Supporting asynchronous, discontinuous, collaborative, complex search tasks by the visualization of search trails

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften

der RWTH Aachen University zur Erlangung des akademischen Grades

eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Informatiker Diplom-Wirtschaftsinformatiker

Sebastian Andreas Franken

Aus Geilenkirchen, Deutschland

Berichter: Prof. Wolfgang Prinz, Ph. D.
Prof. Dr. rer. pol. Matthias Jarke
Prof. Dr.-Ing. Thomas Herrmann

Tag der mündlichen Prüfung: 22. Juni 2016

Zusammenfassung

Diese Dissertation befasst sich mit dem Problem der mangelnden Unterstützung von Suchenden bei asynchronen, diskontinuierlichen und komplexen Internet-Suchen in individuellen und kollaborativen Suchszenarien. Beispiele für komplexe Suchaufgaben sind die Planung einer Urlaubsreise oder wissenschaftliche Recherche, bei denen sich das Gesamtergebnis aus individuellen Teilergebnissen zusammensetzt. Meine Analyse verwandter Arbeiten zeigt, dass aktuelle verwandte Systeme besonders bei asynchronen oder diskontinuierlichen Internet-Suchen nur begrenzt hilfreich sind. Um die Forschungsfrage „Können Suchpfade Unterstützung für komplexe Internet-Suchen bieten und wie sollte Werkzeugunterstützung für diesen Ansatz aussehen?“ zu beantworten, entwickle ich eine neuartige Lösung zur Unterstützung von asynchronen, diskontinuierlichen, kollaborativen und komplexen Internet-Suchen. Diese Lösung beinhaltet die Entwicklung einer Webbrowser Erweiterung, welche den Verlauf einer Internet-Suche visualisiert. Dies geschieht als Suchpfad, der die Möglichkeit bietet, wertvolle Information während der Suche zu speichern. Ein webbasierter Mechanismus ermöglicht die Speicherung der Suchpfade auf einem externen Server und den Austausch der Suchpfade zwischen den Benutzern.

Die Idee der Suchpfade geht zurück auf Vannevar Bush, der sie als Erster im Jahr 1945 im Rahmen eines Gedankenexperiments beschrieb. Die Idee hatte zwar Einfluss auf die Bildung von Modellen für Suchprozesse, fand allerdings nur wenig praktische Anwendung. Erst seit wenigen Jahren gewinnt die serverseitige Analyse von Suchpfaden praktische Relevanz für die Generierung von Empfehlungen. Mein Ansatz geht über diese Konzepte und Lösungsansätze hinaus, indem den Suchpfaden individueller Wert zugemessen wird. Dazu unterstützt mein System SearchTrails die Benutzer, indem es die Möglichkeit bietet, wichtige Suchresultate in ihrem jeweiligen Kontext zu speichern und so entstandene Suchpfade später fortsetzen zu können. Suchpfade als Kollaborationsartefakte ermöglichen die direkte Kooperation von Benutzern anhand von besuchten und evaluierten Ressourcen und ermöglichen einen ungefilterten Einblick wo bereits gesucht wurde, und welche Ergebnisse wo gefunden wurden.

Ich zeige die Effektivität und die Effizienz meines Ansatzes in zwei Benutzerstudien. Die erste Benutzerstudie zeigt qualitativ die Effektivität des gewählten Ansatzes, während die zweite Benutzerstudie quantitativ dessen Effizienz belegt. Die zweite Benutzerstudie konzentriert sich auf kollaborative Suchszenarien und zeigt den positiven Einfluss des entwickelten Systems auf die Qualität des Suchprozesses und die Suchresultate. Die Ergebnisse der beiden Studien ermöglichen die positive Beantwortung einer Reihe von Teilforschungsfragen, die die positive Beantwortung der Hauptforschungsfrage ermöglichen.

In den sechs Kapiteln dieser Dissertation beschreibe ich die Entwicklung, die Implementierung und die Evaluation meines Ansatzes. Die Ergebnisse dieser Dissertation wurden in vier Publikationen auf nationalen und internationalen Konferenzen veröffentlicht.

Summary

In this thesis, I address the problem of lacking support for asynchronous, discontinuous, and complex web search tasks in individual and collaborative search scenarios. Examples for complex search tasks are the planning of a holiday trip or scientific research: These search tasks have in common that the overall result is composed of individual partial results, which strongly depend on the searcher's personal preferences.

Especially when these web search tasks happen in an asynchronous or discontinuous way, my related work shows that current solutions provide only limited help. I answer the research question 'Can search trails provide support for complex web search and how should tool support look like?'. To achieve this, I develop a novel solution for supporting asynchronous, discontinuous, collaborative, complex web search tasks. I achieve this by implementing the web browser extension 'SearchTrails' which visualizes the user's web search behavior. These web search trails resemble the course of the user's web search activities and allow storing important search results. A web-based mechanism allows the online storage of the search trails data objects and the transfer of search trails between users. This way, saved search trails can be recreated by the users themselves or exchanged between collaborating searchers.

The idea of web search trails dates back to Vannevar Bush, who presented it in a thought experiment in 1945, and was taken up by only a few authors in the meantime. While the idea had some influence on theoretical models, it was not implemented for a long time. More practical research towards the search trails users leave during web search tasks on server-side dates from 2010, where search trails were evaluated on server-side for generating recommendations. My approach goes beyond existing approaches as it values the user's individual search trail above generalized recommendations. SearchTrails supports the users in several ways. SearchTrails helps users capturing important search results within their search context and enables catching up with previous search results. Search trails as collaboration artifacts enable direct collaboration between users and provide an unfiltered insight where the collaborating searcher has searched before, and where results have been found.

I show the effectivity and efficiency of my developed approach in two user studies. The first user study qualitatively shows the effectiveness of the developed approach, while the second user study quantitatively shows its efficiency. The second user study especially focuses on collaborative search scenarios and shows the impact of the developed system on the quality of the search process and the search results. Based on the findings from the user studies, a set of research questions can be answered positively to validate the main research question, whether it is possible to provide support for complex web search by building search trails.

In the thesis, I describe in six chapters the development, the implementation, and the evaluation of my approach. The results of my thesis have been published in four full papers at national and international conferences.

Acknowledgements

Since I started working at Fraunhofer FIT, this thesis has been a steady companion of mine. From the very early phases of developing an idea and verifying it with respect to current and past research, to the very last days of refining my written work, this project was always at my side. This may hold for the good days, when things worked out, technical challenges were mastered, or scientific problems were solved; but also for the challenging days, when things were stuck, motivation was limited, or technical issues blocked other progress. However, I was never alone doing this work, and there are many people I have to be grateful for. Their direct or indirect support, their motivation and feedback have helped pushing my thesis forward.

First of all, I thank for valuable academic advice. Most important is Professor Wolfgang Prinz, PhD, for offering trust, feedback, motivation, and opening spaces for the work on the thesis. I want to thank my reviewers Professor Dr. Matthias Jarke and Professor Dr. Thomas Herrmann for valuable advice and coaching through the final phases of this thesis. Special thanks go to Professor Dr. Ulrich Norbisrath, for providing motivation and advice, and his support by constant reflection of the work done. Our regular, long-distance telephone conferences were a source of inspiration and motivation for me and kept my mind on following the trails of the thesis. Professor Dr. Dirk Lewandowski offered help and support during the planning of the user studies. Of course, I thank all study participants for investing time and efforts in the evaluation of my systems. I thank my colleagues for their academic and social counsel and for always having an open ear for issues or success. These are – in alphabetical order – Constantin Brosda, Wolfgang Gräther, Nils Jeners, Sabine Kolvenbach, Katja Niemann, and Rudolf Ruland.

Furthermore, I thank my parents Friedrich and Liselotte, my brother Martin, my mother-in-law Helga, and all my friends for their support while realizing my thesis. Both their reminding of the achieved progress and their patience helped me carrying out this work. Most important was the support of my wife Nicola, who got to know me and married me while working on the thesis. You motivated and protected me, and generously left room for my work on this thesis. This one's for you.

Sebastian Andreas Franken

June 2016

Table of contents

Chapter 1
Motivation and introduction

'Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information upon it.'

Samuel Johnson, 1709-1784 (Boswell & Croker, 1831)

This citation of the English writer Samuel Johnson is most cited in its derived version 'The next best thing to knowing something, is knowing where to find it'[1]. Despite its age of around 250 years, the citation of Samuel Johnson did not lose a bit of its relevance when thinking about it from todays' perspective. It rather gains more and more importance through the passing of time, as the flood of information produced is no more meant to be learned by heart, but rather meant to be found on demand. The situation gets even worse when taking a closer look on the second part of Johnson's citation: 'Knowing where to find it' is not a simple task on itself. Facing the vast amount of analog and digital resources of information, such as books, magazines, newspapers, portals, databases, and the enormous possibilities of accessing that information via search portals one can easily see that the access to information gets both easier and more complicated at the same time. This leads to a high fraction of cases where users are unable to find all desired information at once, but have to assemble valuable pieces of information from different sources. Search engines in these cases work as distributors to more specific portals, in which the users have to start over again in their quest for specific pieces of information.

In my thesis, I describe and provide theoretical foundations and tool support for using search trails as artifacts for facilitating asynchronous, discontinuous, collaborative, complex search tasks. Such search tasks can for example be the organization of a holiday trip, which includes deciding for a destination, booking hotels, evaluating transportation options, or the organization of sightseeing tours. Search tasks like these require support for recalling the way of how a user came to a specific piece of information, and also support for assembling valuable information from a multitude of different sources. I provide and evaluate a way to support both single users dealing with long-running and unspecific search tasks as well as multiple users collaborating asynchronously during such tasks. The thesis shows that it is possible to improve both the single user and the collaborative search experience by capturing and storing the trail of navigation through the Web.

Many people nowadays are facing difficulties or frustration when trying to recall information from a large pool of information on a certain topic. This work therefore deals with keeping track of information seen during a research task and selecting the best parts of it. It enables users to handle the overflow of information that was frequently described throughout history.

Samuel Johnson was not the first, nor the last author complaining about the amount of information. Moreover, he stands in a long tradition of intellectuals doing so. Many centuries before Johnson complained, other authors did so. First hints on an overwhelming amount of in-

[1] Retrieved November 12, 2015 from http://www.samueljohnson.com/apocryph.html#12

formation can be found in the Bible, where it says: ‘And further, my son, take note of this: of the making of books there is no end […]’ (Bible, Ecclesiastes 12:12, n.d.). Even the Greek philosopher Seneca writes in a critical way on the amount of information in his letters on ethics to Lucillus: ‘distringit librorum multitudo’ or ‘the amount of books is distracting’. At least since the 13th century, people tried to cope with the amount of information by creating tables of content, indices, compendiums, (alphabetical) orderings, or intelligent layouts. Since the invention of letterpress printing and the resulting strong increase in production of books, there were more complaints on the amount of information, and resulting from this, books were published which collected central parts of other books. During the 18th century, a large number of encyclopedias and dictionaries were published, trying to organize the knowledge of the known world. These works were themselves summaries of information, reacting on the information overflow of those times.

This problem was named the ‘information overload problem’ by Bertram Gross in 1964 (Gross, 1964), and was made known to the wide public by Alvin Toffler by his book ‘Future Shock’ in 1970 (Toffler, 1970). The introductory chapters show that the name of that problem stands in a century old tradition and just forms a somehow normal reaction on the increasing amount of information of the author’s zeitgeist. Traditionally, all scientific disciplines have to cope with that overload. For example, (Eppler & Mengis, 2004) show an overview of publications on information overload in business literature from the last 30 years.

It is interesting to note that the aforementioned descriptions of information overload by Gross and Toffler were created at a time when the Internet did not even exist. Since its invention, the Internet has made a rapid development, with still increasing growth rates, according to Moore’s law (Moore, 1965). The amount of pages on the Internet is not precisely measurable. Different popular search engines are estimated to have indexed some 45 to 55 billion single pages[2]. These measurements do not even count the pages of the Deep Web, meaning the pages that are unreachable to search engines, because they are only visible after logging in or that are generated on demand. As this number is even harder to count, no serious estimations can be made for it (Lewandowski, 2015, p.273 ff.). The reader may just think about pages generated from photo databases or for calendar entries. Such web services can create almost infinite numbers of single pages, each with a certain amount of information present and valuable to some user. Thus, the amount of information on the Internet is not measurable, or only by very abstract means. Many pages and platforms try to attract the user’s interest, and to almost every page or platform one can find a page or platform offering identical or very similar information to the user. The question if certain information exists on the Internet more and more shifts to a quest in finding the resources where this information is available. This results in the problem of investigating where the best version of a piece of information is available or where the desired information is available in the highest quality. Frequently, identical information can be found in many places on the Internet, for example because it is simply copied from Wikipedia, but only few resources present information going beyond that. This links back to the second part of Johnson’s citation. Even if a piece of information is unique at one place of the Internet, it is most certainly not that complete that it covers an information need in its entirety. Other facets, related pieces of information, or additional fact could be found on the Internet and help enriching the unique information to become even more meaningful for the user.

[2] Retrieved July 06, 2015 from http://www.worldwidewebsize.com/

This short historical overview shows that the problem of finding information always seems to be a child of its time, and the term ‘Information Overload Problem’ is just a new term for a century old problem. In every century, solutions were developed to cope with the actual representation of that problem, meaning to make the amount of information manageable for the users. It seems that there is no final and all-fulfilling solution to the problem itself, just a chance to cope with it anew. The main means for coping with the amount of information has never been simply reducing the amount of information, but creating new means of addressing and managing it. Those tables of content, alphabetical indices, dictionaries, compilations, or facet classifications were the means to structure the ever growing amount of information. A historical overview of approaches for summarizing and compiling texts and books is presented in (Blair, 2010b). (Blair, 2010a) summarizes this as ‘[…] it is precisely amid great overabundance that selection and summarizing get all the more valuable’.

The Internet offers an overwhelming amount of information, which easily poses cognitive processing problems to users of all disciplines. Within the context of the thesis, I investigate and use technologies of the Internet, such as search engines, browser extensions, search portals, or databases. Some of these technologies may be helpful when it comes to finding desired information, or can give hints on where a piece of desired information may be located. However, none of these technologies can make the decision how meaningful a piece of information is for a particular user. So the decision about the amount, quality, or meaning of information still has to be made by the user. To enable users to make these decisions and to pick the correct information sources to address their information need, they need to possess a key qualification, which is called ‘Information literacy’ (Gapski & Tekster, 2009, p.12). Users need to be able to distinguish the relevant from the irrelevant information. To some extent, almost all users are able to do so. However, even if users are able to avoid getting lost in the space of search engines and meta search engines, portals, forums, link collections, or recommendations and manage to address their information need in a satisfactory way, the question remains open, how found information is processed. Most of the information found during a web search task gets consumed once, and the respective web page gets closed in the browser and its address is forgotten thereafter. The address containing the precious information is still saved within the browser history, but it is saved undistinguishably together with lots of addresses of irrelevant information. The actual content is always lost. This kind of behavior can be called a throwaway search. Sometimes the users may place a bookmark on precious pages, but this still imposes an effort of organizing those bookmarks. Similarly, limiting the scope of information being presented to the users by some filtering methods cannot be a solution, as this means intentionally hiding available information from them. This has been described as the ‘filter bubble’ by (Pariser, 2011).

Most probably, the users may want to get back to search results they have reached a while ago, maybe because some details were forgotten, some new facet of information is desired, or because someone else asks for advice on a topic that was searched before. However, they can only rarely go back to bookmarks or the browser history, as the website addresses are forgotten or the browser cache gets cleared. Frequently, there is no other chance to get back to search results than to start the search anew, remembering initial queries and trying to find the way back to the relevant pages. The experience with the search topic may help to perform the search faster this time than it was done before. The browser history is only of limited help, as the website addresses are not the main source of information and are therefore forgotten by the user. Additionally, as the history fills up with more and more information, the relevant addresses may lie back in time or are completely forgotten by the browser, or the user has visited so many pages at a specific host that they cannot remember on what specific page the desired information was available.

In a case like 'holiday planning', the user does not look for one specific bit of information that can be assumed to be somewhere on the Internet, but looks more for an overview of a topic. The information need gets ever more complex during the search process. In such cases, there cannot be a single result page covering the information need and it is not possible to form the one optimal search term that could lead directly to the desired result. A long web search will be initiated then, trying to cover different facets of the information need and forming a picture of the result within the users' consciousness or sub-consciousness. At the end of such a research task, there are a number of web pages, each covering a certain aspect of the information need, and a lot of thoughts trying to make sense of gathered information. Storing that information in its entirety is a very hard task until now. Depending on the browser's settings, closing the browser after a complicated research task will result in losing all tabs and its contextual information. Re-finding the results will then get even harder than in the case of the more simple information need, as the user often only remembers that a certain result was found, but not where it was found. The problems inherent with throwaway search get very obvious in these search cases. Ann M. Blair, a Harvard history professor, demands 'methods for building on previous reading and experience, so that we are not reduced to searching for everything anew' (Blair, 2010a).

Finding information anew within a large pool of similar information is also relevant for the described search cases. The search for information in online communities, forums, shops, or small ad portals, in which contributions of many users are collected and published for other users lead to a replication of similar information in highly various levels of quality. Forums may have similar threads, each spanning dozens of pages when it comes to high numbers of contributions. Many shops may offer the same or similar products in several variations. Question answering forums may provide lots of different answers on a specific question, and only the user can figure out what portion of these answers matches their personal information need. During a long web research task, users may have gone through a lot of similar information, trying to make sense of it and to judge its quality. In most cases, only a small portion of information on a page is relevant for the user. The web browser would now save the address of the page, which would lead the user back to the page in its entirety, again posing the effort of qualitative selection to the user.

A web search task will therefore regularly force users to figure out the most important pieces of information from a vast amount of similar information, and to create their personal search result from that. This selection process is always done at least mentally, and very rarely in a computer assisted way. However, this part is important; it helps the overall search result to be more than the sum of its parts. When it gets lost, may it be mentally by forgetting the search details or by not even being created, the damage is done. In such a case, a search would have to be done anew, forcing the users to go back to all the sources of similar information and start the selection process again. Other people with similar information needs also cannot rely on the results that other web searchers gained earlier. Exchanging selected and qualitatively highly valuable information users have created during their searches could have saved whole search processes. This would help users to rely on similar search results and build upon them, refining them with their personal interests, steadily improving the overall quality of the result.

In my thesis, I cover the problems described and work on the topic of developing a possibility to store large web search processes and make them exchangeable. I develop a system that enables the extraction of a large set of pieces of information of various qualities into a highly qualitative set of logically connected information. Its approach is based in equipping the users with digital tools which enable them keeping track of a web search and saving and structuring its results. This helps both showing the full amount of information to the user, and enables the exchange of web search extracts between users.

1.1 Motivating scenarios

This subchapter presents three motivating scenarios, which sensitize the reader for the problems tackled in the thesis and provide a practical glance on possible application cases of the envisioned system. From these scenarios, I extract the research questions of my thesis. The scenarios describe three situations where the system being developed would be needed to cope with the problems of the described situations. The first scenario describes a situation where a user wants to get familiar with a certain topic. The second scenario describes a situation where a web search needs to be exchanged between collaborating workers. The third scenario describes a situation where a web search process is interrupted and needs to be continued later on.

1.1.1 Scenario 1 – Capturing a search process

Frequently, users face the challenge to get acquainted with a topic they are interested in, but they do not know much about until that point in time. The information need in these cases is mostly complex in a way that simply looking up a term in a dictionary or in an encyclopedia does not fulfill the information need. Such information needs can be triggered by various reasons, such as the topics of reports, TV programs, names of places, products, or persons which make their way to the users' attention during a conversation or by consuming some sort of information.

Looking up the interesting topic in the Internet may help to fulfill the information need in such a case. A large share of users will start their search by consulting a search engine and typing in a simple representation of their need in a few words. Up to that time, the information need itself is not well specified; the users only know that they want to acquire some knowledge, without having an idea of the actual details they want to know.

Consider for example the case of a student named Chris, talking to a colleague in an insurance company about coffee consumption. During a watercooler conversation about the quality of coffee in the office, it turns out that the respective colleague knows a lot about coffee. He mentions the names of certain coffee varieties and methods of preparation, which raises Chris' interest. Due to a lack of time, the conversation is very brief, but it still raises Chris' interest. Back home, Chris remembers the conversation with his colleague and starts investigating. As usual, he starts his search in Google with the query 'coffee preparation'. The English Google page returns approximately 90 million results for that query (numbers from November 2015). The query first returns the link of the Wikipedia article on ‘Coffee preparation’. This article covers a lot of details, elaborating on the whole process from green coffee beans to a well-prepared cup of coffee, explaining roasting, grinding, brewing, and extraction. Other topics on that page are common brewing methods and presentation. The student remembers his colleague both talking about the correct choice of beans, in a way that some varieties should be preferred about others, and choosing the correct preparation method.

From these topics, he digs into bean varieties first. To find out more about that topic, he moves from the Wikipedia page on ‘Coffee preparation’ to the page on ‘Coffee beans’. This page tells him more details on the economically most important bean varieties. He learns about distinguishing the varieties and the different origins of coffee. From that, he digs into Youtube, trying to find out about different methods of coffee preparation, using for example Italian coffee makers. He finds lots of videos in very different levels of quality.

He ends his search at this point, having the feeling to have found out a lot about coffee preparation. A week later, he gets the chance to buy an Italian coffee maker in a store. This requires him to go back to his acquired knowledge about choosing the correct granularity of coffee powder and on coffee preparation. At this point, he faces the problem that he can remember to have seen a certain piece of information that would help him with selecting the proper coffee powder for his machine, but he does not remember where he saw it. He can remember some of the Google queries he entered before, but the searches return different results. Thus the search engine provides no orientation in the space of known resources anymore now. He makes similar experiences with the videos he has seen: Some videos were helpful, some not at all. He remembers having found all the videos on Youtube, but which were the helpful ones?

This scenario shows some of the problems users are facing during complex search processes when they are not supported by external tools. Users want to capture search results and informative facts on a topic, because plain bookmarks are less helpful in some cases. As search engines alter their result sets frequently (cf. Lewandowski, 2015, chap.6), performing identical queries will probably not help with re-finding known resources. Some users may try to find ways to overcome the flaw of missing support, such as investigating their browsing history or saving links as bookmarks, some may paste information into text documents or take screenshots. However, all these approaches do only capture fragments of the context and will fail when the value of some information is appreciated after a resource was left behind. This is the point when capturing the trail of a search can be of great value, as it offers the chance to get back to the context of a certain point from a previous search and to extend the search from this point, thus overcoming the flaws of the browser history or bookmarks.

1.1.2 Scenario 2 – Exchanging search results

Consider the context from Scenario 1, where Chris’ interest in coffee preparation gets triggered by talking to a colleague who is really into coffee. Chris starts an extensive Internet search to build up some basic knowledge about coffee preparation and Italian coffee makers. Some days later, Chris talks to a friend named Carol about coffee. He starts explaining that he just dug into the topic and wants to learn more about it, as he is now sensitized for that topic.

Carol explains that she has just bought herself an Italian coffee maker and now wants to get it to work properly. From his previous research task, Chris can tell her that a number of coffees have to be made without cleaning the machine to somehow initialize it and get rid of the machines initial metallic taste. Carol is unsure about which beans to use in that machine and how fine to grind them, and how much coffee powder to put into the machine. Chris can remember having seen a video in which someone put coffee powder into the machine made some remarks about it, but unfortunately, he cannot remember which video it was.

During that conversation between Carol and Chris, some information from some previous search is recalled. This information was found by Chris and gets combined with information from Carol. It turns out that Carol has bought the coffee maker without having acquired much information about it, and Chris has already gathered a lot of information during his search. During the conversation, some information can be recalled, but especially information from the videos cannot be described easily. Similarly, it is hard to recall the names of the helpful videos, as most names are very similar or even identical. So the information that was gained during the search seems to be lost for Chris, and it is very hard to transfer it to Carol. Even harder, information about pages that were not helpful at all cannot be transferred from Chris to Carol. This information is not remembered by Chris, but could help Carol to get into the topic more efficiently.

This scenario shows the problems with transferring search results and whole trails of web search. It is obvious that Carol could have gained a lot of knowledge if she could have seen the results that Chris found helpful, and that it also would have been helpful to see what pieces of information were not leading to the fulfilment of the information need. If Carol could have looked over Chris' shoulders while doing the search, she would have gained a lot of the desired information, but would have needed as much time as Chris needed to do his research task. A compound of Chris' search results, collecting the valuable information and pointing out the valueless resources would help to share the desired information rapidly. This helps to avoid time-consuming detours on valueless resources. Capturing the course of a complex web search process as a trail would offer possibilities like the ones described: It would be possible to see the whole course of another person's search, with all its detours and valuable pieces of information, but could also keep the individually marked valueless resources. All this information could help collaborating searchers to catch up with a previous search and avoid redundant work.

1.1.3 Scenario 3 – Continuing a search

While Scenario 1 and Scenario 2 describe a purely interest-driven complex web search, this scenario shows an application case from a professional domain. Same as in the private context, complex search appears frequently in professional context. Consider the application case from a biomedical researcher. Cancer prevention research has in its first stages not much to do with clinical experiments, but much more with the formulation of hypotheses and a search for evidence that could confirm or disprove these hypotheses. An example hypothesis could be 'A cup of Arabica coffee in the evening helps preventing liver cancer'. It is assumed that certain substances in Arabica beans can have a positive influence to certain cells overnight. Immediate clinical tests would be labor and cost intensive and run the danger that similar studies have already been conducted to confirm other hypotheses. However, the body of biomedical literature is vast, and lots of resources exist to gain further theoretical insights. So the researchers' first way leads to the Internet to find out which data is available to prove the hypothesis right or wrong. One possible path of research could be the following:

- Starting with the 'National Library of Medicine'[3], the researcher looks for the active ingredients of Arabica coffee beans by searching for 'arabica coffee'. It can be found that both polyphenols and ellagic acid are present within the green as well as the roasted coffee beans.
- After that, it should be found out whether there is data that could proof a toxic effect of substances in the roasted coffee beans. In that case, the hypothesis could be discarded immediately. There are three resources for that kind of data on the Internet: DSStox Structure-Browser Database[4], Drugbank[5], and ClinicalTrials.gov[6].
- Now it needs to be checked if there are substances in roasted coffee beans that could damage or alter human DNA. Hints on that may be found within the KEEG Pathway Database[7] or the GeneOntology database[8].
- Next thing to investigate is whether there are substances in roasted coffee beans that could cause cell death. That information can be found within the TOXNET Database[9] by searching for substances that appear in roasted coffee beans that strengthen the activity of some specific genes but reduce the activity of other specific genes.
- A further step would be to search for similar publications within the database of PubMed[10], in which a large share of medical research publications is filed. Typically, publications in PubMed are searched by their active ingredients. The desired publications should answers questions like 'In which dosing has the active ingredient been used so far?', 'Has this active ingredient been tested in living objects?', or 'Which effects have been proved with that active ingredient so far?'.

At this point in time, the researcher gets interrupted by an urgent experiment. A time critical analysis has been completed and requires the researcher's immediate attention, as procrastination would further alter the cell cultures in the other experiment. Therefore, the researcher has to interrupt and abandon the already begun search process. The interruption takes a long time, and when the researcher comes back to his computer the next morning, most probably the data, but at least the cognitive context of yesterday's search is gone. A colleague would now have time to extend yesterday's search, but this specific combination of opened web pages is lost.

On the one hand, this research task is labor intensive, but as there is no single information source supplying all that information, all these single pieces of information have to be collected to form a larger picture. On the other hand, it ensures that as much information as possible about the substances, their ingredients and their effect on living organisms is collected. This helps avoiding unnecessary work and bases the current hypothesis within a basis of related work, and

[3] Retrieved May 24, 2015 from http://www.nlm.nih.gov/
[4] Retrieved May 24, 2015 from http://epa.gov/dsstox_structurebrowser/
[5] Retrieved May 24, 2015 from http://www.drugbank.ca/
[6] Retrieved May 24, 2015 from https://clinicaltrials.gov/
[7] Retrieved May 24, 2015 from http://www.genome.jp/kegg/pathway.html
[8] Retrieved May 24, 2015 from http://geneontology.org/
[9] Retrieved May 24, 2015 from http://toxnet.nlm.nih.gov/
[10] Retrieved May 24, 2015 from http://www.ncbi.nlm.nih.gov/pubmed

therefore saves money. As such it is of high importance to do this preliminary research as thorough as possible. However, this process can be arbitrarily complex and include more resources if it comes to more complex hypotheses. It is possible that this amount of information exceeds the cognitive and temporal limits of a researcher, such that the complex search process has to be interrupted and continued at a later point in time, or maybe by another researcher, e.g. when another urgent experiment requires the researcher's attention and forces the interruption of the ongoing search process. Whether a search process has to be extended gets obvious during the search and cannot be planned. As complex searches are very rarely completed within one session, classical throwaway searches may not be the right means to cope with those questions. Researchers may be interested in capturing the main results of their searches to be available when the search process shall be resumed. The same is true for exchanging a search task: Researchers might want to capture the main results and the resources lacking valuable information for exchanging the search process and to avoid that their colleagues run into the same dead-ends as they might have done before. In this case, capturing the trail of a search would enable users to store the search process, and could provide a way to mark valuable and valueless resources. A search could be stored and abandoned, and be recalled at a later point in time or exchanged with a colleague, who could easily see where relevant information has been found.

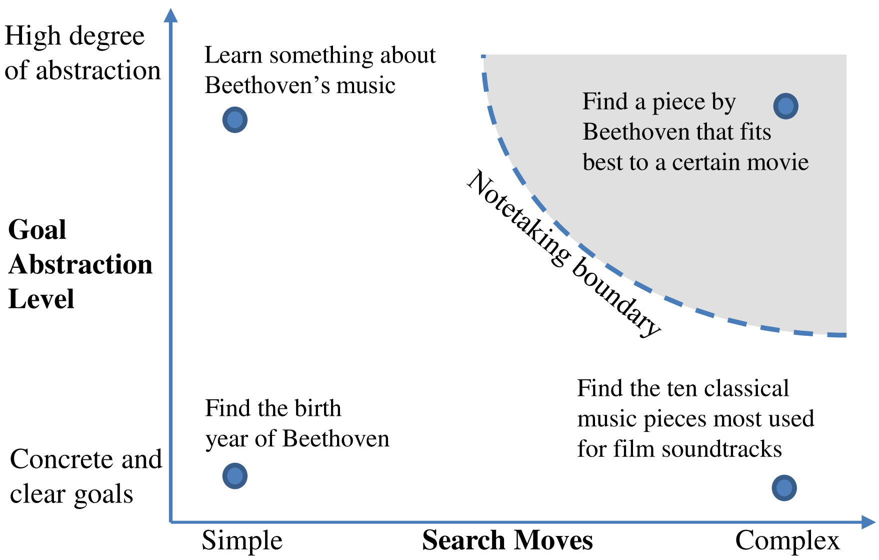


Figure 1: Schematic view of the concept of the notetaking boundary (Aula & Russell, 2008).

The scenarios are exemplarily for a large number of individual, complex search tasks. All types of users are confronted with such problems when they take the Internet as a central source of information. At the end of their searches, users face the challenge of storing or remembering the valuable pieces of information, as the amount of information necessary to fulfill the information need exceeds the cognitive processing abilities of the users. (Aula & Russell, 2008) categorize search tasks by the abstractness of the search goal and the complexity of the necessary search moves. If the combination of both exceeds a certain individual level, it gets necessary to support the search process by taking notes. The authors call this the 'notetaking boundary' (cf. Figure 1, more details in Chapter 2.2.1). In my thesis, I develop support for search cases beyond the notetaking boundary and present an approach for capturing the trail through the Internet as an information medium, capturing detours as well as core results of a web search.

