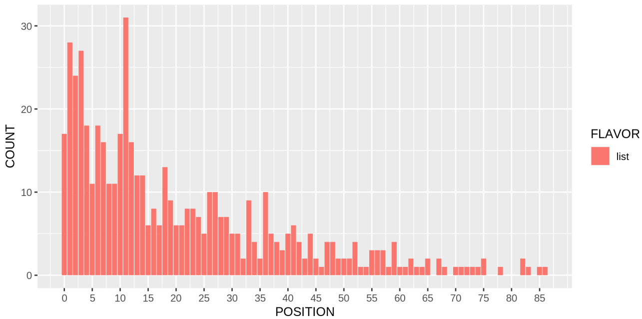


Figure 5.18: Position of added recommendations in the ranked list

Keyword Positions

Position, as already defined in Section 5.3.3, refers to two factors. The ranking of recommendations within a list and the placement in the stack of previously selected aspects when removed from the space. The findings indicate a clear preference among users of the list variant for recommendations that are visible without the need to scroll. The system tracked the resolution of the participants' Web browser window to ensure that it matches our design specifications of displaying at least 18 recommendations in the list before necessitating a scroll bar. This preference and its distribution are shown in Figure 5.18. Besides avoiding to scroll, this indicates that the most fitting recommendations can indeed be found at the top of the ranking.

When users click on a recommendation, in the list or in the space, it is added and composed within the space. For participants assigned to the baseline variant, the system automatically moves the viewport to focus on the new added aspect, in this case. To remove an aspect, an added aspect can be clicked again. The system organizes the added aspect in a stack and records the position of an removed aspect within this stack. If the user removes an aspect he or she just added, the recorded position is 0. When there are e.g. five more added aspects in the meantime, the system records a 5.

Table 5.5: Position of removed aspects

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
List	0	0	2	4.50	6	25
Spatial	0	0	2	10.33	14	41

Table 5.5 presents the quantiles and average values of this position metric. The data show that in 50% of the instances, users opted to remove aspects that were added recently. Interestingly, spatial users exhibited a tendency to remove older aspects, in contrast to list users who generally focused on more recent additions.

5.3.5 Discussion and Conclusion

First, it is important to align the findings of this test with those of Klouche et al. [68], whose setup and comparison inspired this study. The most prominent finding was that their interface, which offered a spatial view of the recommendations and the ability to use this view to re-rank the recommendations had a positive impact on the efficiency (task duration). This matches the results of this study, which show a similar, yet slightly smaller, mean time gain of 58%.

Although this study aimed to minimize influencing factors, pinpointing the exact cause(s) of this observed effect remains challenging. The spatial view utilizes a 2D space for its spatially composed recommendations. Ideally, this arrangement should be meaningful, enhancing the understanding and perception of relationships and assisting in the exploration and comprehension of the presented knowledge structure. However, positive effects may be influenced by the fact that the 2D representation allows users to recognize more recommendations at once. This may be amplified by altering the viewport; navigation events were used more frequently by users of the spatial variant. Is there a positive effect of the composition at all?

This question cannot be answered directly. To do so, a future study should use a different baseline interface. One possible approach could be to utilize the same composition algorithm, but to randomly distribute the recommendations to a certain bounding area. In this way, both interfaces would present a consistent composition, featuring the same 'recommendation density', yet only one utilizes *meaning* as defined within the context of spatial composition. However, the recorded data help indirectly answer the question. For instance, we can consider the number of deletions and additions as an indicator. Given that the general outcome (effectiveness, satisfaction) is the same – which is the case – the user group that needed to revise their aspects more often seems to struggle with the exploratory part of the task. Both interactions were used more frequently by the baseline group. Although the study must negate statistical significance ($p \approx 0.088$), the numbers provide some evidence and are reasonable, as adding and removing aspects is a more time-consuming task than navigating the space. However, future studies should focus more on this question.

As already mentioned, both the satisfaction assessment and the effectiveness of the spatial variant are as good as for the list variant, with a minor improvement tendency for both measures. For satisfaction, this was an expected result. Both interfaces proved to be user-friendly and the task at hand was effectively manageable with either variant. Consequently, any differences in satisfaction should primarily be based on the recommendations of the system. Given that these were identical across both interfaces, it follows that the satisfaction assessment should also be similar. However, the use of 'chosen topics' as a variable to measure effectiveness is not a perfect solution. The open-ended nature of the task allows this variable to be affected by factors such as the participants' personal preferences. To strengthen the validity of the claims presented, future research should incorporate a task specifically designed to address this issue, thereby providing more substantial evidence. A possible approach would be to define a goal that is harder to reach and not subjective if it is reached – while keeping the element of exploration.

Written user feedback revealed some interesting insights and shortcomings of the provided interfaces. Some participants noted that they were slightly confused with the interfaces and how they should be used in the first place. Although there was an introduction and an opportunity for questions, it makes sense to give users more time to become familiar with the interface. However, the aforementioned results can be seen in this light; especially, the functionality of the spatial interface is not very common in average daily computer usage. This raises the question of how both interfaces would perform in a longitudinal study with well-trained participants. For this study, we decided to simplify the knowledge base to aspects, omitting potentially more complex concepts. This eases the creation of the knowledge base and opens references to aspect-based search, in general. Also, the interfaces are indeed simpler because they only need to display simple German nouns. However, the system also needs to demonstrate its capabilities in more complex settings. Some features of the composition were excluded, for the reasons mentioned in Section 5.3.3, especially dynamic behavior with the goal of making (positional) changes traceable. The user feedback received indicates that a significant source of confusion was caused by unpredictable updates within the interface. A responsive composition could have improved this shortcoming, leading to better clarity and experience. Spatial parsing was also disabled, primarily due to the absence of a feature that allows for the movement of aspects. Having a structural understanding could facilitate the provision of recommendations that are more accurately tailored to the context.

The session evaluation sheds some light on the details during the search session and helps to reason about the discovered final numbers. For example, the fact that users of the spatial interface variant had a higher pace throughout the entire session shows that the improved efficiency is not just influenced by certain aspects at a certain stage of the exploration process. Users appear to find the information they want more quickly within a meaningful 2D layout. This advantage, we believe, may diminish for search tasks that are less open-ended and exploratory in nature. The reason for this is that the strength of a ranked list lies in its ability to prominently display the most relevant recommendation at the top. This straightforward presentation is effective for tasks where the objective is to quickly identify the best options, rather than explore a broad range of possibilities.

An interesting alternative for list users could involve a different visualization of the aspects they have already added to the search context. In the experimental setup, aspects were managed in the same way as for spatial users. However, a drawback of this approach is that it requires users to navigate the space to gain a comprehensive overview of all added aspects – a task that list users engaged in less frequently. To better tailor to their preferences, list users might benefit from a more streamlined view, where their selected aspects are displayed in a simple, easily accessible list or table format. This adjustment could improve the user experience by reducing the need for spatial navigation and providing a clearer and more direct overview. In general, this user test offers many possibilities to study further aspects in more depth. Utilizing this variety of variants will provide more in-depth insight into the effective features of the interfaces.

The progression of navigation events and interactions (add/remove aspects) shows how participants used the interface. Those who worked with the spatial variant spent most of their time browsing/exploring the recommendations provided. This becomes apparent as the recorded navigation events indicate this exploration. Watching the session replays reveals that, in the style of the Visual Information Seeking Mantra (cf. Section 3.3, users follow a natural loop of reading ("overview first"), navigation ("zoom and filter"), and context change ("details-on-demand"). In the case of the list interface, there is no direct equivalent to navigating; the closest match is scrolling through the list, something that was not recorded and may be misleading, because of varying window sizes. There is a need for a better navigation measure. A potential strategy could involve the use of eye-tracking technology to identify phases in which users primarily focus their mental efforts on examining the recommendations presented to them.

The *position* of added and removed aspects in the list variant is as expected. Aspects with high relevance with respect to the query, thus at or close to the top of the list, were added more often than those at the end of the list. It is also evident that there is a distinction between positions that are visible without scrolling and those that require the user to scroll down. A reason could be that users tend to avoid too many interactions in the form of scrolling before making a decision. Consequently, they remain on the first page of the list. The recorded data do not answer the question whether this behavior applies for the spatial interface as well. Users probably rarely move their viewport to a completely different location. Composing (locally) relevant recommendations near by amplifies this.

Users tend to remove aspects they have added recently. This makes sense, as new aspects can alter the direction of the resulting recommendations in ways that may not be desired. This pattern holds for both interface variants, but it is observed that spatial users delete aspects that have been present in the space for a longer duration more frequently. As noted previously, list users lack a comprehensive view of the aspects they have added unless they navigate through the spatial view. Therefore, it is reasonable to expect that list users would remove older aspects more frequently if presented with them in a straightforward, easily accessible overview. This would allow them to better manage the influence of these aspects on the recommendations they receive.

Summary

The between-subject study with 43 computer science students compares a ranked list interface and a spatial composition for recommendations. Both interfaces build the frontend to an aspect-based knowledge base about sustainability energy, whereas the list variant represents the baseline for the spatial interface, which features a physics-based composition (cf. Section 4.2.5). The participants completed a task centered on an exploratory search.

The most notable differences observed in the study were in terms of efficiency (measured by task duration) and the extent of navigation within the space. A detailed analysis of session data over time revealed several intriguing insights. These included learning effects and the apparent existence of a threshold for the number of recommendations users are willing to read, a limit that seems unaffected by the type of composition used. However, through the discussions, we identified opportunities for additional research in the future. Certain questions remained only partially answered, suggesting the need for further exploration. The implemented system allows the testing of various adaptations and versions of ranked lists or spatial compositions, offering a rich ground for continued investigation.

The study supports the claim that spatial composition yields beneficial effects, although the spatial interface implemented utilized a reduced set of features. User feedback highlights the importance of compositions which dynamically adapt the changed context in a traceable and smooth way. Spatial parsing as a means to derive more effective queries, along with the ability for users to organize their aspects within the space, could potentially exert a more pronounced positive impact.

6 Discussion

The last chapter summarizes and discusses the results of the research objectives given in the beginning of the thesis. This is concluded by a section on possible future work that may advance the field of spatial composition.

6.1 Summary

This work examines the composition of recommendations within a spatial hypertext interface. Incorporating recommendations makes the whole application a recommender system, with a focus on the specialized, spatial user interface. In this context, composition means the control of visual properties concerning said recommendations, most and foremost their position in the space and their dynamic behavior when the composition needs to adapt to a new situation due to the user changing the spatial hypertext. As recommendations are retrieved from a knowledge base, the thesis provides a simplistic technical definition of knowledge. It is represented as a weighted, undirected graph between domain-specific concepts. A user study designed to investigate the connection between user-generated spatial arrangements of concepts and weights within a knowledge graph revealed a linear correlation. Specifically, the study found that the distance users placed between concepts in a spatial layout directly corresponded to the weights they assigned to those same concepts when presented in a pairwise manner.

This discovery underpins the implementation of five composition algorithms, which inner-working is based on a relative or absolutely measured distance between visual symbols, representing concepts and recommendations. These five composition algorithms are explained in detail and show the iterative development process of spatial composition. Dynamic behavior, space utilization, and meta context are challenges discovered when developing composition algorithms. The algorithms were evaluated in a comparative way to provide a basis for further improvements and completely new approaches.

Two of the five algorithms presented, the physics-based implementation and the pearl necklace composition, are further used and examined in various research prototypes that employed the theoretic concept of spatial composition on a set of real-world problems. These serve as case studies that were used to discuss the interface and spatial composition with experts in the domain. Qualitative feedback influenced the functionality of the entire recommender system, the latest development iteration was demonstrated in Section 2.5

A user study was conducted with computer science students to examine common user experience metrics for a recommender system with spatial composition compared to an interface that renders recommendations in a ranked list. The setup was inspired by a similar study design by Klouche et al. [68], which they used to gather the same metrics for their proposed interface. As in the original study, the interface with spatial composition leads to improved efficiency, as participants completed the given task in a shorter time. Other metrics, such as user satisfaction and effectiveness, remained the same in both variants.

To be able to contribute this discovered effect to the spatial composition and to gain a better understanding of how humans use such an interface, the data of the same study were used for an in-depth analysis. This revealed, among other findings, that increased efficiency is indeed an effect that can be measured throughout the task and not only at a certain state. Additionally, a limit was identified regarding the number of recommendations users reviewed before moving on to their subsequent action, observed across both interface variations. In summary, the in-depth analysis of the session data indicates that users engaging with the spatial composition-based interface were able to navigate and explore the knowledge base more quickly by utilizing the provided spatial arrangement. Future research directions are also provided, which is recapitulated in the next section.

In addition to the primary contributions, this thesis also sheds some light on the overarching system that supports the spatial hypertext interface and the recommendation system. It explains how Wikipedia can serve as a source for generating domain-specific knowledge bases for the system in question. Furthermore, it outlines how the output of spatial parsing can be leveraged to infer knowledge and formulate queries, thereby enabling the provision of recommendations.

6.2 Scientific Contribution

As stated in Section 1.2.2, the VKB system [119] represents the most significant recent effort to integrate spatial structure as an essential part of a recommender system. The practical contribution of this work lies in its description of a comprehensive system that integrates a spatial hypertext user interface along with a recommendation engine. The scientific contribution is evident in the theoretical framework for spatial composition, the detailed algorithms presented, and the research conducted on how distinct elements of the system affect user interaction with it. The highlighted research gap was addressed with three hypotheses and two types of analysis, as detailed in Section 1.2.3.

6.2.1 T1: Relation of Knowledge and Proximity

The utilization of proximity among visual symbols tends to establish a perceived relationship between these symbols for a human observer. This thesis demonstrates that it is possible to transform Euclidean distance and weighted relationships into one another using appropriate mathematical methods. The identified linear correlation between weight and distance could influence spatial parsing, facilitate the extraction of knowledge from spatial configurations, and serve as the cornerstone for the proposed spatial composition algorithms. This is due to the premise that a spatial arrangement of visual symbols can be interpreted in a manner that preserves an understanding of the knowledge graph, which serves as the basis for generating the spatial configuration.

6.2.2 T2: Improved Efficiency for Spatial Composition

The primary motivation for examining spatial composition was the anticipated benefits it offers in terms of 'reading' the outcomes, as it is seamlessly embedded within the spatial hypertext environment. A user study centered on an exploratory search task indicated that this method is more efficient compared to a conventional result visualization presented in a ranked list format. This finding is consistent with other recent research [68] but specifically focuses on a spatial hypertext interface with spatial composition in place. Although this initial finding is encouraging, it is evident that more extended longitudinal studies are required to attribute the observed effects to specific characteristics of the composition comprehensively.