1.2 Motivation

By the work done in the thesis, I provide support for the problems described in the three scenarios. In my thesis I deliver a tool supporting users during extensive web search processes by capturing valuable information as well as the context of a search. My approach both delivers shortcuts to resources, but also preserves context: 'Making and using shortcuts skillfully and responsibly requires judgment, too' (Blair, 2010a). The three scenarios exemplify how users may want to interact with search results and help identifying challenges in the described situations:

1. **Scenario 1: Capturing a search process**
   In case of the coffee preparation search, the information need is not very unique, which results in an overwhelming amount of search results of various qualities. Therefore it could be very useful to capture the search and the valuable pieces of information found, such as informative videos, articles, and other information details within their context. These pieces form a trail of visited resources and a set of relevant pieces of information. Annotating the found resources would help the user to further specify the value of a resource or to describe the contained information in brief words. For example, whether a video provides only background information or is an instructional video.
   The **challenge** raised by this scenario is the **capturing of search results within their context**, going beyond plain browsing history or bookmarks. It can be extended by the marking the more and the less valuable search results.
2. **Scenario 2: Exchanging search results**
   This leads to the second scenario, in where the two colleagues try to share their information about a topic and want to point each other to useful resources. It would be very helpful for persons being new to a topic to dig into the search results other users have generated before, especially when the users know each other and can talk about a search. However, even if they do not know each other, a view at other users' search results may point users to completely new aspects of a search they may not have thought about. Apart from reenacting a given search, users provided with search results would soon start to discover personally interesting aspects of a search. In the described case, Carol would be interested in a video manual for her model of that coffee maker. She would enrich the existing results by information on her specific model of the coffee maker and as such add a new aspect to the overall results.
   In this scenario, the **challenge** is in **enabling the exchange of search results between known or unknown collaborators to achieve asynchronous collaboration.**
3. **Scenario 3: Continuing a search**
   In the third scenario, the biomedical researcher starts a complex search on a newly developed hypothesis. A lot of results can be collected by browsing through the known databases, and most certainly, other aspects of the hypothesis would be discovered during the search process and add to the search. This growing body of information is stored automatically. When the researcher gets interrupted by an urgent task, the process of information gathering gets stopped. When taking up the search process after some time, the researcher would have to update the gathered information to see if new findings have been published in the meantime. The browser would treat all visited sites

equally, and store the addresses of both the relevant and irrelevant sites. Without tool support, the researcher would have to replicate large parts of the search task until the necessary information basis is built up again, which enables the researcher to dive deeper into the information space.
The third scenario raises the **challenge** of **resuming and exchanging interrupted searches and enabling the updating of information on a certain topic.**

The described scenarios stand exemplary for a large number of search activities. They illustrate three challenges arising from problems connected to the search for information with – at first – unspecified information needs. In this thesis, I develop, implement, and evaluate an approach to overcome these problems by capturing search results within their context, making them reusable, extendable, and reproducible. This combination is the main, new feature of the implemented system. My approach enables exchanging search results between known or unknown collaborators to achieve asynchronous collaboration, and allows resuming interrupted searches and enabling the updating of information on a certain topic. On the one hand, this approach captures search results of particular users, and on the other hand, it leaves freedom to the user what to search and which results to pick for a personal collection. This part of a search process cannot be done automatically. (Nelson, 1994) names the selected information 'personally meaningful information'.

In addition to the challenges derived from the scenarios, I want to highlight the importance of the concept of 'information literacy' for the user's personal set of competences. A tool solving the challenges improves user discovery, selection, and capturing of information and raises the users' information literacy. This is defined as 'the ability to detect an information need with respect to a certain problem, and to discover, acquire, evaluate, and use that information effectively'[11] (translated from the German source in: (Gapski & Tekster, 2009, p.13)). The origin of the term 'information literacy' is based on the Anglo-American education system from the late 1970's (Ingold, 2005, p.10) and was aimed towards the efficient use of libraries. Patricia Breivik describes information literacy as 'an integrated set of skills and the knowledge of tools and resources. […] Information literacy is developed through persistence, and attention to detail, and a critical evaluative view of the material found' (Breivik, 1985). In the English language area, some models of 'information skills processes' have developed over time, each naming a set of steps as key in the process of gathering and evaluating information. An overview of three models of information skills is given in (Eisenberg & Brown, 1992), contrasting some models of information literacy. In detail these are the definitions of (Kuhlthau, 1991), (Irving, 1985), and (Stripling & Pitts, 1988). As the authors detect certain overlaps and similar concepts in each of these three models, they contrast them with the model of information problem solving from (Eisenberg & Berkowitz, 1988). From this model, some human abilities can be derived that seem to be central elements of information literacy:

- Task definition,
- Information seeking strategies,
- Locate and access relevant information,
- Use of information,

[11] German origin of the citation: [Informationskompetenz ist die]'Fähigkeit, die es ermöglicht, bezogen auf ein bestimmtes Problem Informationsbedarf zu erkennen, Informationen zu ermitteln und zu beschaffen sowie Informationen zu bewerten und effektiv zu nutzen', in: (Gapski & Tekster, 2009, p.13).

- Synthesis of information, and
- Evaluation of information.

Teaching information literacy gains a growing role in society: In 2009, the month October was declared the 'National Information Literacy Awareness Month'[12] by US President Barack Obama, having the goal to 'learn the skills necessary to acquire, collate, and evaluate information for any situation'[13]: The 'National Information Literacy Awareness Month highlights the need for all Americans to be adept in the skills necessary to effectively navigate the Information Age'[14]. However, information literacy cannot be decreed or produced, but is the result of a set of personal abilities and their technical support. Especially during the process of searching, this support is usually not satisfactory (Weitz, 2009) and needs to be improved. Therefore, also technical tools can be of a help to 'lead people out of their technology-induced inability'[15] (English translation of a German citation from (Ingold, 2005, p.34)). Especially during search processes, users are depending on a combination of cognitive abilities and technical tools. Improving the tools helps to augment cognitive abilities of the user and helps to optimize the overall results.

This thesis thus makes an impact on assuring and improving a modern key competency, by developing, implementing, and evaluating a tool that supports the process of web search. This is achieved by supporting the users to save the context of web searches as well as synthesizing key findings for later reuse, evaluation, extension, or exchange with other users. Therefore, it supports core elements of information literacy, such as the location, use, synthesis, and evaluation of information. At the same time, it enables users to rely on former web search results when performing repeated or similar web searches.

1.3 Goals of my thesis

The scenarios described earlier help identifying user challenges. From them, the overall goals of my thesis can be deduced: Users shall be supported during and after their complex web searches by the possibility to save main search results as well as the context of these results. This includes the possibility of recovering the search process and context after a time and being able to extend the results or share them with other users to support asynchronous collaboration by relying on previously generated information networks. In this thesis, I develop a concept, implement a system for that purpose, and evaluate it.

I take up the core idea of building and storing trails during web search. To achieve this, the users' interaction with the Internet needs to be captured and transformed into a meaningful visualization, to ensure that users are able to understand this information and draw conclusions from it. The envisioned system would also need to provide possibilities to interact with the generated

[12] Retrieved November 17, 2014 from http://www.whitehouse.gov/the_press_office/Presidential-Proclamation-National-Information-Literacy-Awareness-Month
[13] Ibid.
[14] Ibid.
[15] German origin of this citation: 'Herausführung des Menschen aus seiner technologieverschuldeten Unfähigkeit' (Ingold, 2005, p. 34).

information, such as marking both valuable and valueless resources. For easing collaboration, ways to share search trails have to be found.

In order to achieve tool support for asynchronous, discontinuous, collaborative, complex search tasks, I analyze search processes with the help of theoretical models and evaluate related systems for logging search and for supporting collaborative search. I develop a concept for my envisioned system, implement it, and evaluate it with both quantitative and qualitative user studies.

Adding to these high-level goals of the thesis, some basic requirements for the envisioned system can be derived from the scenarios, as they describe the most basic sequences of interaction with the system. These requirements can already serve as a basis for the review of the related theoretical work or systems, but also as a basic scheme for the later evaluation of the system. The requirements are:

- Saving the context of a web search: Store the course of a web search, with all its paths and detours. This includes relevant and irrelevant information, both being visualized for the user.
- Visualizing the course of a web search: In order to present all collected information to the user, a form of visualizing the web search has to be chosen that eases visual evaluation of the course of the web search.
- Saving resources during web search tasks: Store personally meaningful information during web searches with minimal effort, save it for later use, or for sharing between collaborators, or for evaluation.
- Pointing to resources: Point users to unknown, but relevant topics during web searches, provide a mechanism to extract topics from web resources and visualize these topics in a proper way to the user.
- Continuing web searches: Enable the continuation of web searches by storing the web search remotely and persistently and provide a function to recreate the web search from that information.
- Exchanging web searches: Enable the exchange of web searches between collaborators by a simple mechanism.
- Lightweight technical support: Ensure that the search support tool works in a lightweight way, avoiding major installation efforts or an overwhelming scope of functions.

These requirements already form a first picture of the envisioned system. They could be realized both by a desktop system or an Internet browser plugin, accessing an online storage that stores the web search data objects. Apart from these technical requirements, the thesis also investigates how users interact with the envisioned system and confirms the system's effectiveness and efficiency.

1.4 Research questions

The research questions I develop in this subchapter reach from rather technical questions to questions regarding the interaction with the system as well as the collaboration between users with the help of the system. The overall research question of my thesis is: **'Can search trails**

provide support for complex web search and how should tool support look like?'. The term 'complex web search' covers the case of discontinuous, asynchronous, collaborative, complex search. All central terms of the thesis, such as 'search trail', 'exploratory search', or 'complex search' are defined in Chapter 2.2.2.

This research question can be divided into a number of smaller questions, each spanning one aspect of it. The first group of research questions (**RQ1 'Feasibility'**) deals with the feasibility of offering support for complex web search tasks by constructing search trails. In order to ease referencing the research questions throughout the following chapters, each research questions is assigned a short name.

- **RQ1.1 'Search trails valid?'**
  Are search trails a means to resemble the course of a user's web search?
  An answer to this question will be found by a combination of literature review, searching for systems that have taken this effort – if existing – and by implementing and evaluating such a system.
- **RQ1.2 'Which functions?'**
  Which functions does a system need to offer to the users to support capturing both the context as well as key findings of a web search?
  Answers to this question will be derived from the chapter of related work, in which the analysis of concepts, models, and related systems leads to a set of functions to be offered. Verification of the decisions made can be found during evaluation.
- **RQ1.3 'Provide help?'**
  Does a system with the functions from RQ1.2 provide help and support for users during complex search processes?
  This question is answered with in initial evaluation where users are shown the system and can gather first experiences with it.

Question group RQ1 will be answered with the related work in Chapter 2, the evaluation of the related work in Chapter 3, and the results of the first user study in Chapter 5.1. I develop a system called 'SearchTrails' for answering the research question from question group RQ1. The second group of questions (**RQ2 'Value'**) covers the value of search trails as artifacts in individual and collaborative search scenarios and their impact on the quality of search processes. The system 'SearchTrails' is evaluated further to answer the research questions from question block RQ2.

- **RQ2.1 'Ease complex search?'**
  Does SearchTrails help users coping with complex search tasks?
  To answer this question, the concept of complex search tasks has to be split up into its single components with the help of theoretical models. If the system can help with each of the core components of complex search, it does also help the users when performing complex search in general.
- **RQ2.2 'Ease continuation?'**
  Does SearchTrails help users with continuing web searches?
  This question will be answered with a second evaluation, in which this case can be tested among a set of participants.

- **RQ2.3 'Ease sensemaking?'**
  Does visualizing the course of a web search help with understanding what other users have done during a search?
  This question is answered after a second evaluation, when the results get evaluated by recreating the visual paths of the users' searches.

- **RQ2.4 'Ease collaboration?'**
  Does SearchTrails ease collaboration between users by exchanging web searches?
  This question is also answered during a second evaluation in which web searches are exchanged between users.

- **RQ2.5 'Lightweight?'**
  Is SearchTrails lightweight enough to be attractive for users?
  This question can only be answered on a meta-level, most likely by personal opinions of the users towards the system. Therefore, it will be answered by asking the users about the perceived lightweightness of the implemented system.

Question group RQ2 will mostly be answered by the study results from the second user study of the thesis in Chapter 5.2. However, some of them can partially be answered by the analysis of related models and systems, others by evaluating an implemented prototype with users. This needs to be done multiple times, as the thesis follows an iterative approach of continuous learning and improvement of the prototype.

1.5 Contribution of my thesis towards the topics of information science

In order to point out the thesis' contribution towards the state of the art in information science, it is helpful to first embed it into the topics of information science research. This embedding is not always possible in an unambiguous way, but shows intersections with topics touched and therefore helps getting an impression of the relevant research areas. This subchapter embeds this work into the four large subareas of information science. After that, I evaluate how my thesis fits into the Computing Classification System[16] (CCS) of the Association for Computing Machinery (ACM) and point out the research contribution of my thesis.

This work touches three of the four subareas of core computer science, which are theoretical, practical, technical, and applied computer science (Schubert & Schwill, 2011, p.5 ff.). In the field of theoretical computer science, dealing with the research and development of algorithms for the theoretical representation of processes, the foundations of my thesis are located. The theory of graphs and the algorithms calculating their properties can be considered being parts of theoretical computer science. The area of practical computer science covers the development of solutions and the translation of machine-independent program code into machine code, operating systems or database concepts. Within this work, the development of software prototypes, data storage concepts and the transmission of data between parts of the software system are located within the area of practical computer science. The area of technical computer science is

[16] Retrieved November 20, 2014 from http://www.acm.org/about/class/class/2012

less visible in this work. This area covers the development of all technical tools helping to realize computer systems. Of course, hardware is used for this work, but it is not concerned with developing it. The area of applied computer science deals with supporting work processes by the means of computers. As the main goal of the thesis is concerned with enhancing the process of web research by developing a suited tool for this application, this work can be considered being located primarily in the field of applied computer science.

The research-focused ACM classification was introduced in 1964 and faced several refinements in the years 1991, 1998, and 2012. This classification was set up to classify publications with relation to computer science and is well appreciated since. For my thesis, I use the 2012 classification system[17]. For this version of the CCS, there no numbering system exists yet. The ACM classification tries to structure the vast landscape of computer science literature by providing thirteen categories from 'General and reference', touching topics like 'Hardware', 'Networks' and 'Theory of computation' and leading to 'Social and professional topics'. Within this classification, the thesis can mainly be located into the topics of 'Information systems' and 'Human-centered computing'.

The topic 'Information systems' is subdivided into five subtopics, of which 'World Wide Web' and 'Information retrieval' are most relevant. Within the topic 'World Wide Web', the topic '**Web searching and information discovery**' shows the largest intersection with my thesis. Although this topic is then divided into six further subtopics, the best match emerges with 'Web searching and information discovery' itself. My thesis contributes SearchTrails, a software tool enhancing the users' abilities in complex web search and information discovery by pointing towards relevant keywords during the web search process. Within the topic of 'World Wide Web', a minor intersection emerges with the subtopic '**Web log analysis**' located under the topic 'Web mining'. My thesis builds upon related work for web log analysis and offers a way for users to efficiently evaluate and analyze unfiltered search trails. Within the subtopic 'Information retrieval', a minor intersection emerges with the subtopic '**Collaborative search**'. My thesis contributes a novel way of collaborative search by exchanging unfiltered search trails and the continuation of the collaborator's search trails.

In the topic of 'Human-centered computing', a large intersection emerges within the topic '**Computer supported collaborative work**', as my thesis describes a way of asynchronous collaboration between searchers, allowing them to dive into each other's search results and build upon previous work. A minor intersection emerges within the topic of '**Information visualization**'. With the concept of search trails, my thesis contributes a novel way for visualizing both the course of the user's navigation through the Web as well as pointing out valuable resources found during the complex search process.

Table 1 summarizes the aforementioned categorization and research contributions and helps matching the intersected topics from the CCS to the topics of my thesis.

[17] Retrieved November 20, 2014 from http://www.acm.org/about/class/class/2012

Table 1: Intersection of topics between my thesis and the ACM 2012 classification and research contribution.

Topic from the CSS	Intersection	Research contribution of my thesis
Information systems → World Wide Web → **Web searching and information discovery**	**Large**	The system 'SearchTrails', enhancing the users' abilities in complex web search and information discovery by pointing towards relevant keywords during the web search process.
Information systems → World Wide Web → Web mining → **Web log analysis**	Minor	Search trails offer a way for users to efficiently evaluate and analyze unfiltered search trails.
Information systems → Information retrieval → Users and interactive retrieval → **Collaborative search**	Minor	A novel way of collaborative search by exchanging unfiltered search trails and continuing the collaborator's search trails.
Human-centered computing → Collaborative and social computing → Collaborative and social computing theory, concepts and paradigms → **Computer supported collaborative work**	**Large**	A way of asynchronous collaboration between searchers, allowing them to dive into each other's search results and build upon previous work.
Human-centered computing → Visualization → Visualization application domains → **Information visualization**	Minor	A novel way for visualizing both the course of the user's navigation through the Web as well as pointing out valuable resources found during the complex search process.

Considering this embedding into the topics of computer science literature, it becomes obvious that my thesis is located in the intersection of two large branches of computer science: Computer supported cooperative work and web search. On the one hand, the thesis aims for supporting complex search tasks by enhancing information discovery, and on the other hand, it aims for collaboration on the gathered information by exchanging the information networks and providing the possibility to extend them.

Results of my thesis have been published in three full papers at national and international conferences. The main qualitative results on the general approach of building search trails during complex search processes from the first user study have been published in (Franken & Norbisrath, 2014b). The results on visual evaluation of given search trails from the first user study have been published in (Franken & Norbisrath, 2014a). The results on the value search trails for collaborative search and their impact on the usability of SearchTrails are published in (Franken et al., 2015), while results on the improvement of the quality of collaborative search results are published in (Franken et al., 2016).

This embedding into the topics of computer science literature helps defining the scope in which the thesis is located. This actual embedding into the topics may not always be unambiguous, but especially the work within the intersection of topics bears the possibility to provide new and unique solutions for existing problems.

1.6 Research methodology

For all kinds of research in different scientific disciplines, a number of research methodologies exist, which model the process of doing research and of achieving results. One example for a research approach from the biomedical domain is described in Scenario 3 'Continuing a search', where a researcher first develops a hypothesis, then checks a number of databases for first hints on the hypothesis, and only performs practical work when first theoretical results already point into the desired direction. This example of 'hypothesis driven research' (Smalheiser, 2002) is only one approach for biomedical research.

For the discipline of Information Systems, most of the research done can be clustered in one of the two research methodologies: behavioral science and design science (Hevner et al., 2004). While behavioral science is built around explaining the human behavior, design science has a more artifact-centric focus. (Hevner et al., 2004) explain that the 'behavioral science paradigm seeks to develop and verify theories that explain or predict human or organizational behavior', while the 'design science paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts'. The term 'design science' was coined by Richard Buckminster Fuller, who is well-known for being an architect, but also worked with design from a more scientific perspective. He initiated the 'World design science decade 1965-1975' (Buckminster Fuller & McHale, 1967), in which design in architectural contexts was seen from a scientific perspective. The three motivating scenarios in this chapter point towards situations where human capabilities can be extended by innovative artifacts and therefore indicate that the design science methodology is the most suited research methodology for the work done in my thesis.

My work has therefore been done following the design science methodology. The main goal of this methodology is to gain 'knowledge and understanding of a problem domain and its solution' by 'the building and application of the designed artifact' (Hevner et al., 2004). In order to achieve this, Hevner et al. propose a conceptual framework, consisting of seven guidelines. The seven guidelines are (based on (Hevner et al., 2004, p.82 ff.)):

1. **Design as an artifact**
   'A research project following the design-science guidelines must produce a viable artifact, may it be an application, a model, or an instantiation.'
2. **Problem relevance**
   'Design science aims for developing technology based solutions for important and relevant business problems.'
3. **Design evaluation**
   'Whether a product developed in a design science process has a potential practical impact needs to be shown in a well-conducted evaluation.'
4. **Research contributions**
   'The design science process must provide clear and verifiable contributions of at least one of the following types: design artifact, design foundations, or design methodologies.'

5. **Research rigor**
   'The application of rigorous methods during the construction and evaluation of the design artifact is mandatory.'
6. **Design as a search process**
   'The search for an effective artifact must make use of proper methods to reach the desired needs, but must obey the rules of the problem environment.'
7. **Communication of research**
   'The results of a design science research process must be presented to the matching audiences.'

These seven guidelines do not need to be applied in a strict order or as part of a routine. Instead, researchers are demanded to use their creative skills to figure out which of the seven guidelines may help most at what phase of the ongoing research. Design science as a research method has also been described by other authors. (Vaishnavi & Kuechler, 2004) present an overview of design science research approaches for the information system science domain. (Peffers et al., 2007) present a more formal model for design science research processes, redefining some of the original seven guidelines to form a process model of six consecutive steps. At the end of the six steps, this model features a connection to earlier steps in the model, called 'process iteration', thus making the process iterative.

The term 'iterative software design' has been coined by Jakob Nielsen and was originally aiming for the incremental improvement of user interfaces (Nielsen, 1993). Very soon, it turned out that iterative design can also help in software development, as it has a number of benefits (based on (Kruchten, 2001)):

- Detection of misunderstandings early in the development cycle, making corrections relatively cheap.
- Enable and encourage frequent user feedback, to keep customers involved in the process.
- Developers need to focus first on the most critical issues for a project.
- Iterative testing determines the projects status frequently.
- Inconsistencies between requirements, design, and implementations are detected early and frequent.
- The workload is balanced throughout the development process.
- Lessons learned can be integrated into the software development process very fast.
- The stakeholders get frequent information about the status of the project.

An early model of iterative software development is depicted in Figure 2. It describes the iterative development process of the POLITeam system, being called the 'cooperative and constructive design process' (Klöckner et al., 1995, p.23). This model is based on previous work towards evolutionary systems development, as described in (Eason, 1982, p.207 ff.) and (Floyd & Keil, 1983).

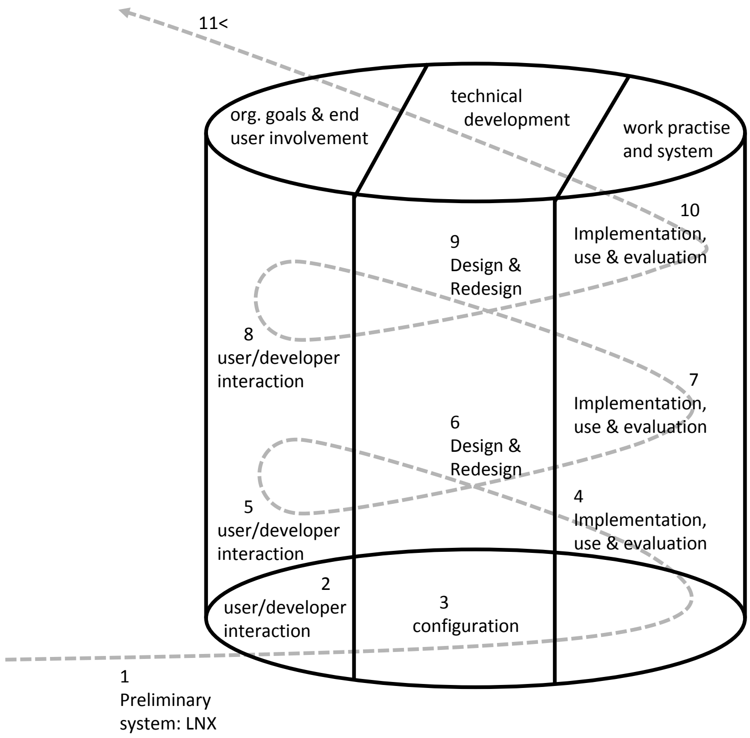


Figure 2: Model of the cooperative evolutionary design process (Klöckner et al., 1995, p.23).

Figure 3 (Kruchten, 2001) visualizes the essence of iterative software development, the possible starting and end points, and the steps during the software development process. This model is more generic than the helical model described in (Klöckner et al., 1995) and therefore allows more flexible iterations.

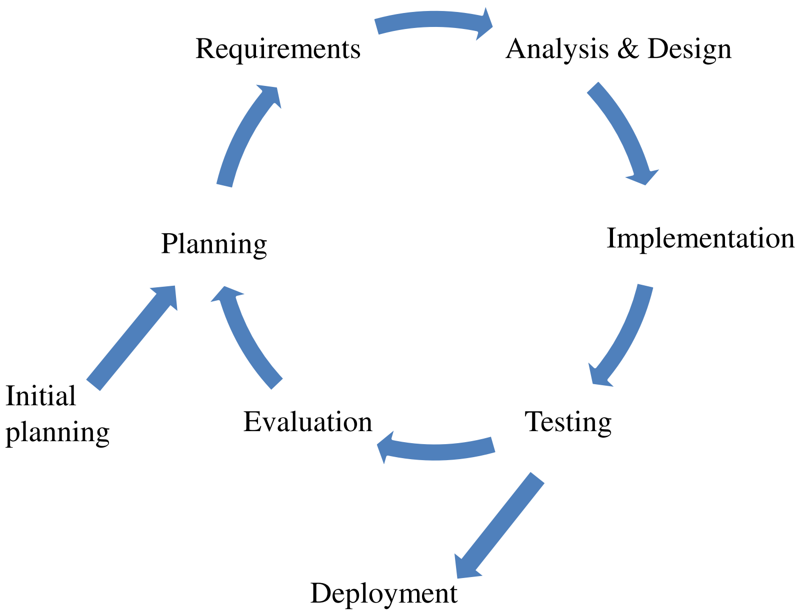


Figure 3: Model of iterative software development (Kruchten, 2001).

The benefits of iterative software development suggest following the combined approach of design science with iterative software design, as proposed by (Peffers et al., 2007). My software prototype is therefore developed in an iterative process, which includes multiple phases of requirements collection, design, implementation, testing, and evaluation. During the respective chapters in the thesis, some parts of these phases are described in more detail than others, e.g. the evaluation takes more room than the testing of the prototypes.

The overall design science research methodology is followed throughout the chapter, and the actions done match the seven guidelines. In the last chapter of my thesis, I summarize how the larger parts of the thesis can be matched with the seven guidelines and confirm that the design science research methodology has been consistently followed.

1.7 Conclusion and outlook

This chapter shall convince the reader of the relevance of my work. My analysis of the historical origins of the information overload problem shows that always innovative ways have been found for coping with information overload. In three scenarios I show where – from today's point of view – necessary support is missing. This holds for supporting individual and collaborative search processes as well as for offering support for resuming interrupted searches. Support for these scenarios could be ensured by capturing and exchanging the user's search trails of search process.

Three challenges should be overcome with this thesis, which are capturing search results as a whole, enabling the exchange of search results between collaborators, and resuming interrupted searches. The goal of my thesis is supporting users during and after complex collaborative web search processes by capturing search trails. My main research question is **'Can search trails provide support for complex web search and how should tool support look like?'**. I want to investigate whether search trails can provide support for complex web search and what is their value. The research contribution of my work is mainly to the fields of web search support and collaborative work. Altogether, I develop, implement, and evaluate an approach which allows capturing the course of the user's web search and the exchange of search trails as collaboration artifacts. So far, this was not possible.

The rest of the thesis is divided into six chapters. In Chapter 2, I provide an overview of the theoretical foundations and thematically related systems for this work, as well as an overview of the state of the art. In Chapter 3, I describe the development of requirements for the envisioned system, based on the core findings of the related work. In Chapter 4, I describe the implementation and the different cycles of the iterative design process of the system. Chapter 5 covers the evaluation, while Chapter 6 presents a conclusion and a summary of my thesis.

Chapter 2
Related work

In Chapter 1 of my thesis, I present a motivation and describe three scenarios in which complex web search processes are currently lacking support. In order to be able to build upon these scenarios and to offer support for asynchronous, discontinuous, collaborative, and complex search processes, I develop an overview of related work to make sure that the envisioned approach has not been followed so far. Therefore, this chapter presents an overview of a range of topics related to providing support for asynchronous, collaborative, complex search tasks by building search trails. It is worth mentioning that the overviews of related theories, models, resources, and systems are exemplary for their respective category. The presented approaches cannot be fully exhaustive overviews of related systems. Such overviews would be immediately outdated and exceed the scope of the thesis.

The overview of the related work is divided into eight subchapters. The first subchapter presents historical origins of the research idea, while the second subchapter takes up theoretic approaches related to the topic. The third subchapter takes a look on how real users actually perform web search. The fourth subchapter then presents related techniques, concepts, and approaches that touch the topic of my thesis and distinguish the topic of my thesis from the related work. The fifth subchapter presents both related search logging systems as well as related collaborative search systems and analyzes their features and weaknesses. The sixth subchapter presents a brief overview of related work for other parts of the thesis, such as information visualization. In the seventh subchapter, I finally present some recent trends regarding search trails and search support, while the last subchapter concludes the related work.

The overview of related work has two main goals: It distinguishes my approach from existing approaches and provides a set of related systems to build upon. For this purpose, I develop two comparing tables for the two most important fields of the thesis, namely search logging systems (cf. Chapter 2.5.1) and collaborative search support (cf. Chapter 2.5.2). This chapter also delivers first indications on answering the research questions RQ1.1 ‘Search trails valid?’ in Chapter 2.1 and Chapter 2.7, and for the research question RQ1.2 ‘Which functions?’ in the comparing tables in Chapter 2.5.1.2 and Chapter 2.5.2.2.

The results of this chapter will be evaluated further in the thesis. In Chapter 3, I draw conclusions from the presented related work and evaluate the comparing tables of search logging systems and collaborative search support systems developed in Chapter 2.5. This results in a set of requirements and derives technical features for the system called ‘SearchTrails’, which is realized in my thesis.

2.1 Historical origins of the idea of search trails

The idea of search trails is older than most people think; it dates back to the mid-1940's and was picked up only a few times in the meantime. This chapter presents a historical overview of the ideas and approaches towards building search trails.

The historical origins of the idea of capturing the trail of navigation date back to Vannevar Bush in 1945. Together with John H. Howard, Bush invented the 'rapid selector', a device that allowed high-speed referencing of information stored on microfilm (Caspi et al., 2004). These developments influenced his essay 'As we may think' (Bush, 1945), in which he described the 'memex' device, which was mostly a thought experiment. Bush suggests that the memex could keep its data on 'improved microfilm' (Bush, 1945, chap.6), such that it is a large microfilm reader, being equipped with electro-mechanical controls. The memex should have the form of a desk ('It consists of a desk, and while it can presumably be operated from a distance, it is primarily the piece of furniture at which he works', (Bush, 1945, chap.6) and therefore be the working place of the user, equipped with screens to consume mostly textual information ('On the top are slanting translucent screens, on which material can be projected for convenient reading.', (Bush, 1945, chap.6)). Bush then describes scanning through and working with information on the memex, such as consulting books, scanning through pages, or adding notes and comments to the resources stored. Most readers interpret Bush's concepts as one of the first descriptions of hypertext and its links between documents (Caspi et al., 2004).

Bush already argues that the memex eases information processing, but 'the human mind does not work that way' (Bush, 1945, chap.6), where he means finding information in a linear way, with only one way to address a certain resource. Therefore, he develops the concept of 'trails' in Chapter 7, claiming: 'The process of tying two items together is the important thing'. He then describes the process of accessing and manually tying two resources together, building a trail from that, giving a name to the emerging trail, and storing it to the memex. This way, the memex allows the associative connection of valuable resources, which is close to the associative human cognitive processes. He describes scrolling through more documents by some sort of physical lever to pull, and mentions the possibility to integrate one resource into multiple trails: 'any item can be joined into numerous trails' (Bush, 1945, chap.7).

Bush describes the use case of a single person creating a trail, who is interested in the history of bow and arrow as follows: 'The owner of the memex, let us say, is interested in the origin and properties of the bow and arrow. Specifically he is studying why the short Turkish bow was apparently superior to the English long bow in the skirmishes of the Crusades. He has dozens of possibly pertinent books and articles in his memex. First he runs through an encyclopedia, finds an interesting but sketchy article, leaves it projected. Next, in a history, he finds another pertinent item, and ties the two together. Thus he goes, building a trail of many items. Occasionally he inserts a comment of his own, either linking it into the main trail or joining it by a side trail to a particular item. When it becomes evident that the elastic properties of available materials had a great deal to do with the bow, he branches off on a side trail which takes him through textbooks on elasticity and tables of physical constants. He inserts a page of longhand analysis of his own. Thus he builds a trail of his interest through the maze of materials available to him.' (Bush, 1945, chap.7).