6.2.3 T3: Reasoning about Efficiency

Assessing efficiency and other commonly used statistics alone might be inadequate if the goal is to understand the underlying causes of the observed values. Therefore, in addition to other methodologies, we segment the recorded user sessions into distinct steps. These steps are constructed based on the interactions users had with the interface. Subsequently, we correlate the observed effects with the state of each specific step. The findings demonstrated that study participants approached the given task similarly, regardless of the interface used. Nonetheless, the interface featuring spatial composition facilitated a more rapid processing of the provided information throughout the session. This enhancement leads to the observed improvement in task efficiency.

As noted above, this area requires additional studies for a more comprehensive understanding. The method introduced in this study could serve as a valuable tool for examining these and related research efforts. This is because it offers a structured approach to evaluating efficiency over time, rather than simply presenting a final static value. Comprehending the manner in which users perceive and process displayed information is crucial for the design and enhancement of spatial composition algorithms. For example, it is important to recognize that there exists an upper limit to the number of information units that users can effectively process before they move on to the subsequent interaction.

6.2.4 T4: Spatial Composition Algorithms

The concept of spatial composition, as detailed in this thesis, is broadly correlated with the domain of information visualization and encounters several challenges similar to those found in graph drawing, as discussed in Section 3.3.3. In addition to these commonalities, spatial composition addresses a distinct gap by primarily aiming to enhance structures generated predominantly by humans with supplementary information derived from a knowledge base. Building upon established visualization methodologies, we elaborated on a series of five algorithms and conducted a comparative analysis. This analysis is intended to serve as a fundamental basis for future research since it evaluates several critical aspects, such as responsiveness and space utilization, which are considered significant. The algorithms themselves provide a foundational framework for both current and forthcoming user studies focused on spatial composition.

6.2.5 T5: Case Studies

The proposed spatial hypertext interface, which incorporates recommendations through spatial composition, is suggested to be effective in various domains requiring exploratory search. Though its universal applicability cannot be confirmed, the interface has been utilized in multiple research projects and application areas. Feedback from domain experts and users has been crucial in refining the system, influencing the composition methodology, and enhancing the understanding of exploratory search and spatial hypertext.

6.3 Future Work

This thesis aims to set the ground for further investigations by establishing spatial composition as a research topic. Exemplary implementations, along with analysis and user studies to test their applicability, should help develop spatial composition in specific directions. The following ideas and questions are possible next steps – academically as well as for application in real-world domains.

Aliasing

As mentioned in Section 4.2.5, aliasing in spatial composition indicates the duplication of recommendations to minimize the impact restrictions implied by MDS. Currently, recommendations exist exactly once per context, and sometimes this necessitates the composition algorithm to make compromises when positioning a recommendation. With aliasing, a recommendation can exist at more than one position, making misinterpretation due to these compromises less likely. All proposed composition algorithms are working on the structure and are unaware of the recommendations themselves; therefore, an aliasing component could work independently in a preprocessing step. The arrangement of the user concepts and the weights of the recommendation graph need to be considered. Simply put, if a recommendation has a significant high weight to more than one user concept, and if they are not close to each other, the recommendation must be duplicated; such that each alias features a subset of the original edges the recommendation had. Besides the technical challenges, aliasing offers room for interesting further studies on this topic and may further advance spatial composition.

Structure Awareness

The proposed algorithms are only partially aware of the user-generated structure. They try to adapt to the restrictions imposed by the structure, such as using only available space or maintaining the meta context, but they do not recognize the type of structure. For example, consider a list of objects. Assume the list consists of four objects numbered one to four. In an ideal scenario, the spatial parser would recognize that the user is creating a list. Then it would derive a query with the payload and the structural information. The knowledge base processes this information and responds with a recommendation like “five”. Finally, the composition algorithm offers the possibility to continue the list structure with that recommendation.

To achieve this, two major additions are needed. First, much like *generative artificial intelligence* is used in large language models like GPT, understanding of the structure must be used to predict possible future changes to the context, such as adding “five” to the list. This prediction is not limited to additions, but may include positional changes to existing objects as well. Second, query derivation (cf. Section 4.3.2) must be advanced. Currently, the structure is converted to a weighted, undirected graph, which is then used to identify groups of objects that visually belong together. Instead, the query must contain the graph itself, along with attributes that describe the type of structure. This way, a knowledge base could provide structure-aware recommendations.

Long-Term Studies

One significant limitation of the current work is the short-term nature of the user studies conducted. Although these studies provided valuable initial insights into user interactions and the effectiveness of spatial composition, they do not capture the long-term user experience and potential changes in user behavior over time. Future work should include long-term user studies to evaluate the sustained impact and usability of spatial composition. These studies should monitor user engagement, learning curves, and adaptability over extended periods to understand how users integrate and benefit from spatial composition in their workflows. Longitudinal studies will provide deeper insight into how spatial composition influences user satisfaction, productivity, and overall interaction quality. In addition, further studies may incorporate real-world scenarios with a well-defined group of users, providing stronger evidence for the claims derived due to a more focused approach.

Iucundi sunt acti labores. — Cicero

Bibliography

[1] *2Dsearch. Web Application.* URL: `https://app.2dsearch.com/` (visited on 07/04/2023).

[2] Michele Angelaccio, Tiziana Catarci, and Giuseppe Santucci. "Query by Diagram: A Fully Visual Query System". In: *Journal of Visual Languages & Computing* 1.3 (Sept. 1990), pp. 255–273. ISSN: 1045-926X. DOI: `10.1016/S1045-926X(05)80009-6`.

[3] Peter G. Anick et al. "A Direct Manipulation Interface for Boolean Information Retrieval via Natural Language Query". In: *Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1990* (Dec. 1989), pp. 135–150. DOI: `10.1145/96749.98015`.

[4] Alessio Antonini et al. "Hypertext as Method: Reflections on Hypertext as Design Logic". In: *Proceedings of the 34th ACM Conference on Hypertext and Social Media.* HT '23. Rome, Italy: Association for Computing Machinery, 2023. DOI: `10.1145/3603163.3609074`.

[5] Claus Atzenbeck. "WildDocs: Investigating Construction of Metaphors in Office Work". PhD thesis. Denmark: Aalborg University, 2006. ISBN: 8776060144.

[6] Claus Atzenbeck, Sam Brooker, and Daniel Roßner. "Storytelling Machines". In: *Proceedings of the 6th Workshop on Human Factors in Hypertext.* New York, NY, USA: ACM, Sept. 2023, pp. 1–9. DOI: `10.1145/3603607.3613481`.

[7] Claus Atzenbeck, Eelco Herder, and Daniel Roßner. “Breaking the Routine: Spatial Hypertext Concepts for Active Decision Making in Recommender Systems”. In: *New Review of Hypermedia and Multimedia* 29.1 (Jan. 2023), pp. 1–35. ISSN: 1361-4568. DOI: 10.1080/13614568.2023.2170474.

[8] Claus Atzenbeck, Peter Nürnberg, and Daniel Roßner. “Synthesising Augmentation and Automation”. In: *New Review of Hypermedia and Multimedia* 27.1-2 (Apr. 2021), pp. 177–203. ISSN: 1361-4568. DOI: 10.1080/13614568.2021.1942237.