Bush's use case is not only applicable to the memex, but also to the navigation through the Internet, and it could possibly help the users in the described motivating examples from Chapter 1. Later, Bush even goes beyond that initial use case. He describes an example of two users exchanging search trails, with one user having a benefit from another user's preliminary work: 'And his trails do not fade. Several years later, his talk with a friend turns to the queer ways in which a people resist innovations, even of vital interest. He has an example, in the fact that the outraged Europeans still failed to adopt the Turkish bow. In fact he has a trail on it. A touch brings up the code book. Tapping a few keys projects the head of the trail. A lever runs through it at will, stopping at interesting items, going off on side excursions. It is an interesting trail, pertinent to the discussion. So he sets a reproducer in action, photographs the whole trail out, and passes it to his friend for insertion in his own memex, there to be linked into the more general trail.' (Bush, 1945, chap.7).

In Chapter 8 of his essay, Bush develops some scenarios of people exchanging trails and enriching their own searches by trails that are generated by professionals. In fact, Bush describes asynchronous collaboration for complex search processes by generating and exchanging search trails. His use cases include both exchanging trails between searchers that know each other, but also continuing search trails that have been created by unknown and more professional collaborators. Nevertheless, it seems that there have been no approaches of taking the term 'search trails' literally, for example by implementing solutions that realize his visions.

Bush's idea has not been picked up for a long time, as the technical prerequisites were simply not available. Even if hypothetical approaches towards a more complex system than hypertext were made in the project Xanadu (Nelson, 1960) in the mid-1960's, hypertext was not to come. Hypertext was not realized until the mid-1980's, until first experimental hypertext applications were implemented. An early commercial hypermedia system was HyperCard, invented by Apple Computer before the World Wide Web was available (Goodman, 1988).

Marcia Bates is one of the first authors to pick up the concept of trails, even if she does not explicitly cite Bush in her paper on 'The design of browsing and berrypicking techniques for the online search interface' (Bates, 1989). As early as 1989, she describes a model of search processes called 'berrypicking', which she explicitly designs for online information systems. She claims that this model comes closer to human information seeking than the already known approach of 'information retrieval', where one document is assumed to perfectly fit the user's information need. She describes users going through several resources when conducting a search (in contrast to sticking to one source of information for the complete search), and selecting pieces of information from the various resources (Bates, 1989, p.409). Every piece of information contributes to the overall picture. Bates names these valuable pieces of information 'berries', which are picked like berries from different bushes when strolling through a forest. I present more details on Bates' model in Chapter 2.2.4.

A similar approach was presented in 2006, when Fox et al. introduced software called 'stepping stones and pathways' (Fox et al., 2006). This software takes a query of two search terms and triggers searches for both search terms. In the resulting set of documents, the software tries to find and identify the documents which include both search terms and keeps these topics as stepping stones between the two search terms. The user can then select one of the topics and further browse into the documents that deal with the respective topic and contain both search terms. Although this approach incorporates the idea of paths leading to documents, it has a major difference to the approaches of Bush and Bates: It relies on precomputed literature databases, making it only suitable for queries within that specified database. This implies the assumption that there exists one document in the database which fulfills the user's information need. This assumption contrasts earlier approaches presented before, as they do not expect one document to fulfill the user's information need, but admit that an overview of a topic can be achieved by a set of meaningful documents. (Fox et al., 2006) present two other approaches for finding text documents in a preprocessed database: Clustering resulting documents according to a carrot²-clustering algorithm[18], and visualizing search results based on the ACM classification. The authors state that their 'experiments have shown that visualization is a key ingredient in many of the recipes for exploration' (Fox et al., 2006, p.58).

The aforementioned paper is exemplary for a bunch of publications of the years 2000-2010, in which authors try to ease satisfying information needs by providing more convenient access to single databases. This results in several approaches where databases are preprocessed and equipped with new functions for accessing their content. These approaches contain the described ways of clustering or organizing results by applying classifications. Many of the approaches try to ease access to information by providing faceted access to databases, such as the Open Video Digital Library (Marchionini, 2006), the Flamenco faceted search engine (Hearst, 2006), or the mSpace browser for classical music (schraefel et al., 2006). (Kules & Capra, 2008) describe a set of criteria for developing use cases for exploratory search within the faceted Online Public Access Catalog (OPAC) of the North Carolina State University (NCSU). I use these criteria for developing complex search tasks for the evaluation of my system.

(Assfalg, 1999) presents some thoughts on forming an overview of a topic from a set of different resources. He introduces the concept of 'fluid information'. 'Fluid information' according to this concept describes heterogeneously cross-linked information (Assfalg, 1999, p.182). The author claims from his view of 1999 that in the future, documents or reports will be consisting more and more of thematically ordered and heterogeneous information resources. These will be interconnected with each other and show a multimedia character, rather than being monographic compounds of information.

My literature research of the historical origins of building search trails shows that the concept of trails through arbitrary web resources has not been realized since. It seems that after the initial ideas by (Bush, 1945) and (Bates, 1989) – dating from times before the Internet became publicly available – the trend rather went to facilitating access to specific databases (like the approaches described above) than to easing interaction with the Web itself. Research approaches focusing on easing access to databases can be found until the mid-2000's. More general approaches that focused their research on investigating the interaction of users with the Web appeared in the mid-2000's and are presented in more detail in the subchapter on search logging systems (Chapter 2.5.1). Approaches focusing on collaboration during search processes emerged even later; I introduce them in Chapter 2.5.2.

[18] Retrieved April 16, 2015 from http://search.carrot2.org/stable/search

2.2 Concepts, definitions, and models

A commonality of all the described tools and scenarios around search trails and related concepts from the previous subchapter is that they are all built around non-trivial search tasks. These search tasks are often driven by the interest of the searcher, and are characterized by unspecific search needs that evolve as the searcher learns about a search topic. In the following sections of this subchapter, I further investigate the history of classifications of such search tasks and present approaches for formalizing search tasks. The following sections present work on the origins of the concepts of exploratory and complex search and provide some definitions which are needed throughout the thesis. Afterwards, I present different approaches of search task classifications and some models of individual and collaborative search. A section on the value of negative search results concludes this subchapter.

2.2.1 Origins of exploratory search and complex search

In Chapter 2.1, I give an introduction into historical origins of the idea of creating search trails. As I want to use the idea of search trails for supporting complex search tasks, I investigate the origins and development of the terms exploratory search and complex search in this section.

Going back in the scientific literature, it turns out that the term ‘exploratory search’ is used already in the 1970’s. Back then, it is mainly used for describing a process in which the parameters of a calculation are altered by a fixed delta for each new calculation. For example, (Kramer & Sandor, 1975) in their paper on optimization for planar mechanism use the term exploratory search as well as (Altiok & Stidham, 1983) in their publication on the allocation of buffer capacities. Similarly, this term is used in (Hendrickson et al., 1988), where exploratory search means incrementing parameter values in calculations.

In her paper on berrypicking dating back to the year 1989, Marcia Bates does not only compare the process of search with the picking of berries in a forest, she also mentions the term ‘exploratory search’ for one of the first times in the context of information search (Bates, 1989, p.411). Bates mentions this term when referring to (Kuhlthau, 1988), where an experiment was conducted in which students performed library searches on scientific topics. Bates states that ‘there is a great deal of exploratory searching going on before and after a topic for a paper is selected’ (Bates, 1989, p.411). This approach reminds of the rather mathematic approaches from above, in which slight alterations of parameters broaden the result set of a calculation. Bates’ research already deals with unspecific information needs and covers a type of search that can be characterized by the term ‘exploratory’. However, no efforts have been made to formalize this concept. It took until 2006 when first definitions of exploratory search were formulated, and it took until 2012 when the term ‘complex search’ was initially defined (cf. Chapter 2.2.2).

In the time between the early 1990’s and the late 2000’s, some research has been conducted on large information collections or unspecific information needs. Early approaches mainly dealt with library search, such as (Breivik, 1985) or (Kuhlthau, 1988). In 1991, (Kuhlthau, 1991) developed a model of search. I present this model in Chapter 2.2.4. When motivating this model, Kuhlthau describes the discrepancy between the very organized systems for information retrieval and the user’s unstructured need for information: ‘Users’ problems are characterized by uncertainty and confusion' (Kuhlthau, 1991, p.361). Similarly, she describes a phase in the information seeking process in which there exists 'an inability to express precisely what information is needed' (Kuhlthau, 1991, p.366).

In 1994, PhD student Mark Nelson described current problems with information overload from his perspective. He states that information on the Internet is posted in a number of different formats and located at thousands of sites, which makes finding the relevant pieces of information hard. In order to find 'personally meaningful information', users have to 'take on the responsibility of designing and structuring the parameters of their searches to match their own points of view' (Nelson, 1994, p.12). This approach also describes a process of searching information by exploring a collection of resources with unspecific information needs. Nelson finds two main factors that determine the user's ability to access information. These factors are based both on the user and the system side. The first factor is the user's information literacy (cf. Chapter 1.2), which is 'the ability to effectively access and evaluate information for a given need' (Breivik, 1985, p.723). The second factor is the usability of the application used to access the resources containing information.

One of the first mentions of the term 'complex search' can be found in (Xu et al., 1999). In order to facilitate complex search tasks, an experiment was conducted by (Xu et al., 1999) to find out how to efficiently structure online resources. The authors identify complex documents, such as troubleshooting materials, training materials, or maintenance materials. These documents are hypertext documents which describe and model a work domain by interlinked pieces of text. The authors define complex documents as documents 'containing numerous interdependencies in a work domain' (Xu et al., 1999, p.206). The authors do not define the term 'complex search' explicitly, but use it for describing the difficulty of the search tasks. They let the participants solve increasingly difficult search tasks in their study: simple, complex, and problem-solving search tasks. The complexity of the tasks is implicitly defined by the number of nodes from which information needed to be synthesized to be able to give an answer to the given question. Simple tasks can be solved with information that is contained in only one node, and complex search tasks can only be solved by synthesizing information from several nodes. The problem-solving search tasks consist of several related complex search tasks. The authors develop two approaches for organizing information in a visual interface and evaluate with which type of visualization the users can react best to the given search tasks.

A similar, but more mathematical approach was followed in (McEneaney, 2001), who also assessed navigation in hypertext documents. For these purposes, a hypertext representation of a student advising handbook is investigated. The author constructs adjacency matrices and graph based visualizations from the hypertext document. The user's navigation through the hypertext documents gets captured and transformed into navigation graphs. The users' navigation graphs are mathematically evaluated with the help of graph metrics. The goal was finding out whether the searcher is using a rather linear approach of searching (like turning pages in a handbook), or followed a more strategic approach of searching, which connects resources based on topics. In this context, McEaneaney uses the term 'complex learning tasks' for activities like searching, learning, or reading in hypertext (McEneaney, 2001, p.781).

An overview of the first publications investigating the nature of exploratory search in more detail can be found in (White et al., 2006a). This paper is an opener article for a special issue of the 'Communications of the ACM' (CACM) on supporting exploratory search.

In his paper on temporal patterns of interactions from the CACM April 2006 special issue on supporting exploratory search, (Jansen, 2006) uses both the terms 'exploratory search' and 'complex search'. He describes 'more complex searching situations' by 'user uncertainty about the information needs, the content space, or the system's capabilities' (Jansen, 2006, p.72). This implies that web search engines are not able to support these kinds of searches, as they 'effectively support searching of short queries, with session durations of typically 15 minutes'. In his paper, he presents a way of supporting complex or exploratory search tasks in a closed document collection by providing assistance when one of his 26 detected search patterns is followed. The publication claims that 'in 70% of the cases, the searchers on the system with the pattern-based searching assistance performed better than searchers on the other system' (Jansen, 2006, p.73). While the pattern based system provided support for the searcher just in the critical points of a search process as defined in the patterns, another investigated system provides support at every point of the search process. The results show that too much support may turn out to disturb complex search processes. These findings get confirmed e.g. by the finding that tools for supporting collaborative search need to be lightweight first and foremost (Ringel Morris, 2013), which are further elaborated in Chapter 2.2.5.

(Gersh et al., 2006) present a vision for capturing the process of exploratory search, although not explaining how this should be done. The authors first present a model for so-called 'rich information collections', which can be explored to extract 'nuggets of information' that will later 'be organized into a formal report' (Gersh et al., 2006, p.64). They develop a software prototype that supports the access and visual display of the 'rich information collections'. The publication points out the value of the search process for the searcher. This implies that 'the purpose of exploratory search is insight, not data' (Gersh et al., 2006, p.68), meaning that data collection is not the primary goal of a long-running search process, but gaining insights. The user interface of the presented system strongly relies on semantic concepts and self-contained data collections to be explored. It is interesting that most of the studies presented so far used self-contained data collections, may it be databases or extensive hypertext data collections.

Besides the presented systems and approaches, the CACM issue contains a paper in which a definition of 'exploratory search' is given for the first time (Marchionini, 2006). Marchionini groups search activities into the three categories of 'Lookup', 'Learn', and 'Investigate', and defines exploratory search as to be pertinent to the 'Learn' and 'Investigate' categories. Further information on this characterization and a definition of exploratory search will be given in the following section (cf. Chapter 2.2.2) on definitions. Besides this first approach on formalizing the so far rather vague concept of exploratory search, Marchionini presents some more practical approaches on supporting exploratory search. He points out the value of menus and expandable hierarchical file structures over command based systems (Marchionini, 2006, p.43 f.). Providing relevance feedback is another way of adding value to results to help searchers, but searchers are often unwilling to do that extra step. He presents the Open Video Digital Library (a system which is now offline), which provided ways of searching the database of videos by making use of facets, filtering, and clustering. Marchionini's example is similar to (Gersh et al., 2006), as it is a self-contained data collection to perform exploratory searches on. Besides that, it is interesting that this publication is one of the few that cites Garfield's concept of 'negative search' ((Garfield, 1970), cf. Chapter 2.2.6) and puts it into relation with exploratory search. Nevertheless, some confusion on the nature of exploratory search still remains: 'Defining what constitutes an exploratory search is challenging. […] In exploratory search, users generally combine querying and browsing strategies to foster learning and investigation.' (White et al., 2006a, p.38).

Marchionini's definition of exploratory search was accepted well, as it was taken up relatively fast. (Pirolli, 2009) mentions it besides sensemaking and social search as one of three conceptual frameworks that enrich the scope of information seeking tasks (Pirolli, 2009, p.36). In his publication, Pirolli describes ten temporal powers of ten, in which search processes take place: from milliseconds (10^{-2}) to months (10^{7}). Table 2 gives an overview of the described temporal units and relates them to time units and different levels where search processes can happen. The last column relates the temporal units to search activities. This categorization therefore resembles how exploratory search (as well as complex search) covers extensive timespans.

Table 2: Mapping of time scales to search activities.

Scale (seconds)	Time unit	Level	Search activities
10^{7}	Months	Social	Social search, communication about search results.
10^{6}	Weeks		Exploratory search with long-term goals, composed of short-term goals.
10^{5}	Days		
10^{4}	Hours	Rational	
10^{3}	10 minutes		
10^{2}	Minutes		Information retrieval, checking whether a result fulfills the information need.
10^{1}	10 seconds	Cognitive / Psychological	
10^{0}	1 second		
10^{-1}	100 ms		Neuronal processing of information.
10^{-2}	10 ms	Biological	

Besides its impact in the literature on search support, the concept of exploratory search also had impact on the evaluation of search supporting systems. Several authors realized that systems have been developed, which are intended to facilitate exploratory search, even if they are not explicitly named 'exploratory search support systems'. One of the first publications in this context is (White et al., 2008), who describe systems for information visualization, document clustering, and browsing as 'exploratory search systems' (ESS). Interestingly, this publication claims to cite a definition of exploratory search from (Marchionini, 2006), which is actually not given there. I further elaborate on the given definition in Chapter 2.2.2.

Another publication talking up Marchionini's definition of exploratory search is (Aula & Russell, 2008). Besides mentioning exploratory search, Aula & Russell mention the term 'complex search'. Although Aula & Russell rely on the concept of exploratory search, they try to establish a more mathematical and more measurable way of looking at exploratory search, by avoiding the concepts of 'learning' and 'investigation'.

In order to more accurately define the difference between the two concepts, Aula and Russell propose a matrix in which every individual search task can be located with respect to its complexity. The matrix is defined by two dimensions, the 'Goal Abstraction Level (GAL)' and the number of 'Search Moves (SM)'. According to their definition, the GAL describes the degree of abstraction of the search goal and reaches from 'concrete and clear goals' to 'high degree of abstraction'. The second dimension (SM) is an estimation of the number of web search steps needed to achieve the search goal and serves as an indicator for the complexity of the search task. This dimension reaches from 'simple' to 'complex' (cf. Figure 4). Both dimensions together determine the overall complexity of the search task. A task may be complex when it needs a lot of search steps to gather the necessary information, but does not necessarily rely on the 'learn' and 'investigate' search tasks proposed by Marchionini. Nevertheless, tasks like 'Find the ten classical music pieces most used for film soundtracks' are complex, even if they do not require much learning. Similarly, a task like 'Learn something about Beethoven's music' does not require many web search steps to find some information, but is highly abstract and requires substantial learning. Aula & Russell introduce the concept of the 'Notetaking boundary', which marks a border after which the mental load for the searcher gets so high that taking notes becomes inevitable to cope with the amount of information.

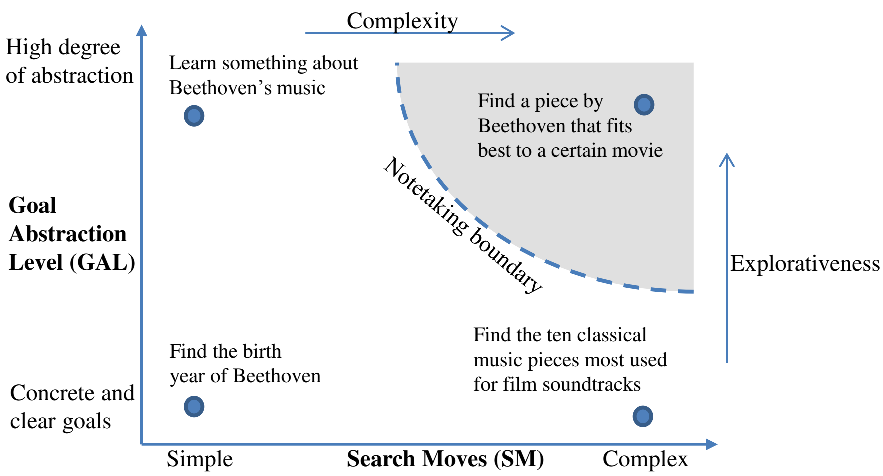


Figure 4: Schematic view of the concept of the notetaking boundary (Aula & Russell, 2008).

Despite the formal definition of exploratory search by (Marchionini, 2006) and (White et al., 2008) and one approach of defining complex search in (Aula & Russell, 2008), authors still relied on various definitions of complex search. (Villa et al., 2009) present an interface for supporting complex search tasks. In a section on complex search tasks, the authors mention ways of adding complexity to a task (e.g. by removing search specifics). For their experiments with search tasks in a database, the authors rely on a definition of general task complexity by the number of solutions and the number of possible ways to a solution, which is based on a general definition of task complexity by (Campbell, 1988). The authors present an 'aspectual search interface' that allows users to search separately within different aspects of a topic at the same time. This is done by a faceted interface that allows investigating up to three facets at the same time. It turns out that with the presented aspectual search interface, the users searched more and found more relevant results.

The first and only approach of defining complex search as a counterpart to exploratory search was made by (Singer et al., 2012a). The authors claim that exploratory search was mainly defined in (Marchionini, 2006) as 'encompassing all aspects of search not covered by current search [engines]' (Singer et al., 2012a, p.90) and that it therefore covers complex cognitive processes, such as comprehension, integration, analysis, or planning. In order to avoid the cognitively high loaded aspects of Marchionini's definition of exploratory search, Singer et al. define complex search as 'a multi-step and time-consuming process that requires multiple queries, scanning through many documents, and extracting and compiling information from multiple sources' (Singer et al., 2012a, p.90). This definition therefore relies on the three core elements of complex search, which are aggregation, discovery, and synthesis. I give more details on the definitions and their key elements in the Chapter 2.2.2. What is most important about the definition itself is that the definition of complex search is the first approach to free complex search tasks from the cognitive aspects from Marchionini and to focus more on the work load induced by complex search tasks.

It is interesting that the definition of complex search stands at the end of a long process of work, as the authors started working on exploratory search tasks until they found that this definition may not be suitable for their needs. Publications like (Singer et al., 2011) are therefore still dealing with the concept of exploratory search. Later publications clearly make a difference between both concepts, such as (Singer et al., 2012b).

This section presents an overview of multiple attempts of defining exploratory search and complex search in the literature of library science, computer science, and web search applications. It is worth noting that a lot of research has been done without a clear definition of the nature of search tasks, just by relying on a common sense of the complexity of a task. The next section presents the most important definitions related to complex search support and therefore provides common ground for all further research of my thesis.

2.2.2 Definitions

In order to avoid misunderstandings about terms that are frequently used in my thesis, it is necessary to create some common ground. In this section, I present the most important definitions of terms which are used throughout the thesis. Some definitions are taken from other sources, while I developed some more to fit into the context of search trails. Some basic terms that need to be defined are information need, precision, and recall.

- 'An **information need** is the perceived need for information that leads to someone using an information retrieval system in the first place.' (Shneiderman, 1997)
- '**Precision** is the ratio of the number of documents retrieved that 'should' have been retrieved – i.e., the number of retrieved documents that were really relevant to the query – to the total number of documents retrieved.' (Shneiderman, 1997)
- '**Recall** is the ratio of the number of relevant documents retrieved to the number of relevant documents in the database.' (Shneiderman, 1997)

While the term ‘information need’ is important as a trigger for inducing search behavior in general, the terms of precision and recall are used in the context of information retrieval. Precision means the share of relevant results in the result set, while recall means the share of relevant documents retrieved from the database.

A very important definition for this thesis is the one for exploratory search, as the concept of complex search stems from it. The most definition for exploratory search cited most is the following:

> *‘Exploratory search can be used to describe an information-seeking problem context that is open-ended, persistent, and multi-faceted; and to describe information seeking processes that are opportunistic, iterative, and multi-tactical. In the first sense, exploratory search is commonly used in scientific discovery, learning, and decision-making contexts. In the second sense, exploratory tactics are used in all manner of information seeking and reflect user preferences as much as the goal.’*

(White & Roth, 2009) and (White et al., 2008)

This definition is two-fold. The first part of the definition defines a much narrower search case than the second part. The first part of the definition covers information needs that are well-defined, persistent, but open ended in nature, and can be illustrated by scientific discovery. The second part of the definition applies to many more search cases. These may be randomly induced, resulting in users with unclear information need, and trigger an iterative process of searching, in which the goals are adjusted to the searcher’s needs. There has been some confusion with this definition, as it is cited by both (White & Roth, 2009) and (White et al., 2008) who claim that it is mentioned in (Marchionini, 2006), but in fact, this actual formulation does not come from this source. As Marchionini was the editor of (White & Roth, 2009), we can assume that he agrees to the definition above.

In his original publication (Marchionini, 2006) does not give a clear definition of exploratory search. He identifies three kinds of search activities, which he calls ‘Lookup’, ‘Learn’, and ‘Investigate’ (cf. the upper part of Figure 5) and mentions example search activities for them, e.g. verification, knowledge acquisition, or analysis. These tasks may have certain overlaps. Marchionini then defines exploratory search as consisting of the ‘Learn’ and ‘Investigate’ activities. A further evaluation of the exemplary tasks for each of the two activities reveals that Marchionini collects basically all search related activities under ‘Learn’ or ‘Investigate’, and therefore collects many cognitively highly demanding aspects under ‘Exploratory search’. In his definition, Marchionini mixes search activities that are based on raw information (so-called ‘first-level search activities’ like aggregation, discovery, or synthesis, bold in Figure 5) with activities that are based on processed information (so-called ‘second-level search activities’ like comparison, evaluation, or analysis) (Singer, 2012).

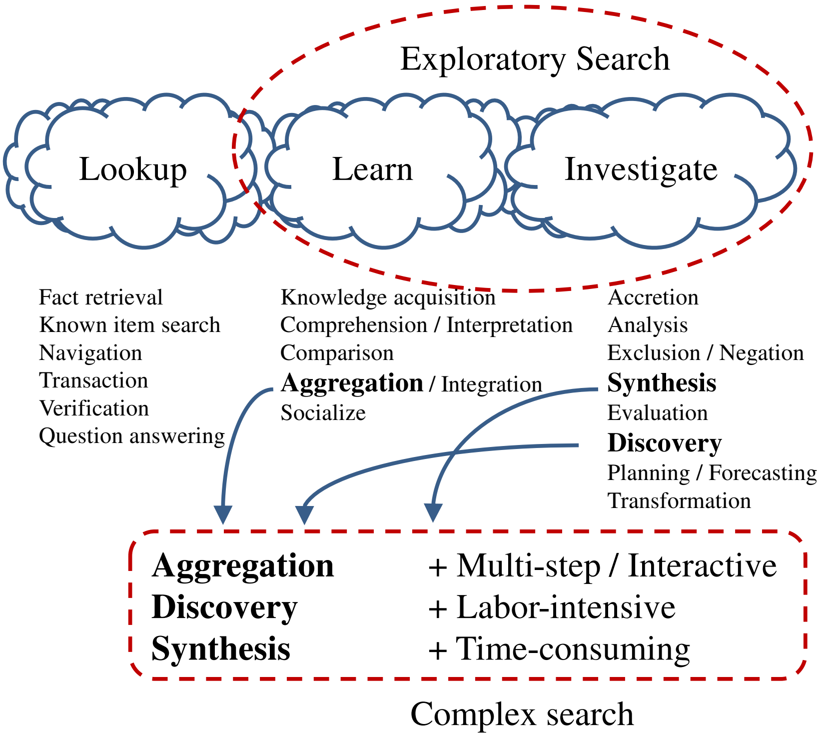


Figure 5: Connection between the definitions of exploratory and complex search (cf. (Marchionini, 2006; Singer, 2012, p.19 f.)).

This mixture of first and second level search activities implies that the original definition of exploratory search is not fully suitable for the needs of the thesis, as the envisioned use cases do not necessarily need second-level search activities for turning out to be complex. Therefore, (Singer et al., 2012a) develop the concept of ‘complex search’. The term ‘complex search’ does not exist for long. This does not mean that there were no complex search tasks before; it only says that the most suitable definition to be used in my thesis dates from that point in time. In their publication, (Singer et al., 2012a) define complex search as ‘a multi-step and time-consuming process that requires multiple queries, scanning through many documents, and extracting and compiling information from multiple sources’ (Singer et al., 2012a, p.90).

The aforementioned definition is refined in (Singer, 2012). For my thesis, I use Singer’s definition, as it explicitly highlights the importance of the steps of aggregation, discovery, and synthesis. According to this source, complex search tasks are defined as follows:

> *‘Complex search tasks are tasks where users are required to follow a multi-step and time consuming process that is not answerable with one query, requiring synthesized information from more than one retrieved web page or document to be solved. The process to work off complex search tasks usually comprises at least one of the process steps aggregation, discovery and synthesis.’*

(Singer, 2012, p.17)

According to Singer’s definition, complex search is based on the three first-level search activities of aggregation, discovery, and synthesis from Marchionini’s definition, and adds a multi-step or interactive process to it, which is labor-intensive and time time-consuming (cf. the lower part of Figure 5).

Singer defines **aggregation** as a part of complex search as ‘the support for selecting, storing, and accessing the relevant documents for a certain aspect of a search need’ (Singer et al., 2012a, p.91). He defines **discovery** as ‘the support for finding new, relevant documents and at the same time extending the searcher’s view of the topic towards previously unknown, related categories and concepts’ (Singer et al., 2012a, p.91). **Synthesis** is defined as ‘the support for compiling multiple documents into one and extracting relevant information – this means also just parts – from the already found or selected documents’ (Singer et al., 2012a, p.91).

In contrast to the definition of exploratory search, the definition of complex search avoids the cognitive concepts from Marchionini’s definition and focuses more on the work effort that needs to be invested into a complex search task to fulfil the information need. As Singer’s definition concentrates on first-level search activities, tool support for complex search tasks is more easily achievable and measurable than when dealing with all the cognitive concepts from Marchionini’s definition. A search task like checking the price and availability of a desired good or product on a number of portals induces lots of first-level search activities and is not a very demanding task with respect to learning. However, it is a multi-step process, which is labor-intensive and time-consuming. This qualifies the task as complex rather than exploratory.

As the exploratory or complex search tasks may span over a certain amount of time, these processes may be divided into sessions. A session is therefore not congruent with a search process, as one session may contain searches on different tasks, or be just a part of one search task.

With respect to the desired support for discontinuous, asynchronous, collaborative, complex search tasks that are introduced in the later chapters of the thesis and the analyses made in the user studies, I need to introduce some more definitions. The definitions regarding the characteristics of a search are as follows:

- The **searcher** is the person conducting a search process.
- A **search task** is ‘a sequence of activities with the goal of finding specified information - the specification may range from narrow and detailed, e.g., a fact, to broad and vague, e.g. something about memory problems in old age.’ (Ingwersen & Järvelin, 2005, p.73). A search task induces an information need and a search process.
- The **search process** is the sequence of actions related to the acquisition of information to reach an information goal. The search process is also referred to as the ‘search’.
- A **session** is a sequence of queries and viewed documents which are performed by the user in a certain timespan. The end of a session can be expressed by the user interrupting or ending the process explicitly, or a certain amount of time passing without user interaction (Lewandowski, 2015, p.72 ff.).
- The **search result** is the sum of all artifacts that are created during the search process.
- **Search support** is the sum of all tools and methods selected by the searcher to facilitate the search process.
- The **search trail** is the logical representation of the searcher’s actions when traversing the Internet during a search process.

The search trail **S** is formalized as a set

$$S = \{U, T, I, N, R\}$$

where

U is the username,
T is the title of the search trail,
I is the identification code of the search trail,
N is a set containing the information about all nodes of the search trail, and
R is a tuple containing the sequence of events leading to the search trail.

The information about a node **N** is formalized as a set of tuples

$$N = \{(u, ti, Hi, Su, V, h): URL \times Title \times\ Highlights \times Successors \times Visits \times Hostname\}$$

where each tuple consist of the following entries:

u is the URL of the node,
ti is the title of the node,
$Hi = (h_1, \dots, h_n\ | h_i\ \in Highlights)$ is the tuple of highlights as defined by the user,
$Su = \{n_1, \dots, n_n\ | n_i \in N\}$ is the set of the node's successor nodes,
$V = (v_1, \dots, v_n | v_i\ \in Timestamps)$ is the tuple of timestamps of node visits, and
h is the node's hostname.

The sequence of events **R** is formalized as the tuple of sets

$$R = (\{n, t, e\} : N\ \times Timestamps\ \times Event\text{–}Identifier)$$

where each set consists of the following entries:

$n\ \in N$ is a node,
t is the timestamp specifying the time the event happened, and
e is the identifier of the event occurred at time t.

This structure helps gathering and storing the data necessary for building the search trail graph representation during the search process (cf. Chapter 4.2.3).

- The **search graph** is the visualization of the search trail as a force-directed graph.

During the evaluations, not only the search results are of importance, but also the quality of certain artifacts. I therefore define the quality of search-related concepts as follows:

- The **quality of the search process** is the intrinsic quality of the process undertaken by the searcher. It is judged by the searcher. The quality of the search process can be measured by asking the searcher whether the search process was perceived successful, if it was liked, or if it was motivating for the searcher.
- The **quality of the search results** is the quality of all artifacts being created during the search process. This quality can be judged by external experts. It may for example include the depth and breadth of the found information, the number of visited pages, and the diversity of the gathered information.
- The **quality of the search support** is the quality of all tools and method selected by the searcher to facilitate the search process. It is a subjective measure of support during the

search process. This quality can be judged by the searcher, for example by judging the user experience with the selected search support tools.