[9] Claus Atzenbeck and Peter J. Nürnberg. “Hypertext as Method”. In: *HT 2019 - Proceedings of the 30th ACM Conference on Hypertext and Social Media*. New York, NY, USA: ACM, Sept. 2019, pp. 29–38. ISBN: 9781450368858. DOI: 10.1145/3342220.3343669.

[10] Claus Atzenbeck and Peter J. Nürnberg. “Looking Beyond Computer Applications: Investigating Rich Structures”. In: *Lecture Notes in Computer Science* 3511 (2005), pp. 51–65. ISSN: 03029743. DOI: 10.1007/11518358_5.

[11] Claus Atzenbeck and Daniel Roßner. “Thoughts Reflection Machine”. In: *Proceedings of the 31st ACM Conference on Hypertext and Social Media, HT 2020*. New York, NY, USA: Association for Computing Machinery, Inc, July 2020, pp. 117–121. ISBN: 9781450370981. DOI: 10.1145/3372923.3404837.

[12] Claus Atzenbeck, Daniel Roßner, and Manolis Tzagarakis. “Mother - An Integrated Approach to Hypertext Domains”. In: *HT 2018 - Proceedings of the 29th ACM Conference on Hypertext and Social Media*. New York, New York, USA: ACM Press, 2018, pp. 145–149. ISBN: 9781450354271. DOI: 10.1145/3209542.3209570.

[13] Claus Atzenbeck et al. “Revisiting Hypertext Infrastructure”. In: *HT 2017 - Proceedings of the 28th ACM Conference on Hypertext and Social Media*. HT ’17. New York, NY, USA: ACM, 2017, pp. 35–44. ISBN: 9781450347082. DOI: 10.1145/3078714.3078718.

[14] Belinda Barnet, Niels Brügger, Ian Milligan, and Belinda Barnet. “Hypertext Before the Web – Or, What the Web Could Have Been”. In: *The SAGE Handbook of Web History*. Ed. by Niels Brügger and Ian Milligan. 1 Oliver’s Yard, 55 City Road London EC1Y 1SP: SAGE Publications Ltd, May 2019. Chap. 15, pp. 215–226. DOI: 10.4135/9781526470546.n15.

[15] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. “A Taxonomy and Survey of Dynamic Graph Visualization”. In: *Computer Graphics Forum* 36.1 (Jan. 2017), pp. 133–159. ISSN: 1467-8659. DOI: 10.1111/CGF.12791.

[16] Christoph Beckmann and Tom Gross. “Towards a Group Recommender Process Model for Ad-Hoc Groups and On-Demand Recommendations”. In: *Proceedings of the 16th ACM International Conference on Supporting Group Work - GROUP ’10*. New York, New York, USA: ACM Press, 2010, p. 329. ISBN: 9781450303873. DOI: 10.1145/1880071.1880134.

[17] Tim Berners-lee, James Hendler, and Ora Lassila. “The Semantic Web”. In: *Scientific American* 284.5 (2001), pp. 34–43. DOI: 10.2307/26059207.

[18] Tim Berners-Lee et al. “The World-Wide Web”. In: *Communications of the ACM* 37.8 (Aug. 1994), pp. 76–82. ISSN: 0001-0782. DOI: 10.1145/179606.179671.

[19] Mark Bernstein. “Collage, Composites, Construction”. In: *Proceedings of the ACM Conference on Hypertext*. Vol. 14. New

York, New York, USA: Association for Computing Machinery (ACM), 2003, pp. 122–123. DOI: 10.1145/900051.900077.

[20] Katy Börner, Andreas Bueckle, and Michael Ginda. "Data Visualization Literacy: Definitions, Conceptual Frameworks, Exercises, and Assessments". In: *Proceedings of the National Academy of Sciences* (2019). ISSN: 0027-8424. DOI: 10.1073/pnas.1807180116.

[21] Robert Botsch. *Significance and Measures of Association.* 2011. Chap. 12. URL: https://www.agreestat.com/book4/9780970806284_chap12.pdf (visited on 07/22/2024).

[22] Vannevar Bush. "As We May Think". In: *The Atlantic Monthly* 176.1 (1945), pp. 101–108. URL: http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm.

[23] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in Information Visualization : Using Vision to Think.* San Francisco: Morgan Kaufmann Publishers Inc., 1999, p. 686. ISBN: 978-1-55860-533-6. URL: https://dl.acm.org/doi/10.5555/300679.

[24] Erin Catto. "Iterative dynamics with temporal coherence". In: *Game Developer Conference* (Mar. 2005). URL: https://api.semanticscholar.org/CorpusID:11084512.

[25] CEN. *Maintenance Terminology.* European Standard EN 13306:2001. European Committee for Standardization, June 2001.

[26] George Merrill Chaikin. "An Algorithm for High-Speed Curve Generation". In: *Computer Graphics and Image Processing* 3.4 (Dec. 1974), pp. 346–349. ISSN: 0146664X. DOI: 10.1016/0146-664X(74)90028-8.

[27] Matthew Chalmers and Paul Chitson. "Bead: Explorations in Information Visualization". In: *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* SIGIR '92. New York, NY, USA: ACM, 1992, pp. 330–337. ISBN: 0-89791-523-2. DOI: 10.1145/133160.133215.

[28] Mohamed Amine Chatti, Mouadh Guesmi, and Arham Muslim. "Visualization for Recommendation Explainability: A Survey and New Perspectives". In: *ACM Transactions on Interactive Intelligent Systems* (June 2024). ISSN: 2160-6455. DOI: 10.1145/3672276.

[29] Chaomei Chen and Mary Czerwinski. "From Latent Semantics to Spatial Hypertext — An Integrated Approach". In: *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia.* New York, New York, USA: ACM Press, 1998, pp. 77–86. ISBN: 0897919726. DOI: 10.1145/276627.276636.

[30] Xiaorui Chen and Sara McMains. "Polygon Offsetting by Computing Winding Numbers". In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* Long Beach, California, USA: ASME, Sept. 2005, pp. 565–575. ISBN: 0-7918-4739-X. DOI: 10.1115/DETC2005-85513.

[31] Domenic V. Cicchetti. "Guidelines, Criteria, and Rules of Thumb for Evaluating Normed and Standardized Assessment Instruments in Psychology". In: *Psychological Assessment* 6.4 (1994), pp. 284–290. ISSN: 10403590. DOI: 10.1037/1040-3590.6.4.284.

[32] Jacob Cohen. "Statistical Power Analysis for the Behavioral Sciences". In: *Statistical Power Analysis for the Behavioral Sciences* (May 2013). DOI: 10.4324/9780203771587.

[33] Jeff Conklin. “Hypertext: An Introduction and Survey”. In: *Computer* 20.9 (Sept. 1987), pp. 17–41. ISSN: 0018-9162. DOI: 10.1109/MC.1987.1663693.

[34] Kenneth James Williams Craik. *The Nature of Explanation.* Cambridge University Press: Cambridge, UK, 1943.

[35] Andries van Dam. “Reflections on a Half-Century of Hypertext”. In: *Proceedings of the 30th ACM Conference on Hypertext and Social Media.* New York, NY, USA: ACM, Sept. 2019, pp. 3–4. ISBN: 9781450368858. DOI: 10.1145/3342220.3344782.