The definitions in this section help building common ground for the following sections. Especially the definitions of the characteristics of search and their qualities are taken up in the chapter on the evaluation of the search support tool I develop in my thesis.

2.2.3 Search task classification

First thoughts on classifying search tasks were presented by (Shneiderman et al., 1997) in January 1997. The authors identify the problem that textual search interfaces are 'often needlessly complex', and complain that 'zero-hit outcomes occur on 30% of searches at some services', or that 'huge numbers of hits distract users'. The authors note that search engines of these days are widely used (e.g. Infoseek, AltaVista, or Lycos) although they are not optimal. To better support textual search, they propose a four-phase framework for search, consisting of the following four phases (Shneiderman et al., 1997):

1. **Formulation** (Expressing the search)
   This phase includes the four subtasks of defining the sources where to search, which fields or attributes of a document to search, what text to search for, and deciding which variants of the desired text to accept.
2. **Action** (Launching the search)
   In this phase, the users take the action and enter the search terms into a search interface.
3. **Results** (Reading messages and outcomes)
   During this phase, the results gathered get reviewed by the users, who have the possibility to customize the results by sorting and filtering.
4. **Refinement** (Formulating the next step)
   The users may now refine the query by adding words or asking for more information of a certain type.

It is remarkable that this very early model of search already presents a cyclic approach of searching, and therefore overcomes the assumption of information retrieval that there is one document that fulfills the searcher's information need completely. Shneiderman et al. propose that search interfaces should follow the eight rules for user interface design (Shneiderman, 1992) and guide the users through all four phases of a search. A search interface should allow the users to explicitly state parameters for all phases.

A second publication by (Shneiderman, 1997) cites (Shneiderman et al., 1997) and presents thoughts on the design of information-abundant websites, together with issues and recommendations. This publication was meant to be a trigger for further research, more concerned with raising questions than with providing answers. Shneiderman complains about an 'ocean of information', just like generations of intellectuals before. Portals or lifeboats to this ocean can be thoroughly designed websites, for which he presents design recommendations. Similar to (Fox et al., 2006), he claims that 'users consistently praised screens that provided overviews of large information spaces' (Shneiderman, 1997, p.5). He identifies the user's tasks to base design recommendations for websites on the tasks that users are about to perform on them. His classification includes 4 types of tasks (Shneiderman, 1997, p.11 ff.):

- **Specific fact-finding** (Known item search)
  Example: 'Find the Library of Congress call number of 'Future Shock'!'
- **Extended fact-finding**
  Example: 'What other books are there by the author of 'Jurassic Park'?'
- **Open-ended browsing**
  Example: 'Is there work on voice recognition in Japan?'
- **Exploration of availability**
  Example: 'What genealogy information is at the National Archives?'

The example queries show how the possible search tasks range from very specific to very unspecific information needs. All these different information needs should be supported by the presented four-phase framework for search. The author presents some screenshots of prototypes that allow for very narrow and very broad searches, such as a restaurant finder that allows specification of food style, price, opening times, payment options, or a specific standard of a restaurant.

It is worth pointing out that these publications have the perspective of 1997. The presented approaches assume that a low number (ideally one) of perfectly fitting search results are considered optimal. These approaches are therefore closer to information retrieval than to complex search. The papers present ways to guide users to the one and optimal search query, for example by specifying as many search parameters as necessary to achieve this. From today's perspective, the described approaches seem awkward, as the amount of indexed information on the Internet cannot be limited down to the one perfect search result. The presented approaches are in some cases strongly related to faceted approaches, where a meaningful selection of facets narrows down the search results to only the desired results.

Shneiderman's publication (Shneiderman et al., 1997) is cited by (Broder, 2002). Broder presents his taxonomy of web search that alters the classification in (Shneiderman et al., 1997). The classification of four different search tasks is broken down by Broder to form three different classes of search tasks (Broder, 2002, p.5):

1. **Navigational** search tasks
2. **Informational** search tasks
3. **Transactional** search tasks

During navigational search tasks, the users' intent is reaching a particular website, whereas during transactional search tasks, the user's intent is performing some web-mediated activity (e.g. shopping on a page or finding a database to download certain files). The class of informational search tasks comes closest to the topic of complex search, as the intent is finding information on a certain topic, which implies 'that many informational queries are extremely wide' (Broder, 2002, p.6).

Broder is also the first author to do a statistical analysis of search tasks to find out to what percentage queries of which type are performed. He uses a query log from AltaVista for that and evaluates and classifies the search terms to match one of the three categories. He shows that approximately 20% of the queries show a navigational character. Approximately 30% of the queries show a transactional character, and approximately 50% show an informational character. He conducted a supplementary user survey with the users of AltaVista, which returns that ‘in almost 15% of all searches the desired target is a good collection of links on the subject, rather than a good document’ (Broder, 2002, p.6). This user survey confirms the results from the query analysis.

(Rose & Levinson, 2004) continue Broder’s work, as they developed a framework for search goals in 2004. They develop a classification that comes close to Broder’s classification on the top-level categories, but adds additional layers to it. The authors manually classify approximately 1,500 English search queries with their classification. Unlike Broder, Rose and Levinson introduce subcategories which specify the search goals further. The classification is the following:

1. **Navigational**
   Go to a specific website
2. **Informational**
   Learn something by reading or viewing web pages
   - **2.1. Directed**: Learn something in particular about my topic
     - **2.1.1. Closed**: Get an answer to a question with an unambiguous answer
     - **2.1.2. Open**: Get an answer to an open-ended question
   - **2.2. Undirected**: Learn everything about a topic
   - **2.3. Advice**: Get advice, ideas, suggestions, or instructions
   - **2.4. Locate**: Find out where a real-world good can be obtained
   - **2.5. List**: Get a list of suggested websites on a topic
3. **Resource**
   Obtain a resource which is not information
   - **3.1. Download**: Download a resource that must be on a computer to be useful
   - **3.2. Entertainment**: Be entertained by viewing items on a page
   - **3.3. Interact**: Interact with an online resource
   - **3.4. Obtain**: Obtain a resource that does not require a computer

In its top-level categories, this classification is congruent with Broder’s classification, except for the renaming of ‘Transactional’ to ‘Resource’. The authors evaluate three sets of approximately 500 queries to a search engine each from three different days. They find the distribution of top-level queries as shown in Figure 6 (Rose & Levinson, Study 1 to Study 3). A more detailed breakdown of the results shows that the undirected search tasks (Classification item 2.2) account for a major share among the informational search tasks. The other large share is the ‘Locate’ search tasks (Classification item 2.4), which together with the ‘Undirected’ search tasks account for roughly 50% of all search queries.

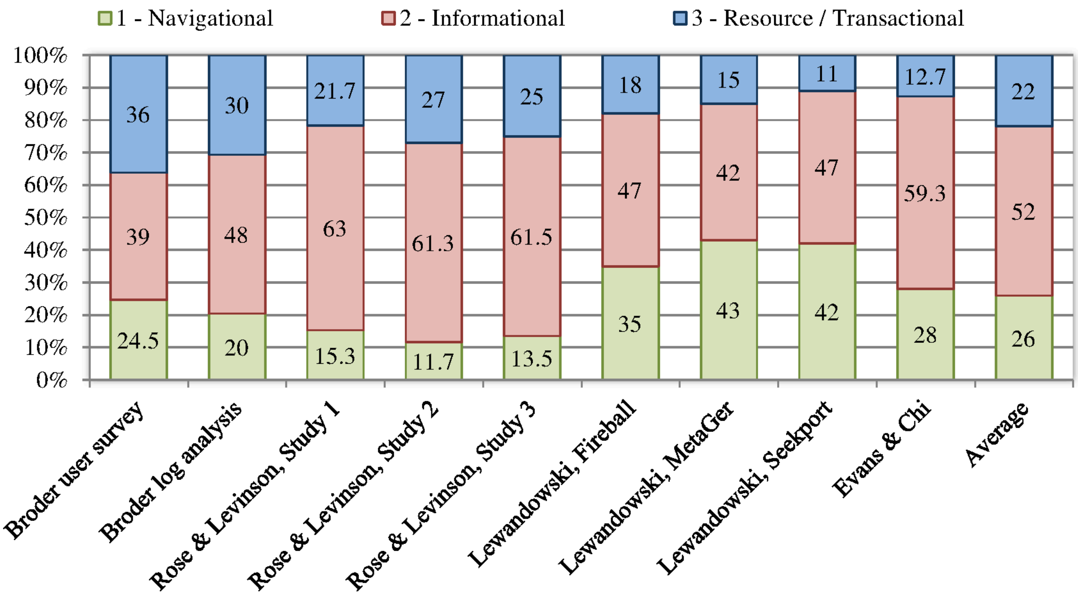


Figure 6: Overview of search task classification results for several studies.

With this analysis, Broder's results get confirmed and the authors state that although the share of navigational searches is relatively small, search engines seem to be constructed to support these ones best, while support for informational searches is missing.

Similar analyses were performed by (Lewandowski, 2006), who refers to both Broder and Rose and Levinson in his paper. With a team, he classified approximately 1,500 search queries from three German search engines (Fireball[19], MetaGer[20], and Seekport[21]). The classification is based on Broder's scheme and returned similar results like the two previous studies: The share of informational search queries is largest, followed by navigational queries.

The columns in Figure 6 show a comparison of Broder's, Rose & Levinson's, Lewandowski's, and Evans & Chi's (cf. below) results. Be aware that Broder's results do not sum up to 100%, which is due to the given numbers in the original publication. Transactional search queries account for the smallest share of search queries. Calculating the average of all studies, informational search queries account for 26%, transactional queries for 23%, and informational search queries for 51% of all queries. As the primary focus of search engines is fact-finding, most search queries therefore cannot be well supported by search engines.

While the previous approaches tried to conceptualize the search behavior of a single user, some researchers started with analyzing the social search behavior of user groups. (Evans & Chi, 2008a) and (Evans & Chi, 2008b) present a model of social search, in which they identify the phases of a search process in which social interaction before, during, and after a search process may happen. The authors present a study in which they evaluate 150 questionnaires and ask their participants about their latest search acts. The authors classify the answers based on

[19] Retrieved April 15, 2015 from http://www.fireball.de/indexjs.asp
[20] Retrieved April 15, 2015 from https://www.metager.de/
[21] Retrieved April 15, 2015 from http://www.seekport.com/

Broder's classification and found that 12.7% of the queries are transactional, 28.0% are navigational, and 59.3% are informational, which confirms the results from above.

In a study from 2008, (Lewandowski, 2008) investigates the retrieval effectiveness of search engines with informational queries and finds that Google and Yahoo perform best among the investigated search engines, which also included MSN, Ask, and Seekport. Although the results are good for Google and Yahoo, they are not optimal, as more than 50% of the delivered results are not relevant to the search query (Lewandowski, 2008, p.11). A somehow related approach of classifying tasks in general was done by (Li, 2004), who presents a faceted classification of tasks in general (not only Internet search tasks). She proposes eight top-level facets and numerous sub-facets and values for each of the facets. A classification like this could also be mapped to web search, but my related work shows that a more simple approach like the one from Broder got used more widely.

For my thesis, the concept of informational search tasks from the classification by (Broder, 2002) is most important, as I want to support users coping with this sort of unspecific information needs. All further studies could support that this classification is valid and could overall confirm its results.

2.2.4 Models of search processes

The previous sections of this chapter present an historical overview of the idea of search trails, some historical insights on the development of exploratory and complex search tasks, the according definitions, and an overview of search task classifications. The approaches consider the search task as an entity which can be defined and classified. In fact, the process going on during a search is much more complex. Several authors have taken efforts in developing models breaking down search processes into single activities. This section gives a chronological overview of these models.

Before the Internet was used for retrieving information, libraries were a primary source of information. Theories and models of information retrieval (IR) exist for long, and a frequently used model of information retrieval is shown in Figure 7. It originates from a publication of (Bookstein & Cooper, 1976), who presented a mathematical model for information retrieval systems. It was cited by (Robertson, 1977), who rephrased the model without the mathematical background (Robertson, 1977, p.129). This version of the model was illustrated by (Bates, 1989), and was taken up by (Pirolli, 2009), among others. The model describes the classical process of information retrieval. A database of documents (e.g. a library) is processed into a document representation (e.g. by indexing), which is tried to be matched with an information need represented by a query. The key assumption is that one document is assumed to fulfill the information need and matches the query. The main work of the matching algorithm is to produce the most accurate results for the given query (Bates, 1989, p.409).

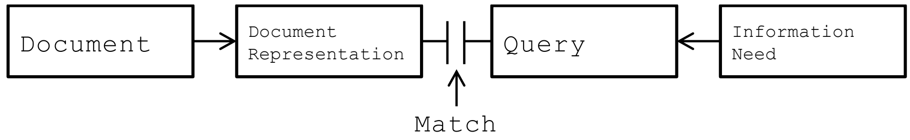


Figure 7: Original model of information retrieval (Bates, 1989).

This model may hold for application cases in libraries, where one book offers enough information to fulfill the information need completely, but the book is hard to find. (Bates, 1989) already mentions that in real-life, this key assumption may not be valid, and one single resource may not be sufficient: 'In other words, the query is satisfied not by a single final retrieved set, but by a series of selections of individual references and bits of information at each stage of the ever-modifying search.' (Bates, 1989, p.410). Based on these findings, Bates develops her model of berrypicking, in which pieces of information are picked from various documents retrieved by various queries, each gradually fulfilling the information need (cf. Figure 8). In this model, Q0 means an initial query with its variations Q1 to Qn, while T stands for thoughts and E for exit. The document symbols represent information sources.

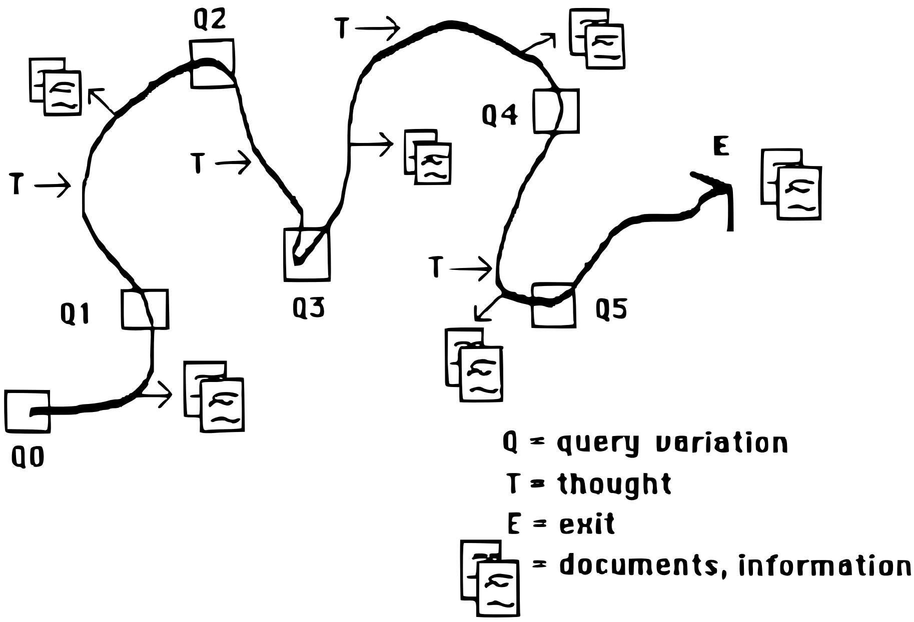


Figure 8: Model of berrypicking search (Bates, 1989).

In her model, Bates describes for the first time the sequence of searcher actions during the course of a search task. The line stands for 'the continuity of a single human being moving through many actions toward a general goal of a satisfactory completion of research related to an information need' (Bates, 1989, p.410). The line depicts the changes of the query based on the additional information and shows an evolutionary process influencing the information need as the searcher's knowledge grows. In a way, Bates builds upon the classic model of information retrieval, but connects many instances of the model to form an evolving process.

A similar view on search processes as a multi-step processes is presented in (Kuhlthau, 1991). She developed a model of the 'Information Search Process (ISP)', in which she splits information search processes into six phases (Kuhlthau, 1991, p.366 ff.). Most of the phases are confronted with the feelings, thoughts, and actions that are common to each phase of the information search process. It is worth noting that phases like exploration, formulation, or collection do not necessarily need to happen linearly, but the searchers can connect stages as they want. This may result in iterative search processes, with constant refinement of the user's information needs. Nevertheless, most searchers step from early stages to later stages as the search progresses. Table 3 shows an overview of the stages of the process and the corresponding feelings, thoughts, and actions.

Table 3: Six phases of the Information Search Process (ISP) (Kuhlthau, 1991).

	Stage	Explanation	Feelings	Thoughts	Actions
1	Initiation	A person becomes aware of a lack of knowledge or understanding.	Uncertainty	General / Vague	Seeking background information
2	Selection	Identify and select the general topic to be investigated.	Optimism		
3	Exploration	Investigate information on the general topic to extend personal understanding.	Confusion / Frustration / Doubt		Seeking relevant information
4	Formulation	Identify and select ideas within the information gathered, from which a focused perspective of the topic is formed.	Clarity	Narrowed / Clearer	
5	Collection	Gather information related to the focused topic.	Sense of Direction / Confidence	Increased interest	Seeking relevant or focused information
6	Presentation	Complete the search and prepare to present or otherwise use the findings.	Relief / Satisfaction *or* Disappointment	Clearer or Focused	

In contrast to Bates' model, Kuhlthau's model incorporates the searcher's internal and mental processes during the search process. These feelings and thoughts are what make the searcher act in a certain way. It is also interesting to point out how the feelings move from uncertainty to confidence, or how the thoughts develop from vague to focused. More recent publications on exploratory or complex search use the term of vague information needs ((Pirolli, 2009), (Singer et al., 2011), (Jansen, 2006), (Singla et al., 2010)) to describe the initial complexity of an information seeking process.

A similar textual model was provided by (Shneiderman, 1997, p.20 ff.), who presents a four phase framework for search in World Wide Web textual libraries (cf. Table 4 and Chapter 2.2.3). Shneiderman presents no theories on how the model was developed; he just argues that something needs to be done to provide common ground for both designers and developers.

Table 4: Four phase framework for search (Shneiderman, 1997).

	Phase	Explanation	Details
1	Formulation	Expressing the search	Define sources of information, phrases to enter and variants to try.
2	Action	Launching the search	Explicitly click on a button 'Search' or change search parameters.
3	Review of results	Reading messages and outcomes	Review outputs, control the size of the result set, or change the sequence of the results.
4	Refinement	Formulating the next step	Refine the query, save or exchange search settings.

His model is meant to guide designers of interactive systems through the search process and to provide support for each of the phases by designing for each phase. Shneiderman's model seems to intersect with Kuhlthau's model but his view is more from a software development perspective. He intends designers to come up with solutions to support each of the phases or actions, while Kuhlthau's view is more human-centered and holistic and covers the whole search process from the vague information need to a presentation of results.

A behavioral model of information seeking on the Web was presented by (Choo et al., 1998). This model is comparable to Kuhlthaus' and Shneidermans' approaches, as it also relies on a set of phases that are worked through during web search processes. (Choo et al., 1998) base their work on six characteristics of information retrieval system design and evaluation, which were presented in (Ellis, 1989, p.178 ff.) and taken up in (Ellis & Haugan, 1997, p.385). The six characteristics resemble a model of search and can be explained as in Table 5 (based on (Choo et al., 1998)):

Table 5: Six phases during web search processes (Choo et al., 1998).

	Phase of information seeking	Content
1	Starting	Starting comprises the activities that form the initial search for information, e.g. by identifying interesting sources. This leads to multiple resources being found.
2	Chaining	As the found resources cannot be evaluated all at once, they need to be chained. Forward chaining means the identification of resources leading to a found resource, while backward chaining means the identification leading from a visited resource to others.
3	Browsing	Within this amount of resources, browsing takes place as semi-directed search. Users may use lists or tables of content to make sense of the found resources.
4	Differentiating	During differentiating, the searcher evaluates the quality of the found information and prioritizes the value of the found resources.
5	Monitoring	Monitoring means the regular checking of core resources to stay updated about a certain topic.
6	Extracting	Extracting covers the identification of valuable information and tends to be a time-consuming process, as the original resources need to be worked through.

(Choo et al., 1998) map the six characteristics to four 'modes of scanning', which are in fact types of search, which differ in their effort and the number of sources being evaluated. The first mode is undirected viewing, which is serendipitous browsing, covering many resources with a low effort. The second mode is conditioned viewing, which contains more elements of learning. In this mode, information is extracted from a limited number of sources with a slightly higher effort than in the previous mode. The third mode is informal search, where the search is focused on a topic, which implies generating queries and investigating a small number of sources with medium effort. The fourth mode is formal search, in which information is systematically gathered with high effort from a high number of sources and from specific targets. The model proposed by (Choo et al., 1998) combines the six categories from (Ellis, 1989) with their four models of scanning. The X's identify where these categories match. The model can be seen in Table 6.

Table 6: Combination of the models by (Choo et al., 1998) and (Ellis, 1989). The X's mark the overlaps between the different categories.

	Starting	**Chaining**	**Browsing**	**Differentiating**	**Monitoring**	**Extracting**
Undirected viewing	X	X				
Conditioned viewing			X	X	X	
Informal search				X	X	X
Formal search					X	X

The model illustrates how a search process starts with rather unspecific queries in the first mode 'undirected viewing'. It then moves towards high quality, high effort search activities, like monitoring and extracting during 'formal search'. Both the formality of the search process and the mental efforts needed for gathering the desired information increase simultaneously. Nevertheless, the model still implies a sort of waterfall approach, leaving out the cyclic nature of search processes.

A more detailed view on the cyclic nature of search processes is provided by (Pirolli & Card, 2005), who investigate the process of search and provided a model for it. The model divides the search process into a foraging loop, which acquires information, and a sensemaking loop, which organizes and evaluates the acquired information. The model is closely related to exploratory and complex search processes as it models how a searcher acts to make sense of an amount of information. Figure 9 depicts the model.

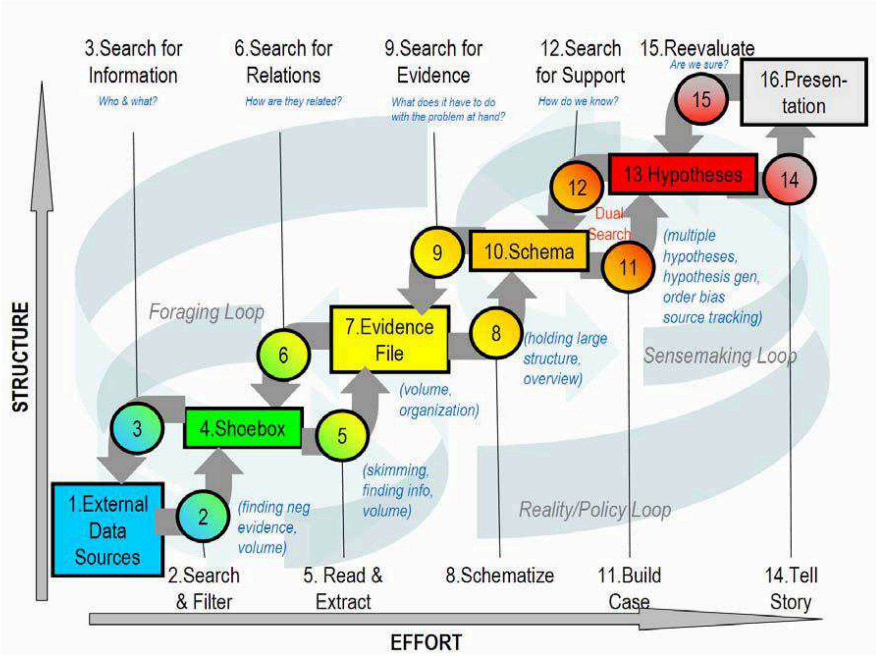


Figure 9: Cyclic model of search processes (Pirolli & Card, 2005).

The model is embedded into a two-dimensional space, being spanned by the dimensions of structure and effort. It consists of the two loops of foraging and sensemaking, which are both enclosed in the reality loop. While the foraging loop is located in the area of low structure and low effort, the sensemaking loop is located in the area of high structure and high effort. The model describes how the foraging loop delivers information from external sources, which gets processed by the searcher. It is evaluated in the sensemaking loop to be ready for presentation at the end of the search process. The foraging loop starts with searching for information from external sources and collects this information into a 'shoebox', from which details are incorporated into an 'evidence file'. It is worth noting that this process may be bidirectional, as the progress of collecting information in the evidence file may again induce further investigation of external data. The connecting element between the foraging and the sensemaking loop is the evidence file, which may be further structured into a schema, from which hypotheses can be deduced. These hypotheses can again induce further iterations of the sensemaking loop and trigger further evaluation of the evidence file. The final results may be presented, as they are condensed and structured enough to be of value for other people.

Pirolli & Card's model is in some points similar to Kuhlthau's model, as it contains steps like 'Search & Filter', which can be matched to Kuhlthau's stage of 'selection', 'Read & Extract' can be matched to 'Exploration', and 'Schematize' can be matched to 'Formulation'. Pirolli & Card clearly point out the cyclic nature of the process of search and embed it into a space of abstraction and necessary effort.

Pirolli & Card's model had some impact on later research. Evans & Chi present a model of social search, which provides a high-level view of social interaction before, during, and after the search process. Their model was published in (Evans & Chi, 2008a), (Evans & Chi, 2008b) and a bit simplified in (Chi, 2009). It includes Pirolli & Card's model and also builds upon Broder's classification of search tasks (cf. Figure 10).

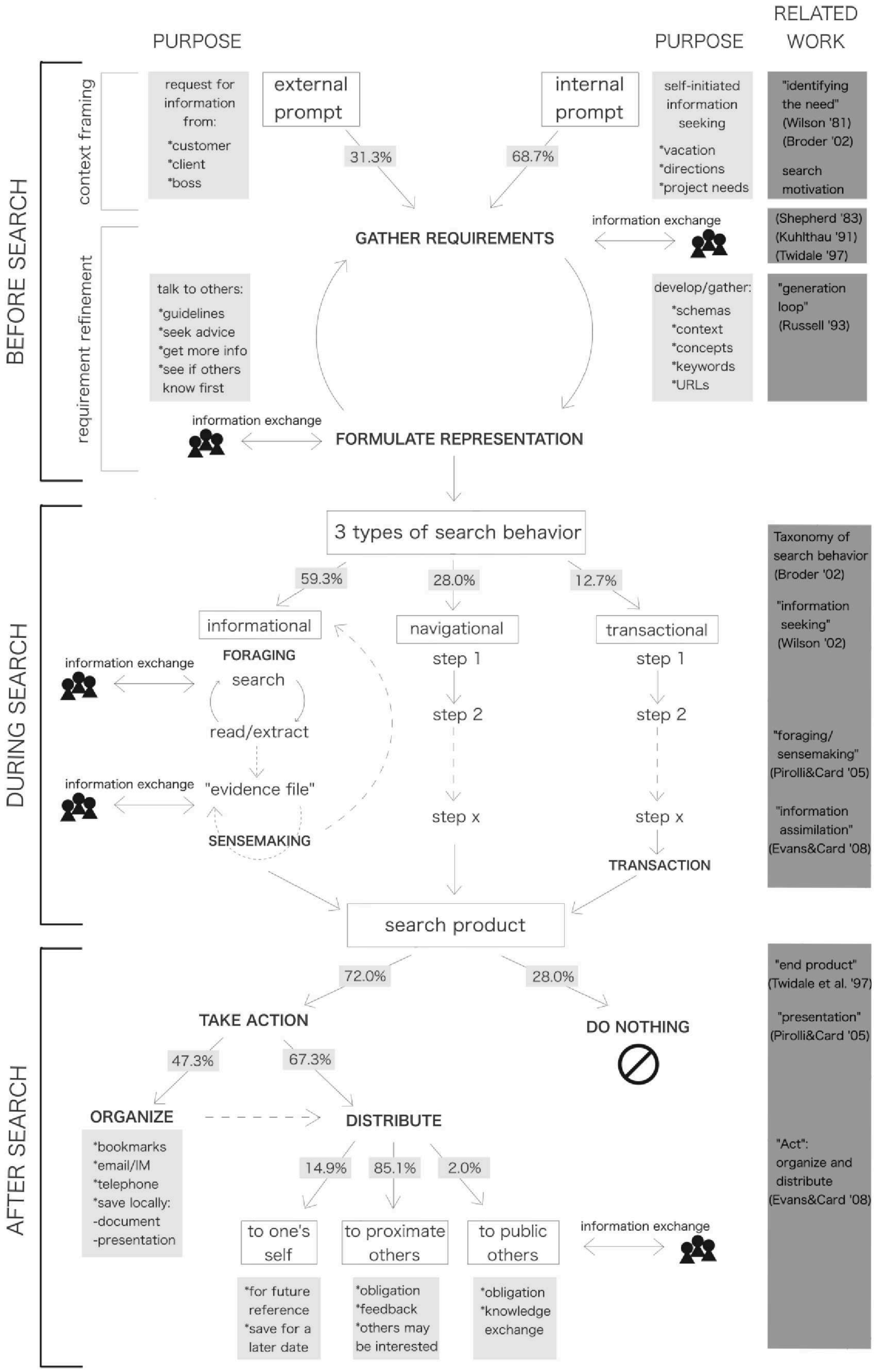


Figure 10: Model of social search (Evans & Chi, 2008a).

Evans & Chi split their model into three phases of a search task, which are 'before', 'during', and 'after' search and back up their model with numbers from empirical studies. In the first phase, before the search actually starts, the motivation for search and clarification of the requirements and search terms are the most important actions to be done. The model describes that during both of these two phases social interaction may take place. Almost two thirds of all searches are induced by intrinsic motivation. In the second phase, during the search, Evans & Chi distinguish Broder's three types of search tasks: informational, navigational, and transactional search behavior. While they assume a rather linear sequence of actions during navigational and transactional searches, the authors rely on the cyclic model of the sensemaking process by Pirolli & Card. The authors claim that social interaction may happen both during foraging and sensemaking. All three of Broder's types of search have in common that they result in a search product, which can be processed further and potentially be distributed to others. In the third phase, after the search, statistics show that almost three fourths of all search results are processed in some way, and of these results, approximately two thirds are further distributed. Most of the distributed search results (85.1%) are distributed to proximate others, just a minor share of the search results (2.0%) is distributed to public others.

Evans & Chi's model combines some of the approaches already presented (such as the search task classification by (Broder, 2002), or the model of search by (Pirolli & Card, 2005)) and covers the whole process of search. It points out social interaction for the first time, and could be extended by further instantiations of this model whenever an artifact induces a search at another person. Evans & Chi point out the value of search results of one user for other searchers, just a method for transferring that kind of information is missing: 'In other words, a method for explicitly or implicitly making available knowledge from single individuals or aggregated social networks may help users prepare for their search tasks' (Evans & Chi, 2008a, p.492 ff.).

The aforementioned models do not explicitly cover exploratory or complex search. Related to Marchionini's definition of exploratory search from 2006, a model of exploratory search was developed in a workshop at the ACM SIGIR conference in 2006 under participation of Marchionini (White et al., 2006b, p.58). This model was developed from components of related models in the workshop and is thus preliminary. It presents and connects core elements of exploratory search (cf. Figure 11).

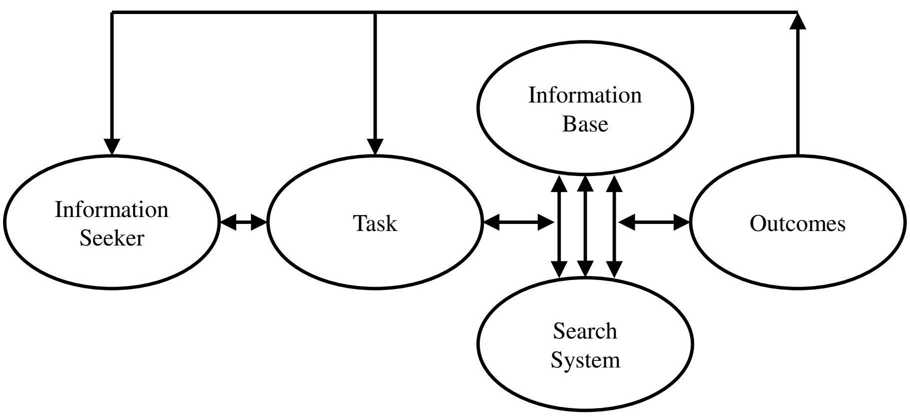


Figure 11: Model of exploratory search (White et al., 2006b).