[36] Douglas R. Dechow and Daniele C. Struppa. *Intertwingled.* Ed. by Douglas R. Dechow and Daniele C. Struppa. History of Computing. Cham: Springer International Publishing, 2015. ISBN: 978-3-319-16924-8. DOI: 10.1007/978-3-319-16925-5.

[37] Oscar Díaz et al. *Rapid Mashup Development Tools.* Ed. by Florian Daniel and Cesare Pautasso. Vol. 591. Communications in Computer and Information Science. Cham: Springer International Publishing, 2016, pp. 82–97. ISBN: 978-3-319-28726-3. DOI: 10.1007/978-3-319-28727-0.

[38] Ernst Diehl and John D. Sterman. “Effects of Feedback Complexity on Dynamic Decision Making”. In: *Organizational Behavior and Human Decision Processes* 62.2 (May 1995), pp. 198–215. ISSN: 0749-5978. DOI: 10.1006/OBHD.1995.1043.

[39] P. Eades, M. Huang, and Junhu Wang. “Online Animated Graph Drawing using a Modified Spring Algorithm”. In: *Journal of Visual Languages and Computing* (1997).

[40] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. “On the Shape of a Set of Points in the Plane”. In: *IEEE Transactions on Information Theory* 29.4 (1983), pp. 551–559. ISSN: 1557-9654. DOI: 10.1109/TIT.1983.1056714.

[41] Deborah M Edwards and Lynda Hardman. "'Lost in Hyperspace': Cognitive Mapping and Navigation in a Hypertext Environment". In: *Hypertext: Theory into Practice*. 1999, pp. 90–105. ISBN: 1871516285. URL: `https://api.semanticscholar.org/CorpusID:261291345`.

[42] Lisa Eidloth, Daniel Roßner, and Claus Atzenbeck. "User Study on Link-Service Usage and Information Processing in the Context of the World Wide Web". In: *Proceedings of the 5th Workshop on Human Factors in Hypertext*. New York, NY, USA: ACM, June 2022, pp. 1–9. ISBN: 9781450394017. DOI: `10.1145/3538882.3542802`.

[43] D. C. Engelbart. *Augmenting Human Intellect: A Conceptual Framework*. Tech. rep. 1962. URL: `https://www.dougengelbart.org/pubs/augment-3906.html`.

[44] G. Fischer et al. "Meta-Design: A Manifesto for End-User Development". In: *Commun. ACM* 47.9 (Sept. 2004), pp. 33–37. ISSN: 0001-0782. DOI: `10.1145/1015864.1015884`.

[45] Ken Fishkin and Maureen C. Stone. "Enhanced Dynamic Queries via Movable Filters". In: *Conference on Human Factors in Computing Systems - Proceedings* 1 (1995), pp. 415–420. DOI: `10.1145/223904.223960`.

[46] Luis Francisco-Revilla and Frank Shipman. "Parsing and Interpreting Ambiguous Structures in Spatial Hypermedia". In: *HT 2005 - 16th ACM Conference on Hypertext and Hypermedia*. New York, New York, USA: ACM Press, 2005, pp. 107–116. ISBN: 1595931686. DOI: `10.1145/1083356.1083376`.

[47] C. Daniel Freeman et al. *Brax - A Differentiable Physics Engine for Large Scale Rigid Body Simulation*. Version 0.9.2. 2021. URL: `http://github.com/google/brax`.

[48] Vivian Genaro Motti, Dave Raggett, Sascha Van Cauwelaert, and Jean Vanderdonckt. "Simplifying the Development of Cross-Platform Web User Interfaces by Collaborative Model-Based Design". In: *Proceedings of the 31st ACM International Conference on Design of Communication.* SIGDOC '13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 55–64. ISBN: 9781450321310. DOI: 10.1145/2507065.2507067.

[49] Walter C. Gogel. "The Visual Perception of Size and Distance". In: *Vision Research* 3.3-4 (Mar. 1963), pp. 101–120. ISSN: 0042-6989. DOI: 10.1016/0042-6989(63)90033-6.

[50] Eric J. Golin. "Parsing Visual Languages with Picture Layout Grammars". In: *Journal of Visual Languages and Computing* 2.4 (Dec. 1991), pp. 371–393. ISSN: 1045926X. DOI: 10.1016/S1045-926X(05)80005-9.

[51] Günther Greiner and Kai Hormann. "Efficient Clipping of Arbitrary Polygons". In: *ACM Transactions on Graphics* 17.2 (Apr. 1998), pp. 71–83. ISSN: 0730-0301. DOI: 10.1145/274363.274364.

[52] Tom Gross. "Supporting Informed Negotiation Processes in Group Recommender Systems". In: *icom* 14.1 (Jan. 2015), pp. 53–61. ISSN: 2196-6826. DOI: 10.1515/icom-2015-0008.

[53] Florian Haag, Steffen Lohmann, Stephan Siek, and Thomas Ertl. "QueryVOWL: A Visual Query Notation for Linked Data". In: *Lecture Notes in Computer Science.* Vol. 9341. Springer Verlag, 2015, pp. 387–402. ISBN: 9783319256382. DOI: 10.1007/978-3-319-25639-9_51.

[54] Frank Halasz and Mayer Schwartz. "The Dexter Hypertext Reference Model". In: *Communications of the ACM* 37.2 (Feb. 1994), pp. 30–39. ISSN: 00010782. DOI: 10.1145/175235.175237.

[55] Frank G. Halasz, Thomas P. Moran, and Randall H. Trigg. "Notecards in a Nutshell". In: *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface* (1986), pp. 45–52. ISSN: 07135424. DOI: 10.1145/29933.30859.

[56] G. Halasz Frank. "Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems". In: *Communications of the ACM* 31.7 (July 1988), pp. 836–852. ISSN: 0001-0782. DOI: 10.1145/48511.48514.

[57] E. Herder, D. Roßner, and C. Atzenbeck. "Reflecting on Social Media Behavior by Structuring and Exploring Posts and Comments". In: *i-com* 19.3 (2021). ISSN: 21966826. DOI: 10.1515/icom-2020-0019.

[58] Eelco Herder, Daniel Roßner, and Claus Atzenbeck. "Hypertext as a Tool for Exploring Personal Data on Social Media". In: *Proceedings of the 31st ACM Conference on Hypertext and Social Media, HT 2020*. New York, NY, USA: Association for Computing Machinery, Inc, July 2020, pp. 135–136. ISBN: 9781450370981. DOI: 10.1145/3372923.3404831.

[59] Eelco Herder, Daniel Roßner, and Claus Atzenbeck. "Structuring and Exploring User Behavioral Patterns in Social Media Traces". In: *Mensch und Computer 2020 - Workshopband*. Bonn: Gesellschaft für Informatik e.V., 2020. DOI: 10.18420/muc2020-ws111-343.

[60] I. Herman, G. Melancon, and M.S. Marshall. "Graph Visualization and Navigation in Information Visualization: A Survey". In: *IEEE Transactions on Visualization and Computer Graphics* 6.1 (2000), pp. 24–43. ISSN: 10772626. DOI: 10.1109/2945.841119.

[61] Hao-Wei Hsieh and Frank M. Shipman. "VITE: A Visual Interface Supporting the Direct Manipulation of Structured Data

Using Two-Way Mappings". In: *IUI '00: Proceedings of the 5th International Conference on Intelligent User Interfaces*. Association for Computing Machinery (ACM), Jan. 2000, pp. 141–148. DOI: 10.1145/325737.325810.

[62] Muhammad Imran et al. "ResEval Mash: A Mashup Tool for Advanced Research Evaluation". In: *Proceedings of the 21st International Conference on World Wide Web*. WWW '12 Companion. New York, NY, USA: ACM, 2012, pp. 361–364. ISBN: 978-1-4503-1230-1. DOI: 10.1145/2187980.2188049.

[63] P.N. Johnson-Laird. "Mental Models in Cognitive Science". In: *Cognitive Science* 4.1 (1980), pp. 71–115. DOI: 10.1016/S0364-0213(81)80005-5.

[64] Contributors of the JoltPhysics Project. *JoltPhysics - A multi core friendly rigid body physics and collision detection library, written in C++, suitable for games and VR applications*. Version 3.0.1. 2022. URL: https://github.com/jrouwe/JoltPhysics/.

[65] Martijn Kagie, Michiel van Wezel, and Patrick J.F. Groenen. "Map Based Visualization of Product Catalogs". In: *Recommender Systems Handbook* (2011), pp. 547–576. DOI: 10.1007/978-0-387-85820-3_17.

[66] Andruid Kerne, Eunyee Koh, Vikram Sundaram, and J. Michael Mistrot. "Generative Semantic Clustering in Spatial Hypertext". In: *Proceedings of the 2005 ACM Symposium on Document Engineering*. New York, New York, USA: ACM Press, 2005, pp. 84–93. ISBN: 1595932402. DOI: 10.1145/1096601.1096624.

[67] Khalil Klouche et al. "Designing for Exploratory Search on Touch Devices". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. New

York, NY, USA: ACM, 2015, pp. 4189–4198. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702489.

[68] Khalil Klouche et al. "Visual Re-Ranking for Multi-Aspect Information Retrieval". In: *Proceedings of the 2017 Conference Human Information Interaction and Retrieval*. CHIIR '17. New York, NY, USA: ACM, 2017, pp. 57–66. ISBN: 9781450346771. DOI: 10.1145/3020165.3020174.

[69] Stephen G. Kobourov. "Spring Embedders and Force Directed Graph Drawing Algorithms". In: *arXiv* (Jan. 2012). URL: http://arxiv.org/abs/1201.3011.

[70] Matija König, Jaka Dirnbek, and Vlado Stankovski. "Architecture of an Open Knowledge Base for Sustainable Buildings Based on Linked Data Technologies". In: *Automation in Construction* 35 (Nov. 2013), pp. 542–550. ISSN: 0926-5805. DOI: 10.1016/J.AUTCON.2013.07.002.

[71] Terry K. Koo and Mae Y. Li. "A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research". In: *Journal of Chiropractic Medicine* 15.2 (June 2016), pp. 155–163. ISSN: 1556-3707. DOI: 10.1016/J.JCM.2016.02.012.

[72] Uwe Krohn. "VINETA: Navigation through Virtual Information Spaces". In: *Proceedings of the workshop on Advanced visual interfaces - AVI '96*. New York, New York, USA: ACM Press, 1996, p. 49. ISBN: 0897918347. DOI: 10.1145/948449.948458.

[73] J. Richard Landis and Gary G. Koch. "The Measurement of Observer Agreement for Categorical Data". In: *Biometrics* 33.1 (1977), pp. 159–174. ISSN: 0006341X, 15410420. URL: http://www.jstor.org/stable/2529310 (visited on 08/14/2023).

[74] Henry Lieberman, Fabio Paternò, Markus Klann, and Volker Wulf. "End-User Development: An Emerging Paradigm". In: *End User Development*. Ed. by Henry Lieberman, Fabio Paternò, and Volker Wulf. Dordrecht: Springer Netherlands, 2006, pp. 1–8. ISBN: 978-1-4020-5386-3. DOI: 10.1007/1-4020-5386-X_1.

[75] Kirstin Lyon and Peter J Nürnberg. "Applying Information Visualisation Techniques to Spatial Hypertext Tools". In: *Metainformatics*. Ed. by Uffe Kock Wiil. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 85–93. ISBN: 978-3-540-32105-7.

[76] Gary Marchionini. "Exploratory Search". In: *Communications of the ACM* 49.4 (Apr. 2006), pp. 41–46. ISSN: 00010782. DOI: 10.1145/1121949.1121979.

[77] Gary Marchionini and Ben Shneiderman. "Finding Facts vs. Browsing Knowledge in Hypertext Systems". In: *Computer* 21.01 (Jan. 1988), pp. 70–80. ISSN: 0018-9162. DOI: 10.1109/2.222119.

[78] C.C. Marshall and F.M. Shipman III. "Searching for the Missing Link: Discovering Implicit Structure in Spatial Hypertext". In: *Proceedings of the fifth ACM conference on Hypertext* November (1993), pp. 217–230. URL: http://portal.acm.org/citation.cfm?id=168826.

[79] Catherine C Marshall, Frank G Halasz, Russell A Rogers, and William C Janssen Jr. "Aquanet: A Hypertext Tool to Hold Your Knowledge in Place". In: *Proceedings of the Third Annual ACM Conference on Hypertext*. HYPERTEXT '91. New York, NY, USA: ACM, 1991, pp. 261–275. ISBN: 0-89791-461-9. DOI: 10.1145/122974.123000.

[80] Catherine C Marshall, Frank M Shipman III, and James H Coombs. "VIKI: Spatial Hypertext Supporting Emergent Structure". In: *Proceedings of the 1994 ACM European Conference on*

Hypermedia Technology. ECHT '94. New York, NY, USA: ACM, 1994, pp. 13–23. ISBN: 0-89791-640-9. DOI: 10.1145/192757.192759.

[81] Catherine C. Marshall and Frank M. Shipman. "Which Semantic Web?" In: *Proceedings of the ACM Conference on Hypertext* 14 (2003), pp. 57–66. DOI: 10.1145/900051.900063.

[82] Francisco Martínez, Carlos Ogayar, Juan R. Jiménez, and Antonio J. Rueda. "A Simple Algorithm for Boolean Operations on Polygons". In: *Advances in Engineering Software* 64 (Oct. 2013), pp. 11–19. ISSN: 09659978. DOI: 10.1016/j.advengsoft.2013.04.004.

[83] Contributors of the Matter.js project. *Matter.js - A JavaScript 2D rigid body physics engine for the web*. Version 0.19.0. 2014. URL: https://github.com/liabru/matter-js.

[84] Don McCready. "On Size, Distance, and Visual Angle Perception". In: *Perception & Psychophysics* 37.4 (1985), pp. 323–334. ISSN: 1532-5962. DOI: 10.3758/BF03211355.

[85] Alexander Milevski and Vladimir Ovsyannikov. *Polygon Offset*. Version 0.3.2. 2022. URL: https://github.com/w8r/polygon-offset.

[86] Dave E. Millard, Luc Moreau, Hugh C. Davis, and Siegfried Reich. "FOHM". In: *Proceedings of the eleventh ACM on Hypertext and hypermedia*. New York, NY, USA: ACM, May 2000, pp. 93–102. ISBN: 1581132271. DOI: 10.1145/336296.336334.

[87] Meredith Ringel Morris and Eric Horvitz. "SearchTogether: An Interface for Collaborative Web Search". In: *UIST: Proceedings of the Annual ACM Symposium on User Interface Softaware and Technology* (2007), pp. 3–12. DOI: 10.1145/1294211.1294215.