The information seeker and the task are identified as the main factors in exploratory search, which influence each other. The task then influences the relation between the information base and the search system. The multiple arrows between these two elements suggest that there are many possible paths that the information seeker may choose. Similarly, the relation between the information base and the search system influences the outcomes of the search process, which again has an influence on both the task and the searcher, who may change or perform further iterations in the search process.

The model of exploratory search in (White et al., 2006b) has its strengths in clearly defining a cyclic process of search and the mutual influences that the core elements may have on each other. The multiple arrows indicate a strong connection on multiple paths between the information base and the search system, which are both influenced by the task itself and the outcomes of the search process. The information seeker interacts both with the search system (which in turn interacts with the information base), and with the information base itself, in case of direct access of information without consulting search systems.

Compared to the models of search described previously, it is apparent that the model of exploratory search stands at the beginning of a modeling process and was planned to be refined in the years to come. Unfortunately, my literature review has shown that this model has not been taken up by other authors or developed further. Nevertheless, this model is a first approach in formalizing the process of exploratory search. The workshop in which the model was developed also focused on the learning aspect of exploratory search, as 'exploratory search is often directed to learning tasks and the cognitive processes we observe as people explore information spaces' (White et al., 2006b, p.57). Based on this definition, it can be assumed that the evolution of this model may rather develop towards these characteristics and not cover the broader concept of complex search, as defined above.

A more recent concept related to exploratory and complex search tasks is 'content curation', which emerged around 2010. Even if content curation is not a strictly defined model, it illustrates a novel approach of dealing with an amount of valuable resources. Early thoughts on content curation were shared in the blog entry from (Bhargava, 2009), who claims that 'a content curator is someone who continually finds, groups, organizes, and shares the best and most relevant content on a specific issue online' (Bhargava, 2009). Based on this definition, Bhargava presents five potential models of content curation (Bhargava, 2011). Bhargava names the models 'Aggregation', 'Distillation', 'Elevation', 'Mashup', and 'Chronology' (cf. Table 7, based on (Bhargava, 2011)).

The five models grow in complexity, as the information contained in the end product of the process gets more and more processed by the content curator. The models for content curation are not as formalized as the models presented earlier in this section. Nevertheless, they provide an interesting overview of the different approaches for processing information. While 'Aggregation' is one key component of a complex search task and seems to be the foundation for the other four models, activities like 'Distillation' and 'Elevation' show similarities to the concept of synthesis from complex search. In a way, the models of content curation go beyond the models of complex or exploratory search, as they embed these processes into a larger scope of activities and provide motivation why searchers search in a complex or exploratory way.

Table 7: Five potential models of content curation (Bhargava, 2011).

	Model	Main tasks in the model
1	Aggregation	'Put together the most relevant information about a particular topic into a single location'. This may for example result in link lists.
2	Distillation	'Curate information into a more simplistic format where only the most important or relevant ideas are shared'. This process tries to simplify the aggregated information to ease understanding the aggregated content.
3	Elevation	'Curate with a mission to identify a larger trend or insight from the curated information'. This process tries to detect trends from scattered pieces of information.
4	Mashup	'Mashups are unique juxtapositions where merging existing content is used to create a new point of view'. A curated set of pieces of information here constructs a new or more complete picture of a topic than before.
5	Chronology	'A chronology brings together historical information organized based on time to show an evolving understanding of a particular topic'. This process extracts the main changes and trends in a topic and creates an overview based on historic events.

This section presents an overview of a number of models for search, ranging from early models of library search processes to late models of exploratory search and to content curation. It is interesting how the models grow more and more complex over time and how classic models of matchmaking are replaced by more cyclic approaches, of which some models build upon each other. For my thesis, the cyclic model by (Pirolli & Card, 2005) is most important. It both contains a cyclic approach of searching and also incorporates two layers of information processing, namely foraging and sensemaking. These layers are reflected in my approach by the aggregation and synthesis of information during complex web search processes.

2.2.5 Collaborative search

The approaches presented in the last sections mainly deals with single user's information needs and their actions and approaches to fulfill these information needs. One approach integrating other individuals into the search process is the model of social search by (Evans & Chi, 2008a). In this section, I present an overview of the research on collaborative search support. This overview collects exemplary positions from the research on collaborative search support.

Early approaches on collaboration for easing access to large amounts of information can be found during the early 1990's, when the Internet started enabling the exchange of information between people. One early approach on taking collaborative efforts to cope with the information overload resulting from newsgroup systems was presented in (Goldberg et al., 1992). The authors present a system that allows collaborative filtering by specifying filters to reduce the number of displayed e-mails in newsgroups. The presented system goes beyond related research of that time by explicitly including humans into the filtering process. Instead of forcing users to only subscribe to specific lists or to filter by keyword, their system provides possibilities to filter e-mails by interaction. Examples could be to only show e-mails that a certain colleague has annotated or that other colleagues have replied to (Goldberg et al., 1992, p.62 ff.). This approach implies that users need to have a good knowledge of other people's interest profiles.

The described manual approach for collaboratively reducing the amount of information to cope with has been automatized in later research. (Delgado et al., 1998) present a system which builds upon the ideas of (Goldberg et al., 1992). Their system allows users to classify web pages and to publish these ratings. A remote computer system collects the classifications and calculates interest profiles from them. Based on similarities in the interest profiles, the system suggests content from other users with similar interest profiles. This early version of a recommender system goes beyond the previously presented approach, as it does not require a single user to have detailed knowledge of the interest profile of other users, but automates and anonymizes this process.

The previous paragraphs describe approaches of coping with large amounts of information from a single user point of view. Collaboratively generated information is used to reduce the information overload for a single user. One of the first approaches for real-time user interaction and collaborative browsing is presented by (Gross, 1998). Gross presents a novel browser called 'CSCW3', which enables cooperation support during web browsing. CSCW3 discloses to all users, which other users are also using CSCW3 and are visiting the same website in this moment. Therefore, CSCW3 enables serendipitous meetings of users during web browsing sessions. Users can connect their browsers and control the other user's browser remotely, for example for the synchronous exchange of information or resources in the Web or for guided browsing. This approach is one of the first to enable direct collaboration between users during web activities and goes beyond the previous approaches in which users rely on collaboratively generated information. Gross' system could therefore be used for supporting collaborative web search tasks, but was not explicitly intended for this purpose.

While Gross' approach focused on collaborative browsing, the focus of the Social Web Cockpit (SWC) (Prinz & Gräther, 2000) is more on collaborative knowledge management. The SWC enables the collaborative creation of information collections. Running in parallel to a web browser, the SWC records the current web page's URL and checks whether it is already connected to a community, which is defined as a group of persons with the same interest. If the web page is connected to a community, the user can check the other sites of the community; if not, the user may add the web page as a resource to the community and therefore add to the information collection. The SWC further allows interaction with information, such as bookmarking or rating web pages, or adding other documents to the community. Adding to that, the users of the SWC constantly get informed which other users are currently working with it, or which pages other users of a community are currently on. This enables serendipitous meetings of experts. The SWC is an early approach towards collaborative search. Even if the main focus is not on search or search support itself, it supports exploratory and complex collaborative searches by allowing the creation of a shared information repository.

While the previous system is intended for general purpose web activities, other authors explicitly identify a need for supporting collaborative search. (Romano et al., 1999) state that there is a 'desire for collaborative search' and a 'demand for systems supporting multi-user information retrieval' (Romano et al., 1999, p.1) and thus point out a need for collaboration during web search. The authors present the CIRE system, which stands for 'Collaborative Information

Retrieval Environment'. This system allows the creation of an information retrieval memory, which captures e.g. the web pages visited and the users' queries. This data can then be shared among users. The system allows capturing relevant data to support both synchronous and asynchronous remote interaction. It turns out that users tend to forget about other users' contributions and make only infrequent use of the captured information when they are not pointed towards this information frequently enough. This system is one of the first approaches to explicitly address collaborative information retrieval. The focus of CIRE is not on exploratory search, but more on information retrieval, which is finding the optimal resource for a given query.

During the early 2000s', it seems that little progress was made in the field of collaborative search. The approaches presented above did not have significant impact on the landscape of research on collaborative search. Keeping in mind that exploratory search was not defined before 2006, this does not surprise much. In the late 2000's, some publications appeared that focused more on collaborative aspects in relation with search. One approach uses the tags that users attach to the web pages in their Delicious[22]-accounts as search terms (Briggs & Smyth, 2007). The system calculates the user's interest profiles based on these search terms and generates a user network from this. The network is then used to calculate a measure of trust between the users. Even if trust is not a focus of my thesis, taking user-generated tags as a basis for search term suggestions is an interesting approach for leveraging user generated tags to be a basis for search term recommendations.

Several papers show how social interaction affects search processes. (Evans & Chi, 2008a) show in which phases social interaction takes place during search processes. The according model can be found in Chapter 2.2.4 on the models of search processes. A similar need for social search is stated by (Chi, 2009), who claims that the view of search as information retrieval needs an update towards more social search processes: 'Information retrieval researchers typically depict information seeking as solitary activities of a single person in front of a web browser. This view is slowly changing.' (Chi, 2009, p.42). The author differentiates two classes of social search systems. The first class contains social answering systems, which utilize people with expertise to answer domain-specific questions. Examples for this are Yahoo!Answers[23] or Stackoverflow[24]. The second class contains social feedback systems, which use some sort of social feedback to rank search results. The feedback could e.g. be votes, tags, or bookmarks. Examples for this were 'direct hit', existing from 1998 to 2000, and an approach by Google to collect feedback by rating search results in 2011[25]. Social feedback systems declined in popularity, as they are prone to possibly automated influences, which could promote inferior resources over superior resources. (Chi, 2009) presents the social search system Mr. Taggy, which relies on 150 million crawled tagged bookmarks from the web. The system calculates the most popular bookmarks for each page and allows the users to search for specific tags, as well as interaction with the tags. If the users agree that some tags are especially helpful or not, they can vote the according tags up or down. Mr. Taggy is therefore a system that massively uses user feedback for improving the search experience of individual users and can be seen as a social search system.

---

[22] Retrieved September 16, 2015 from https://delicious.com/
[23] Retrieved May 12, 2015 from https://answers.yahoo.com/
[24] Retrieved May 12, 2015 from http://stackoverflow.com/
[25] Retrieved July 02, 2015 from
http://googleblog.blogspot.de/2011/03/1s-right-recommendations-right-when-you.html

In a paper on the design of information-seeking support systems (ISSS), (White, 2009) states that ‘we must journey beyond basic information finding and toward the attainment of high-level learning objective such as analysis, synthesis, and evaluation’ (White, 2009, p.55). This strongly reminds of Marchionini’s definition of exploratory search and the core tasks of aggregation, discovery, and synthesis by Singer (cf. Chapter 2.2.2 ‘Definitions’). Furthermore, White formulates a number of requests for future research which could help the searchers. His requests include support for specific search behaviors, support of facets for exploring domains via their attributes, and the visualization of information and according search results. Other requests are the support for learning and collaboration. Information-seeking support systems support the searcher during learning by ‘comprehending the information they encounter’ (White, 2009, p.56). ISSS support collaboration by supporting direct interpersonal collaboration as opposed to relying on indirect recommendations based on massive social data. White’s requests also point out the value of note taking during search tasks, as ‘there may be no single document that contains all of the required information’ (White, 2009, p.57). Even if the author does not introduce a concrete system, the research requests presented by White point towards future developments in the research area of collaborative search support and align very well with the approach of supporting collaborative, complex search.

Similar to the research requests presented in (White, 2009) which go beyond social search, (Golovchinsky et al., 2009) point out that ‘while social search has flourished over the past several years, true collaboration is still in its infancy’ (Golovchinsky et al., 2009, p.49). The authors present two scenarios in which researchers want to research on a complex topic, and also want to manage the acquired information and publish a meaningful subset of this information to collaborating researchers. The authors develop four dimensions of collaboration in information seeking, in which the respective systems can be classified. The four dimensions are (Golovchinsky et al., 2009, p.47 ff.):

1. **Intent: explicit vs. implicit**
   Implicit collaboration happens when systems infer user motivation from data, e.g. calculating recommendations for a searcher from the actions of other searchers. Explicit collaboration relies on a declared understanding of the – possibly changing – information needs of the collaborating searchers, e.g. by direct communication about the search topic.
2. **Depth of mediation: deep vs. shallow**
   This is understood as the technical level at which collaboration is represented in the search support system. The depth of mediation reaches from the shallow UI-level to the deep search engine level. On the shallow UI-level, a searcher directly sees the other user’s actions, but the search engine does not know how many searchers are searching at the same time. At the deep level of mediation (e.g. in collaborative filtering systems), the interaction happens inside the search engine, when for example new recommendations for the user community are generated from the interactions of a single user with the search engine results. The users do not collaborate directly. In these cases, the search engine differentiates the actions of single users.

3. **Concurrency: synchronous vs. asynchronous**
   Search support systems may support real-time interaction of searchers (synchronous), or the exchange of search related data to be evaluated independent of the other user's search (asynchronous).
4. **Location: co-located vs. distributed**
   The searchers may work at the same place at the same time or may be physically distributed.

One example for deep, implicit, asynchronous, and distributed collaboration is presented by (Ahmmari et al., 2012). The authors identify problems when making sense of the amount of information in online collaborative spaces, such as blogs, wikis, or discussion forums. In order to ease the access to potentially interesting content, the authors present an approach that extracts information from such spaces. The authors consider all content in a collaborative space, extract the content, and process the words with standard information retrieval methods, such as calculating word frequencies, normalization, and text extraction. The generated content gets clustered and topics are extracted from the clusters, which are transformed into topic clouds. These topic clouds help users getting an overview of important topics in discussion forums or other sources. The presented approach is rather non-standard in the landscape of collaborative search support, as it relies on massive data and distributed processing to cope with all the data. This approach leads to a reduction of topics to the most frequent ones and contains the danger that interesting, but infrequent information gets lost among the more frequent, but standard information.

A more practical approach for deep, explicit, asynchronous, distributed collaborative search is presented in (Capra et al., 2013). This approach relies on a preprocessed database with some 850,000 newspaper articles which are partially rated according to their usefulness for several given search tasks by study participants before the actual study. A browser called 'ResultsSpace' allows interaction with the database and allows the ranking of the found documents. For the study, the participants are asked to work on one of two predefined tasks. During the study, all participants are supported by the recorded actions of three 'collaborators', which worked individually on the same topic before. 'ResultsSpace' allows the participants to see the three collaborators' ten latest queries and the ratings the collaborators gave to the found documents. From an analysis of the interaction between the recorded collaborators and the participants, the authors extract three collaborative search strategies: independent, parallel, and divergent work.

- **Independent work** means that the collaborators were working without taking notice of the collaborators actions. Following this strategy, the collaborators had no influence on the participants' actions.
- During **parallel work** there is strong interaction with the collaborators' artefacts. Participants take the collaborators queries and investigate their result sets further, try to find more results on the current topic or rate the participants' documents.
- **Divergent work** means that participants try to build upon the collaborators' work to go into new directions. In these cases, participants tried to avoid overlaps and duplications of queries or results.

The presented classification of collaborative search actions is interesting, as it was not mentioned before. Unfortunately, the results are based on a study with only eleven participants. As such, the study shows some drawbacks that several studies on collaborative search share: The use of a limited and preprocessed database, a limited number of participants, and the lack of focus on a real search result, e.g. a report or a synthesis document.

In 2006, a study with 204 individuals – all working at Microsoft – investigates the participants' collaborative web search practices by a web-based survey. The survey consists of open-ended and multiple choice questions, asking for the use of specific search and collaboration technologies and asking about recent collaborative search experiences. In 2012, a replay of this study was done by (Ringel Morris, 2013). The 2012 study includes 167 American adults as participants and reveals that collaborative search is still a very relevant area of research, especially as 'web search is often considered a de facto solo activity, and nearly all mainstream search technologies are designed for single user scenarios' (Ringel Morris, 2013, p.1181). It turns out that in 2006, 54% of the participants from a tech-savvy audience had engaged in collaborative search, while in 2012, 65% of the participants from a general audience had engaged in collaborative search. Generally, younger participants are more likely to be involved in collaborative search than older ones. Almost 80% of all collaborative search efforts are done in groups with 2-4 individuals (Ringel Morris, 2013, p.1184), and mostly (approximately 85%) on informal topics, such as travel, shopping, entertainment, or news. The participants are mostly satisfied with both the informational outcome of the search process (83%) as well as with the ease of collaboration (78%). This result combines nicely with the results of both my evaluations, where participants are also satisfied with the current support for their collaboration and web search (cf. Chapter 5).

The previous findings are true for searches from a PC. The authors also investigate collaborative searches with smartphones. It turns out that almost 93% of all smartphone users reported to engage in co-located collaborative searches. The authors furthermore count activities on social answering systems like Ask, ChaCha, Mahalo, or Quora as social search. The study shows that only a small fraction of all users is actively engaged in such systems. Most users hardly ever use the mentioned systems, as more than 90% never posted a question there. The only exception of a systems used more frequently was Yahoo!Answers, where 24% of all users have at least posted a question once. However, only 4.8% of all users use the system regularly, by posting at least one question or answer per week (Ringel Morris, 2013, p.1188). In general, the study shows that more and more people are engaging in collaborative web search, and that younger users are more likely to do so than older users. There seems to be a tendency 'to appropriate existing communication technologies to create de facto collaborative search solutions' (Ringel Morris, 2013, p.1188), rather than using dedicated collaborative search tools. This gets confirmed by the finding that 'despite the increasing availability of tools designed to support collaborative web search, none of our respondents utilized such technologies' (Ringel Morris, 2013, p.1189). The participants use existing, simple tools that were part of their everyday routines, such as e-mails, phone calls, or instant messaging. This leads to the conclusion that a tool to support collaborative search needs to be lightweight first and foremost. Ringel Morris deduces that 'glue' systems, which connect existing systems, may have larger impact than tools dedicated exclusively to collaborative search (Ringel Morris, 2013, p.1189). Considering future developments, the 'results indicate that there is great potential for technological innovation to enhance the surprisingly commonplace practice of collaborative information seeking' (Ringel Morris, 2013, p.1190).

The value of Ringel Morris' study is both in the temporal dimension and the addressed audience. From the temporal point of view, the study shows by its samples from 2006 and 2012 that collaborative search is still a growing domain with lots of challenges to be addressed. The second study is based on American adults, which provides the research community with important results from non-professional users. These results show the importance and the possible impact of support for collaborative search.

A study of the usage of a collaborative search support system was done in (Kelly & Payne, 2014). The authors evaluated the collaborative search tool 'Coagmento' (González-Ibáñez & Shah, 2011) in the context of everyday tasks, but as this turned out to be defective for one group, they decided to use 'Diigo'[26] for the other study groups instead. The authors confirm the findings of (Ringel Morris, 2013), in that 'the actual success of collaborative search systems, in terms of mainstream take-up, has been fairly limited', as they 'require too much effort' and 'do not offer meaningful benefit over ad hoc practices' (Kelly & Payne, 2014, p.807). From an analysis of previous studies and systems, the authors deduce a set of four basic aspects of collaboration a system should support to be able to support collaborative search. These four basic aspects of collaboration are (Kelly & Payne, 2014, p.808):

- **Awareness**, which means the ability to acquire knowledge about the current and past actions of the interaction partner.
- **Division of labor**, which means the process of distributing a task across members of a group.
- **Persistence**, which means storing and displaying the activities of prior search sessions.
- **Sensemaking**, which means the support of understanding what has been found and how it was found. Example means of supporting sensemaking could be the visualization of search strategies and search trajectories.

The systems selected for the user study provide support for all four aspects of collaboration. For the study, eight pairs of participants are selected from the university population. Following the definitions given in Chapter 2.2.2, the participants are given both exploratory (travel planning, house hunting) and complex (shopping) search tasks. The participants have no time limits for the study, and they performed up to 15 search sessions over 14 days. Diigo is a Firefox browser extension which allows capturing useful pages as bookmark and / or screenshot, sending content to collaborators, and saving the results to a cloud service. Diigo stores only the web pages bookmarked explicitly, while Coagmento stores all visited pages. Diigo allows both synchronous search, where the collaborators' results are exchanged during the search, as well as asynchronous search, where a collaborator can explore all of the partners' actions afterwards. A mobile version of Diigo would have been available, but was used by none of the participants.

During the study, it turns out that the participants are generally positive about the design concepts behind Diigo, and that the system has advantages over ad hoc solutions. This can be argued to be based in the rather lightweight concept of the selected system, especially as all participants state that they did not use the more complex features like recommendations, the shared

[26] Retrieved June 17, 2015 from https://www.diigo.com/

editor, features to read content later, or the possibility to send information to the collaborator. The synchronicity of work seems to depend on the local setting. In the case of co-located search actions, the participant pairs evenly chose to do synchronous and asynchronous search actions, while all participant pairs chose asynchronous search in a distributed setting. Half of the participant pairs organized their work in advance, which means that they split search tasks explicitly, while the other four pairs did not organize their work at all.

With regards to the actual search process, participants report the strategy of starting with broader search topics and narrowing down the topics over time. This process is reported over several sessions; as such, it is important for collaborative search support tools to support splitting search processes into several sessions. While complex features are not used by the participants, all make use of the basic features, which are bookmarking and highlighting valuable content. This also helps avoiding the creation of duplicates during searches. The participants like the highlighting especially because it allows them to easily go back to valuable content by remembering the content, and not the page itself (Kelly & Payne, 2014, p.813). Shortlists of resources and content especially help to agree on valuable and satisfactory search results. When it comes to time critical situations, none of the participants uses the built-in features for sending content directly to the collaborator; all participants go for ad-hoc communication solutions, such as calling, sending e-mails, or messaging.

Considering user interface design decision, the study reveals some insights with respect to the value of certain information. The participants had the chance to look into the collaborators' search history. It turns out that simple chronological lists of visited URLs are almost useless, as the history is overwhelmingly large. It would have been more useful to get rid of irrelevant pages during the search session and to 'see only the most relevant information without having to crawl through the entire history' (Kelly & Payne, 2014, p.814). This could be achieved by selective tracking, where only user-defined parts of the search process get captured. Interesting results could be achieved when it came to the visualization of all visited pages. In Coagmento, each item of the shared search history appears as a thumbnail. The group that used Coagmento did not like the thumbnails, as they asked for a summary of contextual information, such as the relevant findings on the page and a headline of it. Diigo provides the page's title and more contextual information, but no thumbnails. Surprisingly, the Diigo users state that they would have liked to see thumbnails. Nevertheless, participants found it most helpful to annotate certain pages and to state why these pages have been visited and found helpful (Kelly & Payne, 2014, p.815). Additionally, users wanted to classify or categorize the found pages, but did not want to rate pages for other users, as this would have been too much effort. It is important for the users to understand what the collaborators have found during their searches and how they came to their results (Kelly & Payne, 2014, p.817). Interestingly, the authors can confirm Ringel Morris' results with respect to lightweightness of the technical solutions. Both Coagmento and Diigo included many features 'which were not necessary for their task', and 'future solutions could be scaled back in favor of lightweight support for core collaborative search behaviors' (Kelly & Payne, 2014, p.817).

The results of this study point out some important design and usability issues for collaborative, complex search support systems. The value of the study in (Kelly & Payne, 2014) lies in its setting without time limits and very open-ended search tasks. This adds value to the insights gathered by the authors and can be helpful for the design of systems for supporting collaborative search. Nevertheless, the study has its flaws in a limited number of participants and the way in which the results were gathered. The authors chose to do structured interviews with the participants and did not evaluate the data gathered during the searches, which made the participants only remember the most significant events during the possibly long timespans and allowed them to forget other important details.

The approaches in this section give a chronological overview from the early 1990's, where the emerging Internet influenced researchers to think about involving multiple persons into one search process to recent studies on collaborative search. It is interesting how the research on collaborative search becomes more and more open, as it was first oriented towards collaborative browsing and slowly turned towards the creation of collaborative artifacts during search processes. Only the later studies from 2010 onwards show very open search scenarios which no longer rely on predefined databases and encourage users to interact freely over space, time, and the information explored.

For my thesis, the technical classifications of collaborative search systems from (Golovchinsky et al., 2009) and (Kelly & Payne, 2014) are important for classifying collaborative search support systems. Also, the study by (Ringel Morris, 2013) provides valuable insights from user evaluations of collaborative search support systems which influences the design of my approach.

2.2.6 Negative search

The previous sections deal with search as a process for producing results, which are intended to be as meaningful as possible. So far, none of the presented approaches or definitions points out that also no result or a negative result can have a value for the searcher.

In certain cases, searchers may not be interested in finding a result, because their aim is to confirm that the searched information does not exist. This may be not so important for simple fact searches, where searchers rather look for something they know is existing. In more complex search cases, searchers may want to not find a piece of information. (Garfield, 1970) is the first author to point out the value of negative information for a search process, and coins the term 'negative search' for this.

From the twofold definition of exploratory search, two exemplary search cases can be derived in which negative search results could be positive. First, in scientific search scenarios, a scientist may want to confirm that research on a certain subject has not been done before or that a paper about a certain topic has not been published before. Second, in more open-ended scenarios, searchers may deduce from a search result that certain actions are not possible. For example when booking a holiday trip, searchers may discover that it is not possible to take certain train connections in Italy due to winter timetables for certain railway companies.

These search scenarios have in common that they do not produce a definite result that is easily transferable to other searchers. Furthermore, a single positive search result has the potential to diminish the value of all other achieved negative search results, since this single result can possibly confirm that certain research has been done or that a certain train is operated on a specific day. For these searches, the results are less important; it is more important how and from which resources these results were obtained. Based on this information, other searchers would have the possibility to re-trace which resources were initially visited and where the searcher found nothing. Maybe later searchers can easily come up with resources that the previous searcher did not cover and provide suggestions where the intended information can be found. Later searchers could also verify that a searcher covered all relevant resources.

However, Garfield's work did not have a large impact on later research. Even if the paper was written in 1970, it was only cited six times[27]. Despite one citation by Garfield himself in 1971, all other citations appeared after 2005. One citation comes from the context of issue maps. The other four citations appeared in the context of exploratory or complex search, and are in (Marchionini, 2006), (White & Roth, 2009), (White, 2009). The fourth citation was made in my paper published and presented at CASCON 2014 (Franken & Norbisrath, 2014a). Considering the age of the paper, these statistics show the low impact of the concept. This low impact may be due to technical unavailability of systems that support the capturing of this information.

A way of capturing this kind of negative information could thus impact the performance during complex search tasks. Capturing search trails could offer a possibility to capture the complete path of a web search, covering all resources where searchers did not find relevant information.

2.3 How do users search and which problems occur?

The previous subchapters describe the search processes from a scientific point of view. Chapter 2.2 presents definitions of core terms for this thesis. These definitions help framing the scope for the actions to be supported with the work in the thesis. Classifications of search tasks organize the types of actions a user performs when trying to satisfy an information need. Models of search try to formalize the process of search, and a section on collaborative search takes a glance on the concepts for interaction during search processes. Even if the presented findings are derived from the actual behavior of users, there is less literature on search tasks from the users' perspective. This section presents some exemplary insights on the user's typical behavior while performing web search tasks.

An early study on user behavior during search processes is presented in (Marchionini, 1989). Marchionini uses a CD-ROM based digital encyclopedia for his experiment, where he seems surprised by the information capacity of a CD-ROM. The study was conducted with third, fourth, and sixth graders from an urban school and contained two tasks which could be solved by the information in the digital encyclopedia. The first was a simple fact-finding task; the second task was more complex and open-ended. It turns out that all searchers roughly need the same number of actions to get to a result (if they found one), and that older searchers are more successful than younger searchers. The searchers make use of several search engine commands, e.g. 'AND', and other search functions, such as truncation or proximity of search results. This may also be due to two 45-minutes introductory presentations. Marchionini finds that most users do not stick to a fixed search strategy, but that the process of search remains highly

[27] Retrieved April 21, 2015 from
https://scholar.google.de/scholar?cites=963392326401677749&as_sdt=2005&sciodt=0,5&hl=de

intuitive and relies on the results being found. Most participants use a strategy of widening and narrowing the original query, with support of search engine specific functions. Marchionini recommends system designers to make use of meaningful defaults and to reveal options to the user to enable them to make full use of a system's features. The paper is interesting, as all users are introduced to the system and its features during presentation sessions, which results in a strong use of more complex search engine functions. The approach of revealing specific search engine functions to the user seems to illustrate a shift of search systems to take the mental burden off the user and onto the system. For the system developed in my thesis, this implies to make all functions easily visible within the user interface (cf. Chapter 3 and Chapter 4).

An analysis of search engine queries was done by (Jansen et al., 2000). The analysis was conducted with 51,473 queries of the Excite[28] search engine with respect to sessions, queries, and terms in search processes. The authors detect a decreasing value of plain information retrieval for Internet search, as 'real life Internet searching is changing information retrieval' (Jansen et al., 2000, p.208). The evaluation of the queries reveals that on average 2.84 queries are entered by a user per session (Jansen et al., 2000, p.212). The analyses show that users typically do not add or delete many terms from their start query, but rather change words. The average query length of 2.21 terms (Jansen et al., 2000, p.214) makes it hard for search engines to correctly deduce the searcher's intention. As most search engines offer the possibility to use logical operators (e.g. 'AND', 'OR', or 'NOT') or query modifiers like plus, minus, or quotation marks, the authors also investigate their use during the search sessions. It turns out that hardly any searcher makes use of the Boolean operators (5% altogether), and of these users, about a third uses the operators the wrong way. Query modifiers are used more frequently, by approximately 15% of all users. However, they are used incorrectly by about a third of the users in case of the plus and minus operator, while the quotation marks are used correctly in almost all cases. Jansen's results indicate that ordinary search engine users show a somehow static behavior during search processes, which is influenced negatively by misconceptions about technical details of search engines and the faulty use of search engine operators.

A study with 236 experienced users about information search and re-access strategies for known information was conducted by (Aula et al., 2005). Similar to the results of (Jansen et al., 2000), the authors find that only around a third of all search engine users are able to correctly explain which criteria a result must fulfill to appear as a result (Aula et al., 2005, p.589). This leads to a faulty use of search operators. Overall, the use of Boolean operators is rare (Aula et al., 2005, p.588). The authors also find that experienced users rarely rely on the history, as the history function of a browser is limited to one browser and one computer. Adding to that, the history is normally cluttered with irrelevant sites and relies on sometimes misleading page titles, which the users may not remember (Aula et al., 2005, p.588). Similar to the behavior of unexperienced users, the study participants use short queries and rarely check the search results beyond the second search engine result page. Experienced users may be more successful with short queries, as they choose search terms not based on the information they want, but because they 'imagine someone wording a site that contains that information' (Aula et al., 2005, p.589).

The authors find common strategies for experienced users, which rely on a high information processing capacity. Some users manage their search process by using multiple windows

[28] Retrieved May 20, 2015 from http://www.excite.com/

and tabs in parallel. This way, important results can stay open. Some participants even use two browsers; one for actually searching results, and the other one for dragging in links that are found in the first browser (Aula et al., 2005, p.587). Another strategy for extracting information from result sets is categorization. Some participants mention the use of search engines which cluster or categorize the results. This is especially helpful when the searcher is unfamiliar with the search domain (Aula et al., 2005, p.587). For these purposes, the study participants use Vivisimo, a search engine which was acquired by IBM in 2012 and was integrated into IBM Watson Explorer[29]. The authors find that the common strategy of sending important links via e-mail to oneself is not pursued by experienced users, as this mix of resources among different media and tools is cumbersome (Aula et al., 2005, p.588). The results gained in this study reveal that experienced web searchers use strategies that support their high information processing abilities, e.g. tabbed browsing or categorizing results. Nevertheless, some misconceptions as to the functions of search engines are common even among these users.