[88] Oleg R Musin. "Properties of the Delaunay Triangulation". In: *Proceedings of the 13th Annual Symposium on Computational Geometry*. 1997, pp. 424–426.

[89] J. A. Nelder and R. Mead. "A Simplex Method for Function Minimization". In: *The Computer Journal* 7.4 (Jan. 1965), pp. 308–313. ISSN: 0010-4620. DOI: 10.1093/COMJNL/7.4.308.

[90] T. H. Nelson. "Complex Information Processing: A File Structure for the Complex, the Changing and the Indeterminate". In: *Proceedings of the 1965 20th National Conference*. ACM '65. Cleveland, Ohio, USA: Association for Computing Machinery, 1965, pp. 84–100. ISBN: 9781450374958. DOI: 10.1145/800197.806036.

[91] Theodore H. Nelson. *Computer Lib/Dream Machines*. 1974.

[92] A Nuchprayoon and R R Korfhage. "GUIDO, a Visual Tool for Retrieving Documents". In: *Proceedings of 1994 IEEE Symposium on Visual Languages*. 1994, pp. 64–71. DOI: 10.1109/VL.1994.363639.

[93] Kai Olsen, James G Williams, Kenneth M Sochats, and Stephen Hirtle. "Ideation through visualization: The VIBE System". In: *Multimedia Review* 3 (1991).

[94] Fabio Paternò. "End User Development: Survey of an Emerging Field for Empowering People". In: *ISRN Software Engineering* 2013 (2013). ISSN: 2090-7680. DOI: 10.1155/2013/532659.

[95] L Paul Chew. "Constrained Delaunay Triangulations". In: *Algorithmica* 4.1 (1989), pp. 97–108. ISSN: 1432-0541. DOI: 10.1007/BF01553881.

[96] Guillaume Péron. "Weighting the Transitivity of Undirected Weighted Social Networks with Triadic Edge Dissimilarity Scores". In: *Social Networks* 73 (2023), pp. 1–6. ISSN: 0378-8733. DOI: https://doi.org/10.1016/j.socnet.2022.12.001.

[97] Peter Fleischmann. *Conforming Delaunay Triangulation*. Jan. 2000. URL: https://www.iue.tuwien.ac.at/phd/fleischmann/node53.html.

[98] Susanne Purucker, Claus Atzenbeck, and Daniel Roßner. "Intelligent Hypertext for Video Selection: A Design Approach". In: *HUMAN 2019 - Proceedings of the 2nd International Workshop on Human Factors in Hypertext*. New York, New York, USA: ACM Press, 2019, pp. 19–26. ISBN: 9781450368995. DOI: 10.1145/3345509.3349279.

[99] Olav Reinert, Dirk Bucka-Lassen, Claus Aagaard Pedersen, and Peter J Nürnberg. "CAOS: A Collaborative and Open Spatial Structure Service Component with Incremental Spatial Parsing". In: *Proceedings of the Tenth ACM Conference on Hypertext and Hypermedia*. HYPERTEXT '99. New York, NY, USA: ACM, 1999, pp. 49–50. ISBN: 1-58113-064-3. DOI: 10.1145/294469.294484.

[100] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. Boston, MA: Springer US, 2011. ISBN: 978-0-387-85819-7. DOI: 10.1007/978-0-387-85820-3.

[101] Christian Richthammer, Johannes Sänger, and Günther Pernul. "Interactive Visualization of Recommender Systems Data". In: *ACM International Conference Proceeding Series* Part F131370 (June 2017), pp. 19–24. DOI: 10.1145/3099012.3099014.

[102] Daniel Roßner and Claus Atzenbeck. "Demonstration of Weblinks: A Rich Linking Layer Over the Web". In: *Proceedings of the 32st ACM Conference on Hypertext and Social Media*. New York, NY, USA: ACM, Aug. 2021, pp. 283–286. ISBN: 9781450385510. DOI: 10.1145/3465336.3475123.

[103] Daniel Roßner and Claus Atzenbeck. “Spatial Hypertext for End-User Development Tools”. In: *HUMAN 2018 - Proceedings of the 1st Workshop on Human Factors in Hypertext, Co-located with HT 2018*. New York, New York, USA: ACM Press, 2018, pp. 9–15. ISBN: 9781450356589. DOI: 10.1145/3215611.3215612.

[104] Daniel Roßner, Claus Atzenbeck, and Sam Brooker. “SPORE”. In: *Proceedings of the 34th ACM Conference on Hypertext and Social Media*. New York, NY, USA: ACM, Sept. 2023, pp. 1–6. DOI: 10.1145/3603163.3609075.

[105] Daniel Roßner, Claus Atzenbeck, and Tom Gross. “Spatial Layout Versus List Layout: A Comparative Study”. In: *Human-Computer-Interaction – INTERACT 2021*. Springer, Cham, Aug. 2021, pp. 495–498. DOI: 10.1007/978-3-030-85607-6_65.

[106] Daniel Roßner, Claus Atzenbeck, and Tom Gross. “The Effects of Spatial Visualization versus Ranked Lists on Quality, Time Efficiency, and Interaction”. In: *HT 2022: 33rd ACM Conference on Hypertext and Social Media - Co-located with ACM WebSci 2022 and ACM UMAP 2022* 11.22 (June 2022), pp. 132–142. DOI: 10.1145/3511095.3531286.

[107] Daniel Roßner, Claus Atzenbeck, and Tom Gross. “Visualization of the Relevance: Using Physics Simulations for Encoding Context”. In: *HT 2019 - Proceedings of the 30th ACM Conference on Hypertext and Social Media*. New York, New York, USA: ACM Press, 2019, pp. 67–76. ISBN: 9781450368858. DOI: 10.1145/3342220.3343659.

[108] Daniel Roßner, Claus Atzenbeck, and Daniel Urban. “Weblinks: Augmenting Web Browsers with Enhanced Link Services”. In: *Proceedings of the 3rd Workshop on Human Factors in Hypertext, HUMAN 2020*. New York, NY, USA: Association for Com-

puting Machinery, Inc, Dec. 2020, pp. 1–5. ISBN: 9781450380584. DOI: 10.1145/3406853.3432663.

[109] Daniel Roßner, Jae Sook Cheong, and Claus Atzenbeck. "From Maintenance in Industry to Bibliographic Data: Spatial Hypertext as Communication Medium Between User and Machine". In: *Proceedings of the 5th Workshop on Human Factors in Hypertext, HUMAN 2022* (June 2022). DOI: 10.1145/3538882.3542803.

[110] Daniel Roßner, Lisa Eidloth, and Claus Atzenbeck. "PAIRWISE – From Spatial Structure to Knowledge". In: *Proceedings of the 35th ACM Conference on Hypertext and Social Media*. HT '24. Poznan, Poland: Association for Computing Machinery, 2024, pp. 208–216. ISBN: 9798400705953. DOI: 10.1145/3648188.3675137.

[111] Tony Russell-Rose, Jon Chamberlain, and Farhad Shokraneh. "A Visual Approach to Query Formulation for Systematic Search". In: *CHIIR 2019 - Proceedings of the 2019 Conference on Human Information Interaction and Retrieval* (Mar. 2019), pp. 379–383. DOI: 10.1145/3295750.3298919.

[112] Nakissa Salvati. *Und sie drehen sich doch*. München, June 2023.