Jansen's analysis of search engine queries is especially valuable as it shows the search engine usage during the year 2000 for ordinary search engine users. It becomes obvious that user interaction with search engines is rather short, both in number of queries per session and in terms per query, and is not altered a lot. The following paragraphs present a series of studies on the behavior of ordinary users of web search engines, which were conducted by Nadine Höchstötter (nee Schmidt-Mänz) during her doctoral studies in the mid-2000's.

Using an online questionnaire, she collected around 6,000 answers from normal Internet users on their personal search behavior. She asked the participants about the following five categories: the standard use of search engines, the use of more complex search functions, their background knowledge about search engines, navigation on the Internet, and demographical information (Schmidt-Mänz, 2005). Even if this study suffers of a self-selection effect, which is e.g. visible by a high share of power users with 57% of them having their own website, it reveals some insights about the interaction of tech-savvy persons with Internet resources.

As to expect from a German questionnaire, most users (approximately 57%) do their searches exclusively in German language. Almost 77% of all users use short queries, consisting of 2-3 words, leading to an average query length of about 1.7 words. Almost 71% of the participants return to the search engine result page very fast if the desired result is not found immediately. This behavior can also be found in some search trails from my own second user study (cf. Chapter 5.2), which resulted in star- or flower-shaped trails (cf. Appendix A3). Despite a large share of the study participants being power users, it turns out that only about a third has in-depth knowledge about the technical backgrounds and the inner workings of a search engine (Schmidt-Mänz, 2007), which again confirms the findings in (Jansen et al., 2000) and (Aula et al., 2005).

Considering the use of more elaborated search engine functions, she finds that around 52% of all participants use operators like 'AND' frequently or very frequently, while 64% rely on search phrases (delimited by quotation marks) frequently or very frequently. Most users (70%) change their query when no results are found, and almost nobody switches the search engine due to unsatisfactory results. Approximately 75% of all users do not personalize their search

[29] Retrieved May 21, 2015 from http://www-01.ibm.com/software/data/information-optimization/

engine intentionally (Schmidt-Mänz & Bomhardt, 2005, p.7). This means that users stick to one search engine, do not use all of its features, and try to improve the quality of search results by altering the query. Users are most frequently annoyed by results that do not have a relation to their search query, ads, and duplicate search results. Even if users are aware of the quasi-monopoly of Google, almost 90% do not favor having only one search engine available.

A second publication on the same study reveals that circa 34% of the study participants start searching without a concrete information need, mainly for discovering interesting information (Schmidt-Mänz & Bomhardt, 2005). These searches lead the users to pages they already knew to catch up with more recent information. A large share of participants uses bookmarks for this purpose (69%).

From her results, Schmidt-Mänz argues that a sort of taboo search could be helpful. Similar to Garfields remarks on negative search, negative information can be of great value when users know what they do not want. Especially for vague information needs, users can often more easily specify what they do not want to be included in their search results than what they want to see (Schmidt-Mänz, 2007). She also performs time-based analyses of search queries. A visualization of search terms in (Schmidt-Mänz & Koch, 2006) reveals recurring search terms, such as searches for the cinema program on Thursdays. Other analyses on the frequency of search terms reveal that there exist mayflies, which are search terms that pop up once and disappear very fast, while there are also evergreens, which are search terms that appear very frequently, independent of temporal influences, like the search terms 'flight' or 'vacation' (Schmidt-Mänz & Koch, 2006). These results are language independent.

The interaction of searchers with search engines seems to be error prone. While search engines can hardly determine the searcher's intention from short or ambiguous queries (Höchstötter, 2007), search engine users do not interact with search engines properly. An analysis of popular search queries reveals that these contain a number of irrelevant terms, which are often ignored by search engines, e.g. 'the' or 'a'. Similarly, despite the frequent use of search engine operators, they are often used faulty and result in unsatisfactory search results. Höchstötter therefore argues in favor of more intuitive search interfaces (Höchstötter, 2007, p.140).

From the results presented in the aforementioned studies, I deduce that at the time that the studies were conducted, exploratory and complex search lacked support. Additionally, Höchstötter's studies reveal some parallels with Jansen's results, as to the limited query lengths and errors being made when using query operators. Even tech-savvy users are not aware of the inner workings of search engines and use very short queries. A surprisingly large share of all users use search operators and phrases to limit search results. Offering information collections of evergreens to searchers or recommending frequently recurring search terms (like searches for the cinema schedule on Thursdays) could have the potential to help users, but are still rarely supported (Höchstötter, 2007).

A study by (Lewandowski, 2006) analyzes the query types of 500 queries of ordinary German search engine users from each of the search engines Fireball, Seekport, and MetaGer (cf. Chapter 2.2.3). An analysis of all queries reveals that certain categories of search topics are more likely to contain informational queries, as defined by Broder. These categories are 'health or sciences', and 'society, culture, ethnicity, or religion'. Other categories like people, places, commerce, government, education, or arts are more likely to contain navigational or transactional queries. This can be explained by the high complexity of the former topics, resulting in comparably vague search queries. For the latter categories, it can be assumed that users have a clear goal in mind when starting a search on a certain topic. Lewandoswki's research is interesting as he clusters search queries both by Broder's classification and by topic, which reveals the topic's influence on the average complexity of the search.

(Gwizdka, 2008) investigates the subjective difficulty of search tasks. For his study, he uses twelve different search tasks, where four are simple fact finding tasks and the rest are complex information gathering tasks on the English Wikipedia. Half of the information gathering tasks can be solved by hierarchical results, where multiple characteristics of a single concept had to be found, which reminds of a depth search. The other half can be solved by parallel gathering of pieces of information on roughly the same level of abstraction, which reminds of a breadth search. Each participant performed six searches and always had the chance to choose between two tasks of the same category. The study reveals that the main influence factors on the subjective difficulty of the search task are the searcher's effort (as measured by number of visited pages or the number of bookmarks), and the task type and structure (Gwizdka, 2008, p.8 ff.). The searcher's effort is mainly influenced by individual differences in the participant's cognitive characteristics. Even if these results are not very surprising on first sight, they reveal how much the necessary effort for solving a task depends on the searcher and the task itself. These findings indicate why it is especially hard for search engines to provide appropriate support for all searchers. Additionally, users may be experts on one topic, but not on another topic, which would require different levels of support for the same searcher.

In contrast to Gwizdka, where the difficulty of search tasks is judged after the search process, (Singer et al., 2012c) performs an experiment where ordinary search engine users perform simple and complex search tasks and judge the expected complexity of the search tasks beforehand. This is done by answering four binary statements, e.g. 'It will take me less than 5 minutes to complete the task' for assessing the required time. After the task, the searchers judge the complexity of the search task again, which allows the evaluation of the assumptions. It turns out that for simple fact-finding search tasks, the searchers are mostly able to judge the difficulty, time effort, query effort, and their own ability to find the right result. Around 90% of all users can estimate the required efforts correctly for simple search tasks. When it comes to more complex tasks, that number decreases, and only about 65% of all users can still correctly estimate the required efforts. Similarly, users can easier judge whether they achieve a correct result for a simple search task (87%) than for a complex search task (52%). The study reveals that good searchers are significantly better in judging the search tasks' outcome than searchers performing lower. Summing up, the study shows that the task complexity influences the judging performance of users, and as Gwizdka shows, the subjective task difficulty largely depends on the user. As such, the combination of these two studies indicates that it is very hard for users to determine how hard a complex search task will be and how good and meaningful the achieved results are.

Another publication based on the same study is (Singer et al., 2013). In this publication, the authors compare 13 technical measures of search tasks. These measures include the number of sessions, the time spent on search engine result pages, read times, task times, overall search time, and the number of visited pages. The data for the evaluation are generated by ordinary search engine users. The authors compare these values to find out differences in these measures between simple and complex search tasks. They show that on average, there is always a significant difference between simple and complex search tasks for all 13 statistical measures. The authors then compare these values to determine whether they can indicate the success of a search task. Taking a closer look on the first and fourth quartile of successful and unsuccessful users, the differences mostly disappear. Only for three of 13 measures a significant difference can be found with respect to the success of a search task. These measures are the time spent on search engine result pages, the overall task time, and the overall search time, which is the time needed for the task plus breaks for taking notes. These results show that there is a significant difference in complexity between simple and complex search tasks, but it is hardly possible to detect the success of a searcher from statistical measures.

In opposition to the previous experiments with ordinary search engine users, an experiment with library search experts was conducted in (Singer et al., 2012d). The authors define two main classes of Internet search strategies, the 'Known address' and the 'Search engine' strategy, depending if the user starts with navigating to a known (non-search engine) website and starts searching there, or if the search is started on a search engine page. The authors show that most participants make use of the 'Search engine' strategy and narrow and extend their queries until a satisfying result is achieved. All participants solve the given search tasks, which indicates that search engines are good entry points for exploring search spaces. However, other statistical measures collected during the study, like the number of open and closed tabs or the number of tries per search task indicate that search engines only provide a low level of support for complex search tasks. Participants use multiple strategies during the same search task and the strategy used depends on the task, the skills of the searcher, and the knowledge of the person (Singer et al., 2012d, p.95). The results indicate that even information searching professionals do not rely on a fixed strategy and adjust their information seeking behavior to the actual task and situation, which makes the support of complex search tasks especially hard.

The studies presented in this chapter examine the search process from a user perspective. Some of the studies manage to acquire ordinary users; some rely on more tech-savvy users who want to take part in the studies. The studies show differences between these two user groups as well as similarities. The similarities with respect to the very short query lengths are mentioned frequently and get confirmed in (Lewandowski, 2015, p.76), as he mentions that users did not change much during the last decade with respect to query formulation. Overall, it turns out that all influence factors on the search process heavily depend on the individual searcher; may it be the subjective complexity of the process, the objective success of the task, or the ability to estimate the own required efforts and results. This indicates that the level of required support during search is not constant and not the same for all users, which makes offering support for complex search tasks hard. For my thesis, these results imply that a tool offering a very specific feature set may be contra-productive. A lightweight tool may be simple enough to be understood by all users and can be integrated into the user's search process only when needed.

2.4 Related concepts

The first subchapters cover the historical origins of search trails as well as concepts, definitions, and models of search and of search processes. The previous subchapter provides insights on search processes as they are actually performed by searchers who are not necessarily information professionals. These subchapters provide the main related work for my thesis, as they deal with search trails, collaborative search, and search support from an academic perspective. They show an evolutionary process, which started by printed library catalogues and improved along digital library information portals, digital encyclopedias to early forms of web search and later on collaborative search. It is obvious that complex and collaborative search are relatively new ways of interaction with online information, and that the support for collaboration during search processes is still in its infancy.

The following section provides some information on further related concepts for organizing and accessing information and explains the differences between support for collaborative search by building search trails and the mentioned concepts. This is important as the reader could argue that any existing concept can be used to provide the envisioned level of support for collaborative, complex search, but it turns out that this is not possible.

Lots of different ways of structuring and accessing information have been developed during the last years, as well as predictions have been made on what Internet search will look like in the future. These predictions do not necessarily become true, which means that lots of the old problems stay, and lots of established mechanisms are still used for accessing information. Exemplary for these predictions, a vision from (Hanani & Frank, 2001) shall be taken. From a perspective of 2001, the authors claim that digital libraries, which are suited to a specific set of information on one topic, could better provide dedicated services for Internet search than general purpose search engines. For 2001, there were more than 3,450 search engines and more than 10,000 databases and archives available on the Web (Hanani & Frank, 2001, p.212). This leads to a very limited coverage of the Internet by search engines, not exceeding 16% of the estimated size of the publicly indexable web (Hanani & Frank, 2001, p.212). As this problem will most probably become more apparent, search engines have to develop further to catch up with these developments. The authors identify four phases of search engine evolution. The first are basic search engines, which rely on indices and web directories which just enable access to information. The second generation of search engines consists of meta search engines, which do not have an own index but distribute requests to many other search engines at the same time and combine their results. The third generation consists of popularity search engines, which combine indices with information about the structure of the web, such as Google's PageRank algorithm. A similar evolution can be developed for digital libraries, which evolved from self-contained, stand-alone digital libraries over networked federal digital libraries to harvested digital libraries, which connect metadata to the correct resources. The recent fourth generation of both search engines and digital libraries is claimed to be intelligent, combining the information resources with artificial intelligence to exactly support the needs of the searcher (Hanani & Frank, 2001, p.217). From this evolution, the authors envision a convergence of search engines and dynamic libraries into a mega-portal which unifies both approaches. This mega portal was envisioned to be a 'unified, high-level transparent interface that will be used for declarative search and browse of all data repositories' (Hanani & Frank, 2001, p.217). Obviously, the evolution of search engines did not come that far within the last fifteen years, even if the quasi-monopoly of Google dominates online search. For now, some approaches can be seen that provide the searcher with related content, but a mega-portal consuming all available databases and offering declarative search will not be realized within a significant amount of time. This example shows that major changes are to come very slow and that new technologies and approaches need to be integrated with existing services.

The rest of this subchapter is divided into three sections, which show exemplary influence factors on the overall concept of search trails and – where necessary – explain the differences from them. The first part shows techniques for organizing resources, while the second part presents concepts for accessing information. The third part shows some approaches for leveraging and distributing information among communities.

2.4.1 Techniques

There are several techniques which can be used to organize information from complex search tasks for the personal or community use. One technique is bookmarking relevant pages. This has the advantage of every standard web browser being able to support this technique. Another advantage is that the bookmarked URL is available to the browser, which may offer the resource to the user when typing URLs or queries into the address bar. Unfortunately, the value of bookmarks is limited; they are a 'common, but flawed mechanism for storing information' (Aula & Russell, 2008).

This critique is based on several points. First, bookmarks have to be organized after being created. This organization requires a certain effort, especially when it comes to large bookmark collections, as reported in (Aula et al., 2005, p.586), with e.g. hundreds of bookmarks or up to 2,500 bookmarks in about 400 folders. As not all users of web browsers organize their bookmark collections meticulously, we can assume that bookmark collections are often cluttered and therefore are of little help. Re-finding a certain bookmark is a search effort on its own just in a different location. Second, the value of a bookmark is often not apparent when it is being created. Initially interesting web pages may be replaced by more interesting pages, or a resource turns out being valuable after the web page was left. This leads to missed bookmarks or cluttered collections. Third, bookmarks only save the URL of a resource, without comments or other information. As the whole URL is stored, the web page may be much larger than the relevant portion of it that triggered bookmarking the page: 'Participants wanted to annotate and append specific pages to provide rationale about why results have been selected, and to draw attention to specific aspects of web pages' (Kelly & Payne, 2014). Fourth, the actual information may be lost when visiting the bookmarked page again, which may happen when visiting news pages or forums. While a certain page is bookmarked, the entries can have moved to other pages due to more recent entries.

The fact that most bookmarks are not easily transferable from one browser to another is another reason why bookmarks are not the first option for organizing information collections. In this case, social bookmarking could help. Popular platforms for social bookmarking are Delicious[30], Digg[31], or StumbleUpon[32]. Most social bookmarking systems allow – besides the

[30] Retrieved May 21, 2015 from https://delicious.com/

bookmarking of pages – tagging and categorization of bookmarks, the publication of bookmarks, and the import of published bookmarks into own collections. The generated public bookmarks collections are browsable and searchable, which allows the exploration of the public space of bookmarks. The approach of social bookmarking adds a social component to the user-centered browser bookmarks. However, the mentioned flaws of bookmarks remain, as the tags on the social bookmarks are only of limited help. Similar to traditional bookmarks, the social bookmarks require to be organized by tags. Social bookmarking is closely connected to terms like social tagging or folksonomies. Social tagging describes the process of tagging resources by a large number of users, as it 'describes the process by which many users add metadata in the form of keywords to shared content' (Golder & Huberman, 2006). The term 'folksonomy' describes the result of this process, a classification which is generated by collaborative creation of keywords.

Another way of organizing information is a topic map. Topic maps are one of several semantic approaches of formalizing relations between pieces of information. Topic maps are an ISO / IEC standard for the representation of information and the exchange of knowledge[33]. They consist of several semantic elements which are instantiated by information, e.g. subjects, associations, scopes, or occurences (Ullrich et al., 2003). The ISO standard itself foresees a range of application areas for topic maps, which include the structuring of information, the connection of topics to form a network, the development of different views onto one topic (e.g. with dedicated user rights), and the creation of indices, glossaries, or cross reference lists (Widhalm & Mück, 2002, p.90). The ISO standard explicitly describes the creation and the structure of a topic map, but not a data model or a query language for it. The strong formalization of topic maps as an ISO standard ensures reliability and exchangeability, but limits flexibility when it comes to new developments. One could imagine supporting web search processes by topic maps, as they are free to be filled with whatever content, but the strong degree of formalization makes it hard for users to effectively work with them in a flexible way. Especially when users are not trained to work with a formalized tool, interaction can be experienced as tedious and cumbersome. Similar to bookmarks, topic maps have to be explicitly filled with information, which again requires the users to take action to effectively capture relevant information. My literature review shows that topic maps have rarely been used for organizing knowledge in search support.

The presented concepts in this section stand exemplary for a range of techniques dealing with information storing and structuring. While all these technologies have their advantages for their respective use cases, they also have flaws when it comes to multi-user scenarios or the organization of complex information structures, where pieces of information may build upon another.

2.4.2 Concepts

Besides the realized techniques a number of concepts exist that deal with accessing information. These concepts can be instantiated to form real systems, which has been done a number of times. Nevertheless, the practical impact of the presented concepts is limited for my purpose.

---

[31] Retrieved May 21, 2015 from http://digg.com/
[32] Retrieved May 21, 2015 from https://www.stumbleupon.com/
[33] Retrieved May 21, 2015 from http://www.iso.org/iso/catalogue_detail.htm?csnumber=38068

This section is split in two parts: The first part gives a brief overview of older concepts like information retrieval or knowledge management, which have already been mentioned several times. The second part covers the concept of making data semantically accessible on the web, of which I present here Linked Data, RDF, and microformats.

The term 'information retrieval' (IR) has been mentioned in earlier chapters of my thesis. In (Manning et al., 2008), information retrieval is defined as 'finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)' (Manning et al., 2008, p.1). Key problems of information retrieval are the processing and structuring of information to create indices that allow the automatic evaluation of these indices (Manning et al., 2008, p.67 ff.). IR originates from a time where libraries were the primary source of information, and one of the main questions was how to find a printed resource which most probably contains the desired information. Early work on the performance of IR was already done by (Robertson, 1977). These days, the focus is less on finding the title of a resource containing all information, but more on assembling the desired information from multiple resources. This implies that the core assumption of IR has changed over time. Nevertheless, IR is still relevant as it provides many of the technical resources and concepts which enable Internet search on a very basic level.

In order to overcome the emerging gap between classic IR and later developments of more exploratory or complex cases of web search, new terms have been coined, such as 'Interactive Information Retrieval' (IIR). This term unites human interaction with information retrieval systems and thus builds a counterpart to the rather 'system centered' plain information retrieval systems (Robins, 2000). IIR 'focuses on longer term search processes and tries to especially focus on breaking the separation of a search into a user and a system' (Singer et al., 2012a, p.90). Robins already mentions four models of IIR, which add dynamic components to classic IR. Each model puts a different focus on IIR. The term 'exploratory search' is also related to IIR, as it also describes a long term interaction between a user and a system with the goal of collecting information, but it 'focuses stronger on the discovery aspect in Search. Synthesis and aggregation are not explicitly covered in IIR.' (Singer et al., 2012a, p.90).

While the concepts of IR and IIR provide some sort of a basis for search support and model human interaction with the respective systems, the concepts of personal information management (PIM) or personal knowledge management (PKM) model the interaction with one user's personal information or knowledge space. These concepts can potentially be used to organize information that emerges during complex search processes. The concept of PIM describes the way how persons organize their personal information. This is not necessarily limited to search results, but contains all resources that contribute to the personal information space. Many approaches have been developed for unifying and accessing the personal information space. One of these approaches is presented by (Cutrell & Dumais, 2006), who claim that 'it is difficult for users to unambiguously specify what they are looking for, even in their own collections' (Cutrell & Dumais, 2006, p.50). The authors describe 'Phlat UI', a system which integrates all sorts of data on a personal computer and allows keyword and metadata search, as well as active tagging and enriching resources with metadata. PKM extends PIM as it integrates knowledge management into the plain PIM repository of resources. PKM focuses more on 'helping individuals to be more effective in personal, organizational, and social environments' (Pauleen, 2009, p.221). The scope of PKM is therefore more human centered than the document centered

PIM, even if the borders between the two concepts are fluent. Nevertheless, both concepts try to provide one user with a possibly comprehensive data collection that is easily accessible when being needed.

The presented concepts of PIM and PKM deal with providing access to large information collections, and they are explicitly bound to a single user's scope and try to be as comprehensive as possible. This implies that core ideas of these concepts may be relevant and interesting for supporting single user complex search, but the concepts themselves are not transferable between users.

The previous concepts focus on accessing a single user's collection of information, e.g. by enriching it with metadata. Other concepts for easing access to information are related to the idea of the enriching web resources with metadata. This concept is known as the Semantic Web. The core idea is to enrich all available data of the Internet with metadata and to 'bring structure to the meaningful content of web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users' (Berners-Lee et al., 2001). The Semantic Web therefore describes the idea or the goal of enriching all data with metadata to connect it more densely, and make it readable by both machines and humans. One of the core technologies enabling the realization of the Semantic Web is the 'Resource Description Framework' (RDF)[34]. The core idea of the RDF is that statements are made about resources. These statements describe and connect pieces of information by triples consisting of two resources which are connected by a property. The triples are ordered, such that the first resource (the subject) is connected via the predicate to the second resource (the object)[35]. Both resources and properties have to be identifiable by a Uniform Resource Identifier (URI), which allows tracing resources and predicates to their respective origin. Even if RDF is just a basic technique, which was elaborated more and more during the last years, its potential impact cannot be underestimated. Connecting resources via RDF triples leads to a more and more connected web of interlinked resources with machine-readable relations. This leads to a web of linked resources which in the best case enables semantic queries over a multitude of resources in different locations. For this web of semantically connected resources, the term 'Linked Data' was coined by Tim Berners-Lee[36]. Linked data basically relies on four criteria to be fulfilled to create a linked web. These four criteria are[37]:

1. Use URI's as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL).
4. Include links to other URI's so that they can discover more things.

[34] Retrieved May 22, 2015 from http://www.w3.org/RDF/
[35] Retrieved May 22, 2015 from http://www.w3.org/TR/rdf11-concepts/
[36] Retrieved May 22, 2015 from http://www.w3.org/DesignIssues/LinkedData.html
[37] Retrieved May 22, 2015 from http://www.w3.org/DesignIssues/LinkedData.html

These in fact simple criteria have the potential to interlink and to semantically enrich the existing web. Unfortunately, 'a surprising amount of data isn't linked in 2006', as Berners-Lee states in his article[38].

Small steps towards this goal can be taken by enriching small pieces of data by semantic information. This idea is not new, as a publication by (Malone et al., 1987) shows. The authors suggest enriching e-mails with machine-readable blocks of metadata that ease the composition and the processing of e-mails. A newer development of these semantic entities in the Web is called microformats. Microformats are basically additional HTML markups which contain semantic information (Khare, 2006). They have the advantage over XML that in XML each element can only have a single tag name, while HTML allows asserting multiple class names (Khare & Çelik, 2006). An example is the extension of geographic data on a web page by semantic information. The code fragment in the upper part of Figure 12 would then be extended to look like the code fragment the lower part of Figure 12.

```
<div>GEO:
  <span>37.386013</span>
  <span>-122.082932</span>
</div>

<div class="geo">GEO:
  <span class="latitude">37.386013</span>
  <span class="longitude">-122.082932</span>
</div>
```

Figure 12: Example of enhancing geo information in HTML with microformat data.

This has the advantage that the provided HTML fragment could easily be identified as geographic information and web browsers could interpret this information in a meaningful way, e.g. by showing a map when hovering over information like this. Microformats exist for a wide range of applications, such as geographic information, social relations, calendar entries, or business cards (Allsopp, 2007). There are still new microformats emerging, such as the hLocation format to denote geographic references (Dumitriu et al., 2007). Similarly, there exist some convenience tools that try to open microformats to users, as they offer structured editing and help avoiding errors during the creation of microformats (Campoy Flores et al., 2006).

The problem with all these approaches is that they do not yet work in practice in a satisfying way. The approaches described are still too complex, cover only detail solutions, and cannot tell how the searcher came to specific information. If they would work out perfectly, a complex search could easily be supported by interlinked pieces of data that guide users from resource to resource by meaningful relations. The missing practical realization may be explained by the additional effort required to enrich existing data by semantic information, and the comparably low level of demand for such data. Tools for supporting complex search therefore cannot yet rely on this data. As such, the approach presented in my thesis relies on the user's navigation in the web and makes use of information the users encounter during their search processes. My approach

[38] Retrieved May 22, 2015 from http://www.w3.org/DesignIssues/LinkedData.html

does not rely on semantic data as to not limit the search universe for the user to the semantically enriched resources.

The concepts in this section are important for the foundations of supporting complex search, such as IR and IIR, or they describe technologies which have the potential to make an impact on the support of complex search. However, these approaches are not yet powerful enough to cope with the vague information needs. The next section presents several practical approaches that try to organize large collections of data or make such collections more easily accessible.

2.4.3 Approaches

The previous sections present a range of techniques for organizing information and a set of theoretical concepts for accessing information; some of them have already been instantiated. This section concludes the range of related material by presenting some more concrete approaches for structuring information during search processes which have been realized. One approach for structuring and organizing collections of information is content curation; I presented foundations of this approach in Chapter 2.2.4. This section presents some systems that provide technical support for content curation. Another approach for structuring and accessing large collections of information are facet based approaches, which ease access to large information collections. Facet based systems are sometimes mentioned in the context of exploratory search. Another approach for structuring information are web-based systems for taking notes, that arguably could help with organizing search result. In this context, I present Evernote and Google Keep.

Content curation has been introduced as a model of search in Chapter 2.2.4. The approach of content curation has recently been implemented several times. Early thoughts on content curation have been presented in (Bhargava, 2009), and some models for content curation have been presented in (Bhargava, 2011). Since then, the ideas of content curation have been taken up in academic publications. (Zhong et al., 2013) investigate the behavior of users on Pinterest and Last.fm, which can be considered content curation portals for web resources and music. The authors find that most users use content curation to curate their own resources, as a large share of users (85%) uses Pinterest 'as a personal collection or scrapbook, and only 48% of the population uses the site to display their content to others' (Zhong et al., 2013). A more in-depth look on content curation from multiple origins into one large repository was done by (Rotman et al., 2012). The authors define content curation systems as systems to collaboratively integrate existing content into a single repository. This approach is opposed to wiki-systems, as their goal is to collaboratively create content from nothing. The authors investigate the user interaction with the Encyclopedia of Life[39], a large-scale content-curation platform for biological topics. They identify a set of challenges that hinder curating existing content into a single repository, such as the use of different biological classifications, a lack of quality control, or conflicts of competencies. The authors furthermore present a set of ways to overcome these issues.

Even if Wiki systems seem to be opposed to content curation systems in their nature of collaborative creation, they can be used as a technical base for the collaborative integration of re-

[39] Retrieved May 27, 2015 from http://eol.org/

sources. Nevertheless, a lot of more specific content curation systems have emerged. Examples are the WordPress-based tool CurationTraffic[40], and a similar approach is followed by Curation Suite[41]. Paper.li[42] resembles the look and feel of a traditional newspaper for content curation. Among many other systems, Bundlr[43] integrates photos, videos, presentations, texts, tweets, and other resources into digital bundles. A detailed overview of a range of web-based information curation tools is given in (Voyloshnikova & Storey, 2014).

Some authors mention facet based systems in the context of exploratory search. Facet based systems need to prepare a database to extract meaningful properties from their data, which are called facets. These facets can then be used to access a data collection from several viewpoints. For example, data files cannot only be accessed by their name, but also by their file type, creator, creation date, date of the last change, size, access rights, or else. Video files can be accessed by their duration, resolution, used codecs, or other properties. Facet based access to the BSCW system has been realized in (Franken & Prinz, 2009). The drawback of this approach is that any database has to be preprocessed to be able to extract facets that go beyond standard metadata descriptions and that the diversity of objects limits the possible facets. In publications like (Russell et al., 1993), authors have presented approaches for calculating the cost of the automatic extraction of facets from large datasets. Several expert systems have been designed for specific applications or search cases, such as mSpace (schraefel et al., 2006). mSpace is a database for classical music, combining information about composers, musical pieces, and musical periods. It is accessible by a faceted interface, which explicitly allows exploring the data collection by serendipitous browsing from one topic to the next, therefore enforcing exploratory search. While users like the concept, the effort required for establishing the database is very high. (Hearst, 2006) compares the clustering of results with the efforts needed for establishing faceted categories and finds that clustering is a somehow quick and unreliable approach. Establishing faceted categories might work well for smaller data collections, but require too much work and are therefore not useful for large or unlimited collections like the Web. A further drawback of faceted classifications is that the relevant categories all have to be known in advance. If a further interesting category appears, it is not supported unless the relevant facet data is extracted and established in the underlying database.

Besides these approaches there exist several web-based approaches that support note-taking and do not have a strong focus on content curation or other approaches. Examples for such systems are Evernote[44], Google Notebook, and Google Keep[45]. Evernote turned out to be the most common tool the participants of my second study used for taking notes and organizing web search results. Even if it was the most popular tool, Evernote was just used by 2 of 29 participants. Evernote allows collecting and writing pieces of information in a graphical user interface, as well as connecting to other users and discussing with them. The main focus is on clipping resources from the web, which is supported by a desktop browser extension and several mobile applications, which allow sharing the same resources among different devices. A similar approach is followed by Google Keep. This service was started in 2013, as a follow up service to

[40] Retrieved May 27, 2015 from http://curationtraffic.com/
[41] Retrieved May 27, 2015 from http://curationsuite.com/
[42] Retrieved May 27, 2015 from http://paper.li/
[43] Retrieved May 27, 2015 from http://bundlr.com/
[44] Retrieved May 28, 2015 from https://evernote.com/
[45] Retrieved May 28, 2015 from http://www.google.com/keep/

Google Notebook. Google Notebook started in 2006 and was discontinued in 2011. It allowed saving and organizing pieces of information during online research and sharing them with other users, may it be by publishing to all or by sharing within a specific community. Google Keep is based on the mobile Android OS or on Chrome OS and allows the clipping of images, texts, creation of lists, or the sharing of notes. In comparison to Google Notebook, Google Keep seems to have reduced its focus on note-taking and now acts more as a clipping and organizing tool, allowing also the creation of to-do-lists, reminders, or sorting of notes. As it is available as an application for mobile devices and also runs in the Chrome desktop browser, its focus is also more on supporting mobile interaction. Besides these three systems, there exist some more note-taking systems, such as Microsoft OneNote[46], Intellinote[47], or Circus Ponies Notebook[48] for iOS users, beside others.

All the presented techniques, concepts, and approaches can be used for supporting single issues of exploratory and complex search in a way, and the realized software solutions may come in handy in several situations when performing complex search tasks. All note taking tools for example can be used to organize and structure web search results, but none of the systems allows keeping the context where specific results have been found, or tracks visited websites where no valuable results have been found. Generally, all presented approaches focus on specific details, such as enriching specific types of data with metadata, or focus on the results of a search process rather than on the process itself. My system therefore follows a more generalized approach, leaving detailed solutions out in favor of a lightweight approach for supporting asynchronous, collaborative, complex search. In order to evaluate the origins of search results and investigate the search process itself, search logging systems have to be used, which I present in the next section.

2.5 Related systems

The previous subchapter presents models, techniques, and approaches which can support small aspects of search processes, as well as the interaction with and the organization of search results. These approaches are insufficient, and I explain how my approach differs from them. This subchapter presents approaches towards investigating the users' interaction with the Web and supporting collaboration during web search. As exchanging search trails for supporting collaborative search has never been done before, two main fields of related work exist. The first is logging the user's interaction with the Internet during search processes, while the second is supporting collaborative search. Therefore, the first section of this subchapter chronologically presents examples of search logging systems, while the second section chronologically presents examples of systems that support collaborative search.

Logging the user's interaction with the Web provides valuable insights about the search process. It reveals where users have found valuable information and where not. Search logs also provide a sound base for research on search strategies and on exploratory or complex search

[46] Retrieved May 28, 2015 from http://www.onenote.com/
[47] Retrieved May 28, 2015 from https://www.intellinote.net/
[48] Retrieved May 28, 2015 from http://www.circusponies.com/

tasks. For these reasons, several search logging tools have been developed, which I present in the next section.

Other tools focus on supporting collaborative search, may it be synchronous or asynchronous. The second part of this section gives an overview of several systems for supporting collaborative search. Both parts of this section conclude with an overview table, condensing the most important properties of all the presented systems. These overview tables are evaluated in detail in Chapter 3 to form requirements for the system developed in my thesis.

2.5.1 Search logging systems

Several researchers developed systems which log the user's interaction with the Web to be able to further investigate these results. Even if all presented systems for logging search behavior share similar goals, they may differ in some points. Some logging systems log all interaction with the user's operating system; some just log the interaction with the browser. Some systems may reveal the logs to the user, others may not. Finally, some loggers may have been made available for other researchers; others may stay with their respective inventors and are not publicly available. It is important that all these approaches go beyond query analysis, as they try to gather more information and feedback from the user. Query log analysis was e.g. used by (Broder, 2002) to gain an understanding of the different types of search tasks. I now present several search logging systems, ranging from early approaches in 1998 to more recent applications in 2013.

The first part of this section introduces several different systems for logging search, while the second part of this section presents an overview table. This table helps identifying core solutions and features of all the presented systems.

2.5.1.1 Examples of search logging systems

An early example for a system logging the search activities of a user is WebTracker (Turnbull, 1998). WebTracker was used to collect data to develop the behavioral model of information seeking on the web, as presented in Chapter 2.2.4 (Choo et al., 1998). WebTracker is a Windows 3.1 application that monitors the Web browser and collected menu choices, button bar selections, and keystroke actions. It was designed this way to overcome weaknesses of previously used, more complicated methods like proxies for gathering interaction data of the user with the Web (Pitkow & Recker, 1994). These methods were too inaccurate, as they just tracked the URLs of the resources, but could not track the user's interaction with the web pages.

WebTracker tracks the user's interaction with the browser, the visited URLs, and stores the search logs locally on the user's hard disk. This results in a search log containing the following entries for each user action: User ID, browser action, the time and date, URL, and web page title. The tracking of mouse clicks, scroll bars, and more complicated menu functions is not implemented. After activating the system and entering a user ID, WebTracker is completely invisible to the user, therefore being unobtrusive. The logs are stored locally on a daily base and can be accessed by the users, as they are plain ASCII text files. WebTracker was used to determine the participants' moves in the Web and to classify the moves according to the six categories of starting, chaining, browsing, differentiating, monitoring, and extracting information. Interviews with the study participants were conducted to extend the gathered information from the interaction logs. WebTracker is a very early example of a logging system, collecting data at a very low level of user interaction.

Technically similar to WebTracker is The Curious Browser (Claypool et al., 2001). Instead of developing an application for an operating system, the authors implement a new browser, which is able to collect all interaction data. This logger was developed to find out implicit interest indicators on web pages. Explicit interest indicators can be detected by asking the users or by monitoring the active selection of buttons or links on a web page. Implicit interest indicators can be inferred from the user's actions, such as mouse movements or stay times. The Curious Browser was implemented to resemble the user experience of Microsoft's Internet Explorer, with the difference that each time the user left a web page, a box popped up, asking the user how interesting the page was. For each web page visited, The Curious Browser stores the user name, the URL, the time and date, the explicit rating of the website, and some implicit interest indicators. Adding to this feedback, the browser also captures mouse clicks and the number of milliseconds the mouse was moving on the page, as well as the scrollbar activities and the keyboard activities.

The Curious Browser does not reveal the logs to the users, and the authors do not mention whether the logs are stored on the users' hard drive or online. Despite being able to capture extensive search sessions, the focus of The Curious Browser is more on the user's interaction with single pages. The granularity of the captured information was chosen to be able to identify interesting pages rather than for evaluating large-scale navigation histories. Unfortunately, there is not much more information available on this search logging approach. Nevertheless, The Curious Browser is an approach to capture low-level interaction and navigation data by the high effort of programming a new browser. The authors find that the time spent on the page, the amount of scrolling on the page and the combination of both have a strong positive relationship with the explicit user's interest in a page.

A newer approach for logging web interaction is presented by (Reeder et al., 2001). The authors present a framework for performing web usability studies, which consists of two components. Their first tool is named WebEyeMapper and is a physical device that captures the participant's eye movements during web usability experiments. The authors' intention is finding out which HTML elements attract the user's attention. As the computations mapping the eye tracking data to the HTML elements are time-consuming with the technical equipment from 2001, the authors need a tool that captures the course of the user's web search. WebLogger is used to capture the user's interaction with the web. After the search session is finished, a replay of the session can be computed, highlighting the points of interest on every website. WebLogger is based on Microsoft's Internet Explorer for Windows NT and captures and caches all significant user and browser events. The captured data includes the websites themselves, the changing of the websites, the portion of the page displayed, the position of the site on the screen, as well as user events, such as clicking links, or opening tabs. The search logs, as well as all displayed data are stored on the user's hard disk, but without revealing these logs to the user.

WebLogger and WebEyeMapper are two components of a framework for user experience studies. It is interesting that the authors of the tools could access the interaction with the Internet

Explorer and are not forced to rely on self-written browsers. As WebLogger is a subsidiary tool, not much information is revealed about its inner workings.

In 2004, the system WiIRE (Web Interactive Information Retrieval Experimentation System Prototype) was published by (Toms et al., 2004). Similar to the user experience evaluation framework of WebLogger and WebEyeTracker, the system WiIRE is intended to be a framework for conducting interactive information retrieval (IIR) experiments via the Internet. WiIRE is realized as a server based application, running on a Microsoft IIS server. The participants of the respective study can log in via the Internet to the WiIRE IIS application and use it to get access to the requested web resources. This way, WiIRE has full access to the user's interaction with the web, and stores the search logs on a connected Microsoft Access database, leaving the participant's computers untouched.

As WiIRE is intended to be a general purpose framework for IIR, it allows to be configured with experiment configurations for the concrete instantiation of IIR experiments. Possibilities of customization for IIR experiments include consent forms, initial demographic questionnaires, tutorials, instructions, pre-task questionnaires, intermediate and final questionnaires, or thank you messages (Toms et al., 2004, p.664). The actual search interface of WiIRE was specially constructed for IIR tasks (cf. Figure 13). The top bar displays the search task and is used to control the experiment, e.g. by clicking the 'I'm Done' button when the searcher finishes the experiment. The search interface in the lower section is divided into two sections. The left section contains the query input field and displays the results, while the right section provides tools to assist during query formulation.

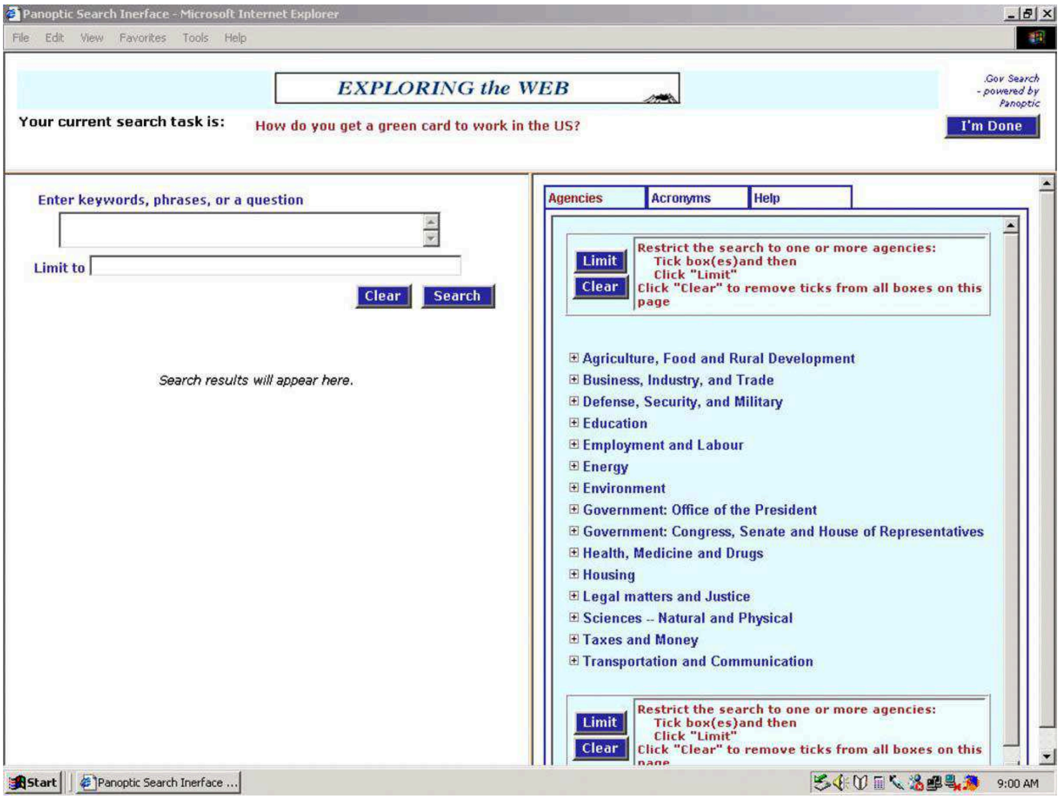


Figure 13: Example of the WiIRE search interface (Toms et al., 2004).

Similar to The Curious Browser, the authors of WiIRE in fact implement a new browser and control the access to it by providing this application as a server-based application. As WiIRE is intended to be a framework for IIR experiments, the authors provide details on the value of WiIRE for experiments and present a validation of the tool. Unfortunately, they do not reveal many details of the logging mechanism itself.

The previously presented systems are rather focused on small scale interaction with web content, such as interest rating or the identification of HTML content attracting the eye of the searcher. In opposition to that, WiIRE is one of the first systems aiming for interactive information retrieval (IIR) and focuses on rather large scale web interaction. As it is based on a custom developed browser, the interaction with the Internet may not feel as natural for the users as with search logging systems that manage to utilize a 'real' browser for such tasks. Nevertheless, the strength of WiIRE lies in providing a framework which allows setting up IIR studies easily.

One of the first web logging systems realized as an extension for the Internet Explorer is presented by (Fox et al., 2005). This research is comparable to the approach of (Claypool et al., 2001), who developed The Curious Browser to discover implicit interest indicators. Fox et al. developed an extension for the Microsoft Internet Explorer to find out whether there is an association between explicit ratings of interest and implicit measures of user interest. The Internet Explorer extension was developed using C# and was installed on the client machines. The extension collects both explicit and implicit feedback on the visited pages and sends it back to a remote SQL database using different XML envelopes for the different types of feedback. Explicit feedback is gathered in three ways. When a site is left, the user is asked whether the resource is liked, not liked, or the user found it interesting, but needs more information. When a search engine query is changed or refined, the user is asked whether the search is new or if the query changed, to find out if the intention changed. When the search session ends, the user is asked about the overall value of the session. Implicit measures are gathered during the user's interaction with the browser. Implicit measures include a number of different data, such as the time spent on the page, the amount of scrolling, the time until the first click, the number of clicks, the number of visits to a web page, or whether the web page was added to the favorites. It is the authors' intention to find the most meaningful implicit factors to explain the user's explicit ratings. Overall, 146 Microsoft employees participated in the study, who used the data collection tool for about six weeks. It is remarkable that no fixed search tasks were given to the users.

The authors find that a proper combination of the right implicit measures can predict the explicit satisfaction ratings well (Fox et al., 2005, p.166). It turns out that the click through on a website, the time spent on a site, and the way of exiting a site are good implicit predictors for explicit satisfaction. Similar to (Claypool et al., 2001), this study has the goal of identifying interest indicators for a web pages during actual search behavior. Even if this study is not focused on complex search, it can provide insights into the architecture of an early search logging tool.

An early study which investigates the user's web search behavior is presented in (Jansen et al., 2006). Jansen et al. describe a system named 'Wrapper', which is intended to analyze the exploratory search behavior of a user. The authors claim that 'exploratory searching is indeed a chaotic process' (Jansen et al., 2006, p.1), which spans multiple systems and multiple episodes of searching. In order to cope with this complexity 'the evaluation should be on the process, not a single system' (Jansen et al., 2006, p.1). Therefore, the presented approach not only logs the interactions of the user with the browser, but also with the operating system, such as the use of the clipboard, printing pages, saving pages, etc. To achieve this, the Wrapper framework comes as a Windows application that is installed and gets started before the search session begins. Wrapper stores all user interaction and sends it to a remote server every five seconds. However, as the authors do not reveal much information about the information collected, it seems to be a time stamped log of the visited URLs together with the interactions with the operating system. Figure 14 shows the visible part of Wrapper on client side, with the field for additional information that is either gathered from the system directly or entered by the user.

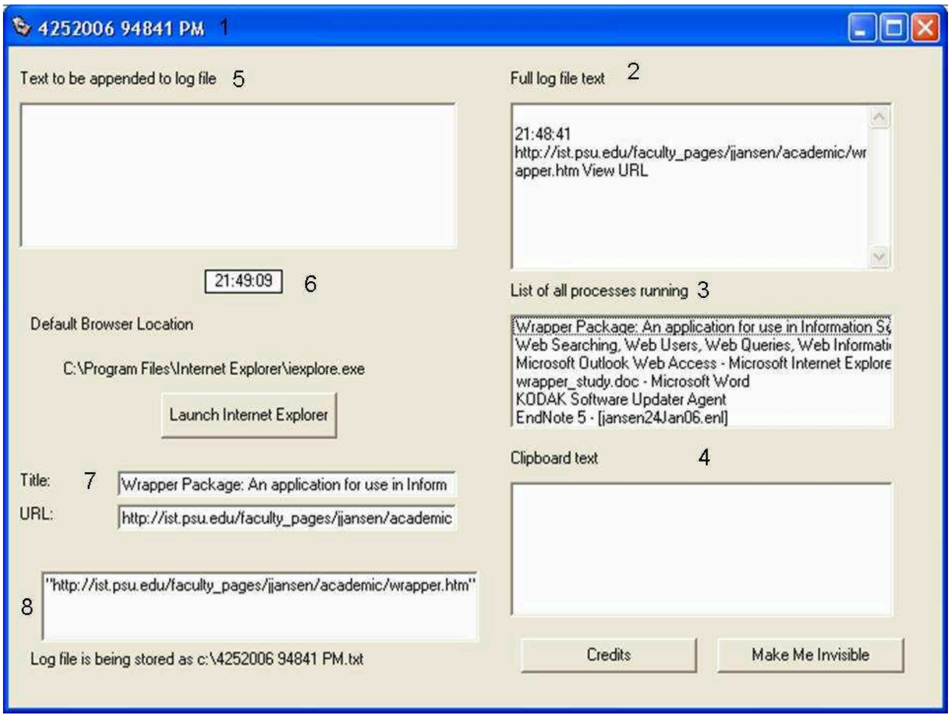


Figure 14: Example of the user interface of the Wrapper-system (Jansen et al., 2006).

The authors test Wrapper with four participants over one full week of searching. They find that 'searchers employ a variety of searching and information sources and return to topics over multiple days' (Jansen et al., 2006, p.4). Wrapper is one of the first search logging frameworks that explicitly tries to gather data on exploratory search behavior. It goes beyond the other approaches in that it also monitors the interaction with the operating system.

In opposition to the trend of the growing complexity of search logging tools, researchers were also interested in the analysis of query logs, as (Clough & Berendt, 2009) shows. This workshop on the ACM SIGIR Forum brought together a number of researchers interested in the analysis of query logs for identifying open questions. These questions include the open availability of log data, the connections between low-level actions like queries and high-level cognitive models, and combination of multiple data sources, such as log data and social data from

Facebook or Twitter. For these purposes, the ‘Lemur Query Log Toolbar’[49] is one tool used for logging queries on client side, being technically a Firefox browser extension.

Similar to the WiIRE system presented earlier, the HCI browser is introduced by (Capra, 2009) as a configurable tool for conducting batch studies on web information seeking behavior. It consists of a Firefox browser extension which collects browser event data, records answers, and allows the configuration and collection of data from pre- and post-task questionnaires. The browser extension has the benefit of enriching a standard browser’s functionality, therefore providing a familiar browsing experience, while being able to record a wide range of user generated events. The extension is also able to intervene with the user’s actions, as it is able to close tabs before new search tasks are started.

Keeping in mind that the HCI browser is described as a framework for conducting search experiments, the authors explain how the HCI browser is configured individually for each experiment session. In order to customize the HCI browser for different application cases, four configuration files are loaded, which define the introduction text, the set of tasks to solve, the pre-task questionnaire, and the post-task questionnaire. Each questionnaire configuration file allows the definition of a full questionnaire with multiple types of questions, such as multiple choice questions, or Likert-type questions. The HCI browser logs a wide range of events, such as the pages loaded, the clicked links, window and tab focus changes, opening and closing of windows and tabs, clicks on the back / forward buttons, the visited URLs, scrolling, and the users activities in the bookmark menu. This results in a log file like in Figure 15, showing a UNIX timestamp in milliseconds, date, time, session number, participant number, task number, action, and the visited URL.

```
1248198386334 21-7-2009 13:46:26 S4 P27 T1 intask LoadCap http://www.google.com/firefox?client=firefox-a&rls=or
1248198386398 21-7-2009 13:46:26 S4 P27 T1 intask Focus http://www.google.com/firefox?client=firefox-a&rls=or
1248198388384 21-7-2009 13:46:28 S4 P27 T1 intask LClick http://news.google.com/nwshp?client=firefox-a&rls=org
1248198389498 21-7-2009 13:46:29 S4 P27 T1 intask Scroll http://news.google.com/nwshp?client=firefox-a&rls=org
1248198389580 21-7-2009 13:46:29 S4 P27 T1 intask Focus http://news.google.com/nwshp?client=firefox-a&rls=org
1248198390167 21-7-2009 13:46:30 S4 P27 T1 intask LoadCap http://news.google.com/nwshp?client=firefox-a&rls=org
1248198401531 21-7-2009 13:46:41 S4 P27 T1 intask Scroll http://news.google.com/nwshp?client=firefox-a&rls=org
1248198411929 21-7-2009 13:46:51 S4 P27 T1 intask RClick http://news.google.com/news/url?sa=t&ct2=us%2F1_0_s_0
1248198413706 21-7-2009 13:46:53 S4 P27 T1 intask AddTab about:blank http://news.google.com/nwshp?client=firef
1248198413712 21-7-2009 13:46:53 S4 P27 T1 intask Info Currently open windows and tabs are logged below, but
1248198413718 21-7-2009 13:46:53 S4 P27 T1 intask Info Window=0, tab=0, url=http://news.google.com/nwshp?cli
1248198413724 21-7-2009 13:46:53 S4 P27 T1 intask Info Window=0, tab=1, url=about:blank http://news.google.c
1248198415971 21-7-2009 13:46:55 S4 P27 T1 intask Scroll http://news.google.com/nwshp?client=firefox-a&rls=org
1248198418755 21-7-2009 13:46:58 S4 P27 T1 intask LoadCap http://www.latimes.com/news/nationworld/nation/la-fg-
1248198419965 21-7-2009 13:46:59 S4 P27 T1 intask SelTab http://www.latimes.com/news/nationworld/nation/la-fg-
1248198420088 21-7-2009 13:47:00 S4 P27 T1 intask Focus http://www.latimes.com/news/nationworld/nation/la-fg-
1248198422564 21-7-2009 13:47:02 S4 P27 T1 intask Scroll http://www.latimes.com/news/nationworld/nation/la-fg-
1248198433480 21-7-2009 13:47:13 S4 P27 T1 intask submittedAnswerURL http://www.latimes.com/news/nationworld/n
1248198433488 21-7-2009 13:47:13 S4 P27 T1 intask submittedAnswerText The rare total eclipse will be visible th
```

Figure 15: Example log data of the HCI browser-system (Capra, 2009).

The HCI browser is a more modern approach for providing a framework for exploratory search studies. It was published several times, while its real impact in user studies was less overwhelming. After its first publication in 2009 it was published as open source software. However, its development seems to have stopped in 2011[50].

A system named ‘Search-Logger’ is presented by (Singer et al., 2011). This system is comparable to the HCI browser, as it is also realized as a Firefox browser extension. The Firefox extension monitors the user’s interaction with the Internet and sends this data to the PHP front-end of a database, which stores the activity logs. An analysis component is able to access the

[49] Retrieved June 01, 2015 from http://www.lemurproject.org/querylogtoolbar/
[50] Retrieved June 02, 2015 from http://ils.unc.edu/hcibrowser/

database and evaluate the results. The extension is visible in the browser as a little icon, which offers a menu after clicking on it. This menu allows the users to enter demographical data, and after this is completed the users may select a search task to work on. While Search-Logger is active, the icon flashes and records the generated data. Users have the chance to pause a search session and the recording of data accordingly. The logged data consists of implicit and explicit data. Implicit data can be e.g. the clicked links, opened or closed tabs, set bookmarks, started or paused search tasks, or clipboard events. Search-Logger also allows the recording of explicit data using questionnaires. Search-Logger is intended to be used in the context of IIR experiments. Participants work freely on a number of search tasks, questionnaires are filled out at the beginning of the experiment, after and before each search task, and at the end of the experiment.

Search-Logger was actually used in practice (Singer et al., 2011). A study was conducted that could prove the existence of complex search according to the definition from (Singer, 2012, p.17) as elaborated in Chapter 2.2.2. It is interesting to remark that Search-Logger was exclusively used for logging data, and did not provide feedback by revealing data to the users. This implies that the user's search process itself could not be supported or facilitated by information gathered during the search process. The similarities of Search-Logger with other search logging systems show that extensions for existing browsers in connection with web-based data repositories seem to have become some sort of standard approach for logging web search behavior.

An approach going beyond plain logging of search results is presented in (Danilov & Vainikko, 2013). This research is done within the context of (Singer et al., 2011) and also develops a Firefox browser extension, but is not exclusively aiming towards logging search activities. Instead, the tool named 'Search Excavator' tries to support users and tracks the opened tabs in the browser and extracts all text from them. The tool captures the text from the open tabs and sends it to a remote analysis engine. The analysis engine removes stop words and searches for similar words in multiple opened tabs. For these words, a frequency analysis is performed to find the infrequent words that appear on several tabs. The then discovered infrequent words are supposed to be meaningful, because they appear on more than one opened web page and most probably do not occur randomly. Search Excavator attaches an extra button to the Google search bar and offers search term suggestions that provide a meaningful extension of the query. This way, Search-Excavator provides user support by real-time analysis of search actions.

This approach is loosely coupled with the search logging approaches presented earlier, as it makes use of a similar approach like some other search loggers. It presents an interesting way of capturing data and using this data for calculating actual help in the form of recommendations for the searchers.

This section presents a number of search logging or related systems from 1998 until 2013. It is worth mentioning that the potential of the search loggers increases as the power of the web browsers increase. This explains the increasing use of browser extensions instead of complicated own developments, frameworks, or native applications. However, there is a surprising amount of frameworks, which were developed but not often used. Nevertheless, this overview of search logging systems shows their impact on research and raises the question how the data gathered during a search session can be used to support the individual searchers. The next section condenses the key properties of the presented systems.

2.5.1.2 Comparison of search logging systems

This section concludes the overview of search logging systems of the previous section. The comparing table helps working out significant features and differences between the systems. One of these properties comes from the literature of search logging systems, while I developed the other criteria during my literature review.

There are several criteria that work out key differences between the individual search logging systems and which can differentiate the approaches that have been taken to gather the users' interaction data. Regarding the technical setup, (Capra, 2009) suggests differentiating systems by **their data collection approach**. User data can technically be gathered in four ways:

1. Intercepting HTTP requests via HTTP proxies, which capture the requested URL and a timestamp. Proxies are limited as they cannot capture user interaction directly.
2. External monitoring programs are installed in the operating system and are allowed to capture the user interface and application specific events.
3. Custom web browsers mimic the functionality of an ordinary web browser, but offer the possibility to access all possible user interaction data, and even manipulation of pages while transferring them. This approach is powerful but requires a high development effort. Even if the custom browser stays close to standard browsers, the approach bears the risk of altering known standards and therefore altering user behavior.
4. Extensions to existing browsers build upon an existing web browser, which gives the extension control over and access to a set of functions. Although not being as powerful as a completely custom developed web browser, a wide range of information can be gathered for analysis, and the required implementation effort is significantly lower than with custom browsers.

During my literature analysis, I developed several other criteria which distinguish the existing approaches for logging the user's actions during search processes. These criteria are the following:

- The **universe** of sites that the search logging system is intended for. Does the search logger work on a local database can the users access the Web?
- What is the **captured data**? This is connected to the data collection approach, but even if full access to all data is available, the presented systems only captured what was necessary.
- Are the resulting **logs accessible to the users**? Or are they hidden on a server and just available to the researchers?
- Are the users **forced** to give **feedback** during the search process?
- Where is the **data stored**? Is it stored locally or on a central server?
- Does the search logging system only **capture the data**, or is the data analyzed during the user's search process to provide real time help?
- For what **purpose** has the search logging system been developed? Was it built for deducing interest indicators, or was its intention more towards recommendation?

Table 8 characterizes the presented search logging systems according to the given criteria. In some publications, not much information about the technical background of the systems is given. Therefore, some table cells cannot be filled with the information available. When systems capture much data, only the most important data types are presented. Some search loggers are presented as frameworks. For these systems, the range of features that are actually used in a study depends on the study setting. In these cases, the table entries focus on the functional scope of the framework. As all loggers are intended to log web resources, all are assumed to capture the visited URLs.

Table 8: Comparison of key features of the 11 presented search logging systems.

	Data collection approach	**Universe**	**Captured data**	**Logs accessible to users?**	**Forced feedback during search or questionnaires?**	**Unobtrusive**	**Data storage**	**Help user or just capture?**	**Purpose / Intention**
WebTracker	External monitoring application	Web	Menu choices, button bar selections, keystrokes	Yes, but not afforded	No	Yes	Local hard disk	Capture	Classify web moves of participants
The Curious Browser	Custom browser based on Microsoft Internet Explorer	Web	Ratings of pages, Timestamps, Clicks, Mouse move times	No	Yes	No	?	Capture	Determining implicit interest indicators
WebEyeMapper	Eye tracker and Internet Explorer instance	Web	Eye tracking data on web pages	No	No	No	Local hard disk	Capture	Framework for replaying browsing sessions to find HTML elements the user is interested in
WebLogger	External monitoring application	Web	Complete dump what the user has seen	No	No	Yes	Local hard disk	Capture	
WiIRE	Server based application	Web	Queries, search statements, clicked URLs	No	Yes	No	Remote MS Access database	Capture	Framework for interactive information retrieval experiments
(84, Fox et al., 2005)	Internet Explorer extension	Web	Queries, page ratings, interaction data with web pages	No	Yes	No	Remote database	Capture	Measuring implicit interest indicators
Wrapper	External monitoring application	Web	User interaction with browser and with operating system	No	No	Yes	Remote database	Capture	Prove the multiple session nature of exploratory search

Lemur Query Log Toolbar	Firefox Browser extension	Web	User queries, timestamps	No	No	Yes	?	Capture	Detecting search contexts from queries
HCI Browser	Firefox browser extension	Web	User interaction with browser, history, bookmark activity	No	Yes	Yes	Most probably remote database	Capture	Framework for conducting search experiments
Search-Logger	Firefox browser extension	Web	User interaction with browser, Interaction with browser features	No	Yes	Yes	Remote database	Capture	Tool to prove the existence of complex search
(65, Danilov & Vainikko, 2013)	Firefox browser extension	Web	Capture queries and words on selected pages	No	No	Yes	Remote database	Help user	Provide recommendations based on words on open pages

This overview compares the different search logging tools chronologically. It also shows that search logging systems have been used for capturing various types of data for various purposes. Unfortunately, most search loggers were never intended to be publicly available, or the code or the according web browsers were outdated when the implementation for the thesis started.

The overview table serves as an input for the development of requirements in Chapter 3 of my thesis. In this chapter, I evaluate all criteria in more detail with regards to the envisaged developments.

2.5.2 Collaborative search systems

While the previous chapter presents a number of search logging systems, this chapter presents a technical overview of systems supporting collaborative search. Some systems have been presented with more focus on the conceptual details in Chapter 2.2.5, such that this section focuses more on the technical details of the presented systems.

This overview of approaches for supporting collaborative search helps condensing this information into a comparing table. This table helps identifying core solutions and features that support collaborative search.

2.5.2.1 Examples of collaborative search support systems

One of the first approaches that can be considered to offer support for collaborative search is the CSCW3 browser (Gross, 1998). CSCW3 is a web browser prototype interpreting every website as a room, where other users can be in at the same time. Multiple instances of CSCW3 are aware of each other and inform their users if another user is visiting the same page. CSCW3 offers support for several types of browsing. Single users are supported by the ability of

CSCW3 to display pages and to capture the sequence of visited pages. Asynchronous collaboration is supported by exchanging bookmarks, group bookmarks, and comments on websites. Synchronous collaboration is supported by informing users when other users are in the same room, respectively on the same page. CSCW3 also offers support for chats between users on the same web page. It is also possible for users to guide other users by remote controlling their instance of CSCW3 or to glance at other user's navigation behavior, e.g. by monitoring which pages other users are currently watching at.

The focus of CSCW3 is on generating awareness for other user's actions and on providing means for collaboration during searching. It is remarkable that CSCW3 is not explicitly tuned towards information retrieval but already contains elements that point towards exploratory or complex search processes. Examples for this are the possibility of bookmarking websites or exchanging bookmarks, which already goes beyond the assumption of IR that one document perfectly matches the user's query.

Another approach for collaborative search is the Collaborative Information Retrieval Environment (CIRE), as presented in (Romano et al., 1999). This system aims towards supporting multi-user information retrieval. CIRE is a browser capturing the user's queries, sends them to AltaVista, and displays the results. CIRE automatically stores the visited pages, the queries executed, comments, and relevance feedback for the visited pages. This information repository is named the 'Information Retrieval Memory'. Users can share the queries executed, search results, and comments. Support for both synchronous and asynchronous search is offered. In contrast to CSCW3, CIRE explicitly focuses on information retrieval and offers additional collaborative features. The users perform single-user searches, add comments to the pages found, and publish them. During the authors' evaluation, it turns out that the collaborative features of CIRE are used only rarely, as the possibility of browsing team members' comments, history, or queries is frequently forgotten or ignored.

The Social Web Cockpit (SWC) is presented in (Prinz & Gräther, 2000). This approach focuses on collaborative knowledge management within virtual communities. The SWC allows collecting web resources as bookmarks and publishing those bookmarks to communities that may be interested in these resources. Similarly, the SWC detects the pages the user is currently visiting and points their attention towards communities that bookmarked the page. It is also possible to rate pages and to access other users' ratings. The SWC includes awareness functions that allow detecting the colleagues' online status or the creation of community specific vocabularies. The collected resources are collected as documents in the online collaboration system BSCW[51] and are available for all members of the respective group, also after the actual search session, allowing both types of concurrency and sensemaking. Even if the focus of the SWC is less on collaborative search and more on collaborative knowledge management, the system is an example for connecting group searches with online document repositories; a feature which is new to collaborative search systems.

During the late 2000's, more research was done on collaborative web search. 'SearchTogether' is presented as an interface for collaborative web search in (Morris & Horvitz, 2007). SearchTogether provides an online interface where users can log in and start and conduct search

[51] Retrieved June 08, 2015 from http://www.bscw.de/

sessions from there. SearchTogether is based on a client-server architecture, where the client is an ordinary browser and the server hosts the SearchTogether logic that organizes the user accounts, supports the search sessions, and stores the search results.