[113] Manojit Sarkar and Marc H. Brown. "Graphical Fisheye Views of Graphs". In: *Conference on Human Factors in Computing Systems - Proceedings* (1992), pp. 83–92. DOI: 10.1145/142750.142763.

[114] Thomas Schedel. "Spatio-Temporal Parsing in Spatial Hypermedia". PhD thesis. 2016. DOI: 10.57944/1051-71.

[115] Thomas Schedel. "Spatio-Temporal Parsing in Spatial Hypermedia". In: *ACM SIGWEB Newsletter*. HT '16 Winter (Feb. 2017), pp. 1–5. ISSN: 1931-1745. DOI: 10.1145/3027141.3027142.

[116] Thomas Schedel and Claus Atzenbeck. "Spatio-Temporal Parsing in Spatial Hypermedia". In: *HT '16: Proceedings of the 27th ACM Conference on Hypertext and Social Media*. HT '16. New York, NY, USA: ACM, 2016, pp. 149–157. ISBN: 9781450342476. DOI: 10.1145/2914586.2914596.

[117] C E Shannon. "Communication In The Presence Of Noise". In: *Proceedings of the IEEE* 86.2 (1998), pp. 447–457. ISSN: 1558-2256. DOI: 10.1109/JPROC.1998.659497.

[118] F. Shipman et al. "Visual and Spatial Communication and Task Organization Using the Visual Knowledge Builder". In: *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work* (2001), pp. 260–269. DOI: 10.1145/500286.500325.

[119] Frank Shipman et al. "Semantics Happen: Knowledge Building in Spatial Hypertext". In: *Proceedings of the ACM Conference on Hypertext*. New York, New York, USA: ACM Press, 2002, pp. 25–34. ISBN: 1581134770. DOI: 10.1145/513349.513350.

[120] Frank M Shipman III, Catherine C Marshall, and Thomas P Moran. "Finding and Using Implicit Structure in Human-organized Spatial Layouts of Information". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '95. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1995, pp. 346–353. ISBN: 0-201-84705-1. DOI: 10.1145/223904.223949.

[121] B. Shneiderman. "The Eyes Have it: A Task by Data Type Taxonomy for Information Visualizations". In: *Proceedings 1996 IEEE Symposium on Visual Languages*. IEEE Comput. Soc. Press, 1996, pp. 336–343. ISBN: 0-8186-7508-X. DOI: 10.1109/VL.1996.545307.

[122] Timothy W. Simpson et al. "Impact of Response Delay and Training on User Performance with Text-Based and Graphical User Interfaces for Engineering Design". In: *Research in Engineering Design* 18.2 (Aug. 2007), pp. 49–65. ISSN: 09349839. DOI: 10.1007/S00163-007-0033-Y/TABLES/3.

[123] Anselm Spoerri. "Infocrystal: A Visual Tool for Information Retrieval". In: *Proceedings of the 4th Conference on Visualization '93*. VIS '93. Washington, DC, USA: IEEE Computer Society, 1993, pp. 150–157. ISBN: 0-8186-3940-7. URL: http://dl.acm.org/citation.cfm?id=949845.949876.

[124] T. C. Sprenger, M. H. Gross, D. Bielser, and T. Strasser. "IVORY - An Object-Oriented Framework for Physics-Based Information Visualization in Java". In: *Proceedings of the IEEE Symposium on Information Visualization*. IEEE Comput. Soc, 1998, pp. 79–86. ISBN: 0-8186-9093-3. DOI: 10.1109/infvis.1998.729562.

[125] David L. Streiner, Geoffrey R. Norman, and John Cairney. *Health Measurement Scales: A Practical Guide to their Development and Use*. Oxford University Press, Nov. 2014. ISBN: 9780199685219. DOI: 10.1093/med/9780199685219.001.0001.

[126] Tanja Svarre and Tony Russell-Rose. "Think Outside the Search Box: A Comparative Study of Visual and Form-Based Query Builders". In: *Journal of Information Science* (Dec. 2022), p. 016555152211385. ISSN: 0165-5515. DOI: 10.1177/01655515221138536.

[127] Yuzuru Tanaka and Kimihito Ito. "Meme Media Architecture for the Reediting and Redistribution of Web Resources". In: 2004, pp. 1–12. DOI: 10.1007/978-3-540-25957-2_1.

[128] Bala R. Vatti. "A Generic Solution to Polygon Clipping". In: *Communications of the ACM* 35.7 (July 1992), pp. 56–63. ISSN: 0001-0782. DOI: 10.1145/129902.129906.

[129] John Vince. *Mathematics for Computer Graphics*. 5th. Springer Publishing Company, Incorporated, 2017. ISBN: 1447173341.

[130] Yang Wang et al. ""I regretted the minute I pressed share"". In: *Proceedings of the Seventh Symposium on Usable Privacy and Security*. New York, NY, USA: ACM, July 2011, pp. 1–16. ISBN: 9781450309110. DOI: 10.1145/2078827.2078841.

[131] Ryen W White and Resa A Roth. "Exploratory Search: Beyond the Query-Response Paradigm". In: *Synthesis Lectures on Information Concepts, Retrieval, and Services* 1.1 (2009), pp. 1–98. DOI: 10.2200/S00174ED1V01Y200901ICR003.

[132] Johannes Wirth, Daniel Roßner, René Peinl, and Claus Atzenbeck. "SPORENLP: A Spatial Recommender System for Scientific Literature". In: *Proceedings of the 19th International Conference on Web Information Systems and Technologies*. SCITEPRESS - Science and Technology Publications, 2023, pp. 429–436. ISBN: 978-989-758-672-9. DOI: 10.5220/0012210400003584.

[133] Ji Soo Yi, Youn Ah Kang, John T. Stasko, and Julie A. Jacko. "Toward a Deeper Understanding of the Role of Interaction in Information Visualization". In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (Nov. 2007), pp. 1224–1231. ISSN: 10772626. DOI: 10.1109/TVCG.2007.70515.

[134] Young Degi and Shneiderman Ben. "A Graphical Filter/Flow Representation of Boolean Queries: A Prototype Implementation and Evaluation". In: *Journal of the American Society for Information Science* (July 1993). DOI: 10.5555/160921.160924.

[135] Jin Zhang. "TOFIR: A Tool of Facilitating Information Retrieval - Introduce a Visual Retrieval Model". In: *Information Processing and Management* 37.4 (2001), pp. 639–657. ISSN: 03064573. DOI: 10.1016/S0306-4573(00)00042-X.

[136] Jin Zhang and Robert R Korfhage. "DARE: Distance and Angle Retrieval Environment: A Tale of the Two Measures". In: *Journal of the American Society for Information Science* 50.9 (1999), pp. 779–787. DOI: 10.1002/(SICI)1097-4571(1999)50:9<779::AID-ASI6>3.0.CO;2-R.

[137] Da-Qian Zhang and Kang Zhang. "VisPro: A Visual Language Generation Toolset". In: *Proceedings. 1998 IEEE Symposium on Visual Languages (Cat. No.98TB100254)*. Sept. 1998, pp. 195–202. DOI: 10.1109/VL.1998.706163.

[138] Yanxia Zhang et al. "Look Together: Using Gaze for Assisting Co-Located Collaborative Search". In: *Personal and Ubiquitous Computing* 21.1 (Feb. 2017), pp. 173–186. ISSN: 1617-4909. DOI: 10.1007/s00779-016-0969-x.