Figure 16: SearchTogether client features: (a) messaging, (b) queries of other searchers, (c) search result view, (d) recommendations, (e) (f) (g) search buttons, (h) page specific comments, (i) SearchTogether toolbar, (j) browser (Morris & Horvitz, 2007).

SearchTogether provides the searcher with an interface that connects several search-supporting features (cf. Figure 16). The interface supports asynchronous interaction by capturing the search query history or storing comments for pages. Synchronous interaction is supported by allowing instant messaging during search sessions or seeing the collaborators' search history grow. All search session results are stored on the SearchTogether server and can be shared by the collaborators. The authors are guided by three criteria which were developed using a survey on web search habits. These design criteria are awareness, division of labor, and persistence. Awareness is achieved by revealing other user's search activities, their queries, comments, and a viewing history of the pages. The authors claim that 'awareness of group members' search processes and findings can enable lightweight collaboration by reducing overhead' (Morris & Horvitz, 2007). Division of labor is indirectly supported by enabling instant messaging or by recommending pages to the collaborators. Another explicit approach of dividing labor is the 'Split search', which distributes the results of one searcher's query among all collaborators to evaluate the results collaboratively. Persistence is achieved by storing the session data on the SearchTogether server and making it available for later re-use. Users can therefore catch up to the state of an asynchronous search process by investigating their collaborator's search process history.

SearchTogether is one of the first systems that strongly aims towards supporting collaborative, complex search processes. The storing of session information and search artifacts in an online repository reminds of the Social Web Cockpit. It is not clear from the evaluation if the authors achieved their goals completely, as drawbacks were not reported. For example, some features seem to be intended well, but may turn out to be annoying. One example for that is the split search option, which clutters the collaborators' browsers, as collaborators can open new search tabs in other users' browsers.

A more practical approach for collaborative browsing is presented in (Amershi & Morris, 2008). This paper presents CoSearch and focuses on synchronous, co-located search processes. The system is intended for physical settings in which not enough computers are available to support each searcher, which leads to co-searchers standing around the main searcher's computer. The authors identify a set of problems that occur in these situations: difficulties in contributing to the search process, lack of the searcher's awareness of quiet group members' contributions, a lack of hands-on learning experience, differences in speed among the searchers, or the loss of found information. CoSearch enables the co-searchers to take part in the search process more actively. It achieves this by connecting several mice to a computer, which is claimed to be easily achievable in low-cost physical settings. Each group member then controls a separate mouse cursor in its own color. This allows the leading searcher to enter queries, from which the co-searchers can select results and append them to the web page queue, which holds the resources for further investigation. CoSearch allows adding notes to pages and sends the search results to all participators. Interestingly, both features are used only infrequently. CoSearch is extendable by integrating smartphones or other devices from co searchers via Bluetooth, allowing them to influence the search process even more, e.g. by entering queries.

Hardware-centric approaches like CoSearch are described rarely in the literature. While the authors report that the co-searchers liked to have more of a say in what is going on during the search, a real gain in the quality of the search process or search results is not investigated.

An early example of an approach which explicitly tries to support exploratory search is Cerchiamo (Golovchinsky et al., 2008). The authors identify the two roles of 'prospector' and 'miner' for collaborative search processes. While the prospector discovers potentially promising search directions, the miner digs deeper into the resources given by the prospector and tries to find interesting information. Cerchiamo explicitly supports synchronous, collaborative search independent of the spatial setting. The prospector's search interface allows searching and selecting documents for further investigation. These documents are then sent to a list of possibly relevant documents in the miner's user interface. While the prospector searches and makes relevance judgements for the results, the miner evaluates the documents and sends query term suggestions to the prospector. Based on the current search terms, the miner's list of documents gets updated based on their relevance. This means that when the prospector turns away from a query path, the according documents in the miner's resource list get ranked down.

Cerchiamo's approach is special with regards to strict roles the users have to act in during a search process. The user interface and the available actions differ depending on the selected role. In reality, some problems can occur with such systems. Users may switch between the different roles based on their experience with a sub-topic of the search process, or users may not have the time to work on a certain topic synchronously.

(Paul & Ringel Morris, 2009) mention a ‘recent emergence of tools to support collaborative web search’. During that time, several solutions focusing on more core problems of collaborative search emerge, such as sensemaking, or specific application areas, or video organization. CoSense is a system focusing on enhancing sensemaking for collaborative web search (Paul & Ringel Morris, 2009). The authors identify a set of challenges which complicate the sensemaking process. These challenges are missing awareness of what other searchers are doing, the missing sense of temporality in search processes, and a missing persistence of both the search process and the search products. Especially when it comes to temporality, participants of the study want to see a chronological ordering of all events in the search process, as well as the evolution of queries to build upon them. CoSense offers support for these three challenges by providing specialized views for each. The ‘search strategies view’ offers an overview of the other search members’ visited pages and therefore created awareness. The ‘timeline view’ provides a unified chronological representation of the actions of all team members. The ‘workspace view’ allows storing all products of sensemaking, such as to-do lists, decisions, notes, or comments (cf. Figure 17).

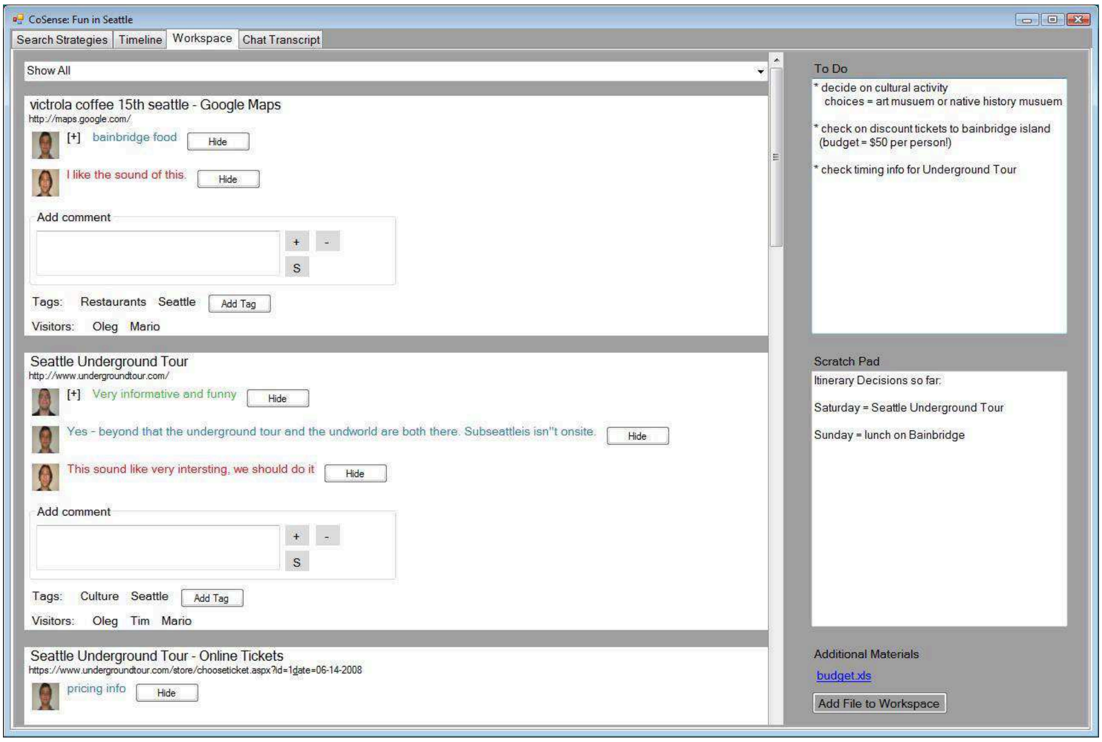


Figure 17: Workspace view of the CoSense system (Paul & Ringel Morris, 2009).

A system for asynchronous, remote, collaborative video search is based on the ViGOR system (Halvey et al., 2010). This is new because most research on collaborative search in multimedia systems concentrates on synchronous, co-located search cases. ViGOR focuses on the problem of collaborative categorization and searching in large video databases in an asynchronous way. Using ViGOR, videos can be annotated and grouped by user defined criteria, which makes this system unique. The developed system supports collaboration explicitly by storing search sessions which can be evaluated and continued by other users. It supports collaboration implicitly by allowing searching all groups of videos that were created previously by other users. Therefore, users are encouraged to reuse as much information from other users as possible.

Compared to plain collaborative search systems, the technical features of the ViGOR based system seem limited, but at least collaboration can be supported by ensuring the persistence of search results.

WeSearch (Ringel Morris et al., 2010) is another approach for collaborative search under special circumstances. WeSearch is an application being designed for a 1.2m by 1.8m large surface table, which supports multi-touch interaction by several users. It therefore aims towards synchronous, co-located, face to face collaborative search. The implementation of WeSearch supports up to four collaborators, each working on one side of the surface with a color-coded toolbar. Users are able to browse the Web and move their browser instances to the collaborators. They can extract pieces from web pages, which is enabled by parsing the Document Object Model (DOM) of the pages. A special 'clip search' automatically triggers a simultaneous search for normal results, news, and images on a given query. This way, labor can be divided between collaborators. A stream of the collaborator's actions is displayed in a specific 'marquee' region in front of each user. Furthermore, clips can be organized in user-defined containers and are automatically attached with metadata. The search results can be exported as XML files for further analyses. This feature set enables users to be aware of the collaborator's actions and to provide persistence to search results. The study shows that although the 'clip search' is intended to support division of labor, users do not use it that way. Labor is rather divided by communication and co-presence than by the use of technical features. With respect to sensemaking, it turns out that clips were useful, and that organizing the found clips is liked by the participants. Further tagging of the clips is used only rarely. WeSearch shows another approach of collaborative search solutions for special technical settings. It is interesting that participants take the line of the least resistance when it comes to more complex tasks like division of labor.

A more commercial approach is SearchTeam[52], which is a commercial online collaborative search support platform which combines search processes with content curation (cf. Figure 18). SearchTeam was launched in 2011. Users can create online search spaces on custom defined topics. The search spaces offer the possibility to organize the found resources into custom defined folders (top bar in the figure) and to add comments, links, or files to each of them. The search spaces can be shared with collaborators, which are displayed in the right sidebar. An activity toolbar informs about the collaborators' recent activity (right sidebar). Resources can be liked, and the commenting function allows communication among team members about the found resources.

[52] Retrieved June 12, 2015 from http://searchteam.com/

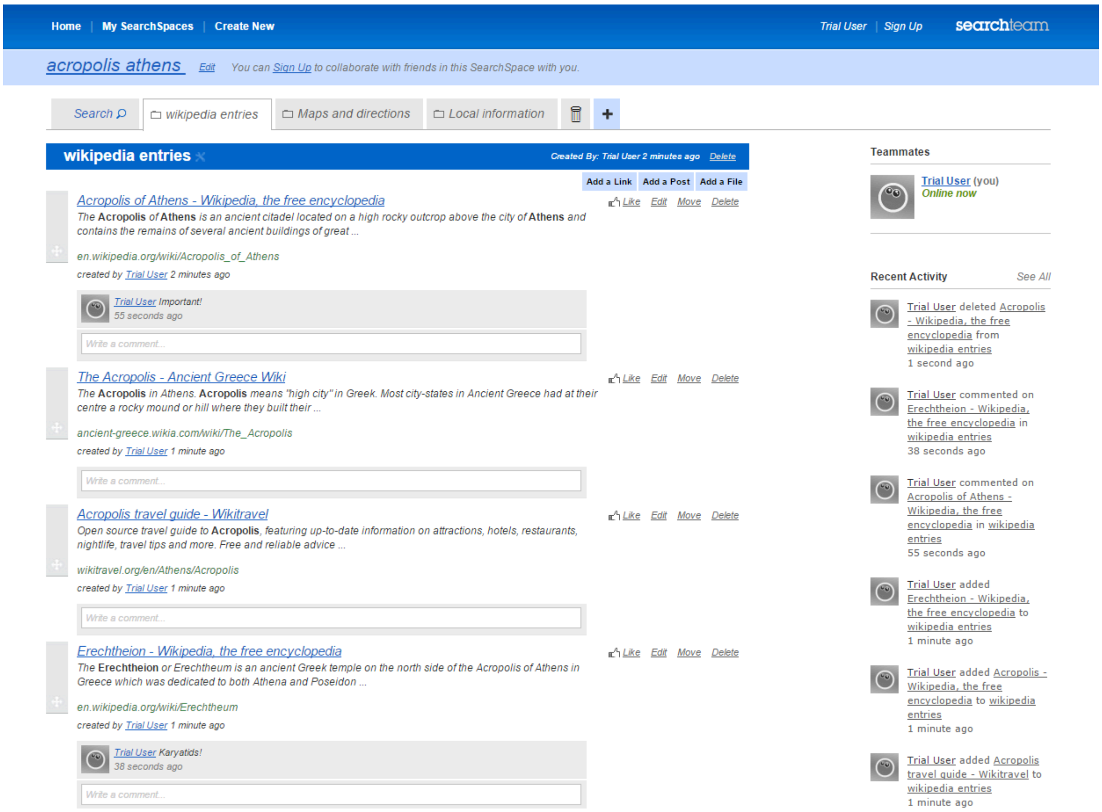


Figure 18: Example screen of the SearchTeam system[53].

SearchTeam is one of a few public collaborative search tools which still are accessible. It is mainly focused on the collection and storing of bookmarks. The integration of social networking services allows contacting e.g. the network of Facebook friends for new searches. SearchTeam does not support extracting and organizing relevant information from pages and therefore stays on the level of a bookmark collection.

Coagmento (González-Ibáñez & Shah, 2011) is explicitly intended for collaborative information seeking and is available as both a Firefox browser extension and a mobile app. While Coagmento was a stand-alone application in earlier versions, it was turned into a Firefox extension after reimplementation. Coagmento is visible in a separate tab during browsing, featuring the capturing of information snippets from web pages, collaborative editing on text content via the collaborative editor Etherpad[54], chatting, rating of information, or accessing an activity stream of the collaborator's actions. With these features, Coagmento allows synchronous collaboration, awareness of other users' actions, and persistence of found information. Similar to many of the already presented systems, the system focuses on collecting and working on valuable resources. The authors used Coagmento several experiments, but is not publicly available.

ResultsSpace is an approach which focuses more on asynchronous search in small groups. It is presented in (Capra et al., 2012) and in (Capra et al., 2013) and is based on preprocessed databases, where search tasks are performed with study participants. Users are able to search the preprocessed database, see the results, rate the results (relevant, maybe, not relevant), filter re-

[53] Generated with own input on June 15, 2015 on http://searchteam.com/
[54] Retrieved June 16, 2015 from http://etherpad.org/

sults, or access query histories. A key element of ResultsSpace is the rating function, which allows users to rate documents. In theory, a universe of ratings is built up in the database, which in the end help users to easily spot documents that are considered helpful by a large number of users. It is interesting that ResultsSpace also offers the possibility to rate a document ‘not helpful’. The ratings are displayed close to the found results on the results overview, and the users can filter the results by both its ratings and by the individual collaborators that have already seen the document. Compared to several other system mentioned earlier, ResultsSpace’s connection to content curation seems to be less prominent, as the database itself is seen as a repository of information which gets enriched by user ratings.

Similar to ResultsSpace is Querium (Golovchinsky et al., 2012), as it also relies on a pre-processed database to take its search results from. Querium supports multi-session exploratory search and relies on the ratings users have given to documents and is therefore a system for social search. The system is intended to accumulate metadata on the documents in the database, which builds up a universe of metadata that can help future users to cope with the amount of information. In order to provide this functionality, significant technical efforts have to be taken. A dedicated search server features two indices, one for document content and one for relevance feedback. A second database collects metadata for all documents. A third database stores the sessions and all actions and user inputs made during these sessions. This setup is able to supply the user with a number of functions, such as four different ways of issuing queries to the search index, offering reviews of documents, or providing overviews of activities in a selected search task. Furthermore, Querium offers the assessment of documents with ‘Like’ and ‘Dislike’, digging into the query history, faceted filters of search results, or retrieval histograms. These histograms show when a document was accessed and which queries returned the document as a result with what relevance over time. As this interface turned out to be very complex, a second design iteration of Querium aimed for simplification of the user interface. The architecture was changed to reduce server-side computation and the query interface was unified to offer only one type of query instead of four. Furthermore, the summarizing view of the session was united with several other elements of the user interface and the filter interface was redesigned to look more consistent and to provide more control.

Querium is an example of a very complex, heavyweight social search interface, following an approach of generating recommendations from explicit user feedback. The authors describe how they try to reduce complexity in the second version of Querium, but still the system remains heavyweight. It requires constant interaction of the users with the found documents to create a helpful metadata corpus for other users.

An approach that combines collaborative search with social elements is the web application So.cl[55] (Farnham et al., 2012). So.cl is explicitly designed as a lightweight combination of searching and sharing functions. So.cl allows its users to create a post on a self-defined topic, which consists of curated elements from web resources and user-written notes on the topic. The post can be made public, which makes it visible to the user community which in turn can comment on the elements in the post. The posts mainly consist of images, which lead to the origins of the images. Users of So.cl can follow other users and stay informed about their posts, which allows serendipitous interaction between users on topics. So.cl is closer to content curation than

[55] Retrieved June 17, 2015 from http://www.so.cl/

to exploratory search, as it picks out and connects valuable resources on a certain topic, leaving out the context in which results are found.

Similar examples of still available, web-based collaborative or social search systems – which can be commercial or free – are Diigo and Heystaks. Diigo[56] was initially launched in 2006 and is a commercial tool focusing on social bookmarking and annotating websites. Users can clip out specific parts of web pages and add comments to them. When annotations are made public, other users can discover them. Heystaks[57] was launched in 2010 and is a platform which analyzes individual search data to build search profiles. Heystaks calculates interest profiles from the search data and tries to match users with similar interests. Heystaks achieves this by being integrated e.g. in shop solutions, where it captures the actions of the users being logged in and calculates their interest profiles. Heystaks is an example of how tools can leverage the gathered search data without requiring explicit interaction from the users.

This section presents a broad range of approaches towards social and collaborative search, even if some of these approaches are similar to each other. Although this overview provides insights on a number of different systems, they can only be an excerpt of the approaches being developed over the last 20 years. The next section analyzes different aspects of all the presented systems and combines them into an overview table featuring the distinctive features of the presented approaches.

2.5.2.2 Comparison of collaborative search systems

This section concludes the collaborative search systems overview of the previous section. A comparing table helps to work out significant properties and features of the systems. Some of these properties come from the literature of collaborative search; some characteristics emerged from the analysis of the systems and are appropriate to work out the differences.

Considering collaboration, (Golovchinsky et al., 2009) work out four dimensions of collaboration for categorizing collaborative search systems. For a more detailed introduction on these dimensions, please refer to Chapter 2.2.5. These dimensions are:

1. **Intent**: explicit vs. implicit.
2. **Depth of mediation**: shallow UI-level vs. deep search engine level.
3. **Concurrency**: synchronous vs. asynchronous.
4. **Location**: co-located vs. distributed.

These dimensions of collaboration can be extended by the four basic aspects of collaboration, which were developed by (Kelly & Payne, 2014). For a more detailed introduction on the basic aspects of collaboration, please refer to Chapter 2.2.5. These aspects are:

1. **Awareness** of the collaborator’s actions.
2. **Division** of labor between collaborators.
3. **Persistence** of collaboratively generated results.
4. Support for collaborative **sensemaking** of found results.

[56] Retrieved June 17, 2015 from https://www.diigo.com/
[57] Retrieved June 17, 2015 from https://www.heystaks.com/

Table 9 summarizes the position of each collaborative search support system towards each of the first eight criteria from the literature review.

Table 9: Comparison of 8 key characteristics of the presented 16 collaborative search support systems.

	Intent	Depth of Mediation	Concurrency	Location	Awareness	Division of labor	Persistence of results	Sensemaking
CSCW3	Explicit	Shallow	Both	Remote	Yes	No	No	No
CIRE	Explicit	Shallow	Both	Remote	No	No	Yes	No
Social Web Cockpit	Explicit	Shallow	Both	Remote	Yes	No	Yes	Yes
SearchTogether	Explicit	Shallow	Both, Catch-up possible	Remote	Yes	Yes	Yes	Yes
CoSearch	Explicit	Shallow	Synchronous	Co-located	Yes	Yes	Yes	No
Cerchiamo	Explicit	Deep	Synchronous	Remote	Yes	Yes	Yes	Yes
CoSense	Explicit	Shallow	Synchronous	Remote	Yes	Yes	Yes	Yes
ViGOR	Explicit	Deep	Asynchronous	Remote	Yes	Not explicitly	Yes	No
WeSearch	Explicit	Shallow	Synchronous	Co-located	Yes	Yes	Yes	Yes
SearchTeam	Explicit	Deep	Synchronous	Remote	Yes	Yes	Yes	No
Coagmento	Explicit	Shallow	Synchronous	Remote	Yes	Yes	Yes	Yes
ResultsSpace	Implicit	Deep	Asynchronous	Remote	No	No	Yes	No
Querium	Implicit	Deep	Asynchronous	Remote	No	No	Yes	No
So.cl	Implicit	Shallow	Asynchronous	Remote	No	No	Yes	No
Diigo	Implicit	Shallow	Asynchronous	Remote	No	No	Yes	No
Heystaks	Implicit	Deep	Asynchronous	Remote	No	No	Yes	No

The aforementioned criteria directly originate from publications on collaborative search systems. Adding to these criteria, I developed several other criteria during my literature review process, which are appropriate for describing key features of collaborative search systems. These criteria are:

- Can the overall system be considered **lightweight** for any user of the system, meaning that the tool requires a low installation effort and low efforts for understanding and using its features?
- Does the system offer the possibility of **synchronous communication**?
- With which **technology** is the system realized: Is it a web browser extension, a standalone system, or another approach?
- Which collaborative **artifacts** are generated during the collaborative search process?
- Is there more **hardware needed** than an ordinary browser and PC involved or needed?

- Is the cooperation with **unknown partners** possible?
- Does the system capture the **temporality** of actions, e.g. the user's actions as a sequence and offer it to other users for analysis?
- In which **universe** does the system work? Is it set up on a predefined database, in a certain collaborative work environment, or can it be used for accessing the Web?
- Is it possible to **rate** the found **resources**, may it be positive or negative?

Table 10 answers these questions for all the presented systems and approaches, where possible. In some cases, the information given in the cited publications is not sufficient to answer all questions, or the information given in the publications does not address the questions explicitly. The entries in the table cells are not exhaustive and concentrate on the envisioned use of the systems. For example, all online systems are technically capable of being used by several users at the same time. However, systems that do not explicitly afford synchronous interaction by their user interface elements are not counted as synchronous systems. Ratings for example can be expressed by any commenting feature, if available. However, when a rating is not explicitly implemented, the system is considered to provide no rating options. Table 10 summarizes the position of each of the collaborative search support systems towards each of the nine self-developed criteria.

Table 10: Comparison of 9 key characteristics of the presented 16 collaborative search support systems.

	Lightweight?	**Synchronous communication?**	**Technology**	**Artifacts**	**Hardware needed?**	**Unknown partners?**	**Temporality?**	**Universe**	**Rate resources?**
CSCW3	No	Yes	Web browser	Bookmarks, Comments, vCards	No	Yes	No	Web	No
CIRE	No	No	Web browser	Comments, Ratings, History	No	No	Yes	Web	Yes
Social Web Cockpit	No	No	Client-side Java application	Bookmarks	No	Yes	No	Web	Yes
SearchTogether	No	Yes	Online interface	Search session object	No	No	No	Web	Yes
CoSearch	No	Yes, offline	Browser + Hardware	Links, comments	Yes	No	No	Web	No
Cerchiamo	No	Yes	Clients, Server with algorithmics	List of documents, ratings	No	No	No	Web	?
CoSense	No	Yes	?	To-Do lists, decisions, comments	No	No	Yes	Web	Yes
ViGOR	No	No	?	Categorized videos	No	Yes	No	Video database	No
WeSearch	No	Yes, offline	Surface table	Save table state	Yes	No	No	Web	No

SearchTeam	Yes	No	Online interface	Bookmarks, excerpts	No	Yes	No	Web	No
Coagmento	Yes	Yes	Firefox browser extension + mobile app	Collaborative notes, bookmarks	No	Yes	Yes	Web	Yes
Results Space	No	No	Special view on database	Ratings	No	No	No	Text database	Yes
Querium	No	No	Special database browser	Ratings	No	Yes	Yes	Text database	Yes
So.cl	Yes	No	Special database browser	Images, comments, bookmarks	No	Yes	No	Web	No
Diigo	Yes	No	Online interface	Web pages, comments	No	Yes	No	Web	No
Heystaks	Yes	No	Online interface	Search activities of users	No	Yes	No	Web	No

This overview shows central criteria for distinguishing collaborative search support systems and their development over time. The overview table serves as an input for the development of requirements in Chapter 3 of my thesis.

2.6 Related work for other parts of the thesis

The previous subchapters focus on several theoretical and practical aspects around search and collaboration. For the system being developed in this thesis to support discontinuous, asynchronous collaborative, complex search, some other aspects are of importance. While some technical aspects like the common approach to build web browser extensions for developing collaborative search supporting systems have been discussed already, others are still missing. Such open points are the ways of visualizing search related data and the mathematical measures for graph complexity.

2.6.1 Visualizing search related data

In cases when information captured during search processes is visualized for non-professional users, the type of visualization is relevant. Several studies have been performed to find out which types of visualization are most appropriate for certain application cases. Although this is not the main focus of my thesis, I provide a glance on the literature of visualizing search related data.

For visualizing the individual user's search behavior, an appropriate visualization has to be found. Studies have been performed that compare the value of textual representation of search results with 2D and 3D representations. (Sebrechts et al., 1999) could show that in terms of locating a resource in a given visualization, text interfaces outperformed 2D and 3D solutions for participants with low computer experience. For participants with high computer experience, the results for the 3D visualization were as good as the results for the textual representation, and the results for the 2D visualization were worse than both. These results indicate that plain text interfaces may lack in visual attractiveness, but are effective nonetheless. Similar results with respect to 2D and 3D representations of semantic information of database content in virtual spaces are reported by (Westerman & Cribbin, 2000), who claim that 'the amount of additional semantic information that can be conveyed by a three-dimensional solution does not outweigh the associated additional cognitive demands'. These results indicate that the power of textual representations of search results should not be underestimated, as they seem to afford scanning through results and therefore enable users to easily locate the desired results.

When it comes to large result sets, additional approaches of providing overviews of such data may be helpful. If the results share certain characteristics, approaches like clustering or faceting can help with the initial organization of the data (Hearst, 2006). Especially facets can help users as they 'promote recognition over recall' (Hearst, 2006, p.61), which helps users as they do not have to remember the results explicitly, but can relocate results in a larger set by evaluating the given facets. However, facets require a data set whose key characteristics are known beforehand to identify and populate the facets, which is not feasible for web search. Clustering is used in some systems' visual interfaces to structure the result sets, such as in the DOPE (Drug Ontology Project for Elsevier) browser (Stuckenschmidt et al., 2004). The DOPE browser is a special development for exploring resources on drugs and diseases. It provides support for thesaurus-based search and topic-based exploration of search results and makes intensive use of clustering result sets in a visual way (Fluit et al., 2006). In comparison with other 2D and 3D data visualization approaches, clustering is generally liked, and helps the participants identifing sets of documents to avoid and to keep track of relevant result sets (Sebrechts et al., 1999).

When it comes to the visualization of smaller result sets, it can be argued that more visual representation may be helpful for users to relocate known resources or to get an overview of a set of unknown resources. This could be achieved by using thumbnails, which are small images of websites, integrated into a visual interface. Surprisingly, thumbnails are shown to be of only limited value in these application cases. In a study, (Czerwinski et al., 1999) let participants find archived web pages by various clues. The participants have seen the web pages several months earlier and stored them in a spatial location memory. The results are presented to the participants after several months either by thumbnails or by blank squares as visual clues. It turns out that the presence of thumbnails does not improve the participants' performance over blank squares after an initiation phase. Similar results are presented in (Dziadosz & Chandrasekar, 2002), where the utility of thumbnails in web search result displays is evaluated. Users are asked to decide on the relevance of pages in cases where there was given only text, only thumbnails, or both text and thumbnails. It turns out that the user decisions with the plain text interface are more accurate than the plain thumbnail interface.

Comparing the plain text and the combined interface, users score a bit better with the combined interface. During the study described in (Kelly & Payne, 2014), participants work with the systems Diigo and Coagmento. Diigo stores web pages as textual descriptions, and Coagmento stores thumbnails and presents them to the users. Paradoxal, the participants using Diigo indicate a desire for thumbnails, where the users of Coagmento demand more textual descriptions of the stored pages. Summing up the evidence from this set of publications, thumbnails alone seem not to be helpful, and when used in parallel to textual descriptions, they can help a little. Despite the limited practical impact of the thumbnails, users tend to desire what is not available. They demand thumbnails when they are not available, and demand text when the thumbnails dominate the user interface.

When it comes to the visualization of connected URLs and user paths through them, graph visualizations are the most natural representations of such information. As long as the graph is dynamically extending due to ongoing user interaction, static layouts need to be laid out anew after each change of the underlying data structure. This may be cumbersome as the user would have to orient in the new graph after each change. These problems can be avoided by choosing force-directed graphs as a layout mechanism. Force-directed graphs are visualized in a floating manner, with virtual springs between the nodes, which make them readjust after every change without changing their look drastically. Force-directed graphs are described in (Eades & Huang, 2000) as being an approach for the visualization of large data sets and especially web graphs are mentioned for these purposes. (Eades & Huang, 2000) describe a rather complex architecture necessary to realize force-directed graphs, while recent JavaScript visualization libraries such as the 'Data Driven Documents'[58] framework are capable of such visualizations within web browsers at runtime.

The results in this section indicate that the textual representation of found search results and user generated data cannot be considered inferior to more visual approaches, and that force-directed visualizations are a means for visualizing navigation information.

2.6.2 Measures of graph complexity

The visualization of a graph helps the viewer to judge the complexity of the graph. While it is easy to judge relatively whether one graph is more complex than another graph, providing absolute measures is hard. Comparing Figure 19 and Figure 20 with respect to complexity, it turns out that the graph in Figure 20 is obviously more complex than the one in Figure 19.

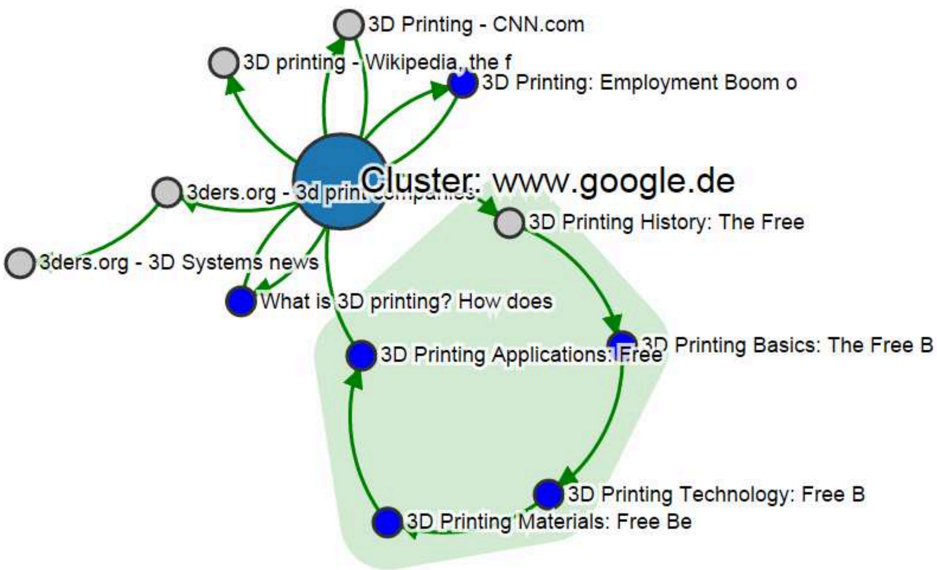


Figure 19: Less complex example search graph from the first SearchTrails user study.

[58] Retrieved June 23, 2015 from http://d3js.org/

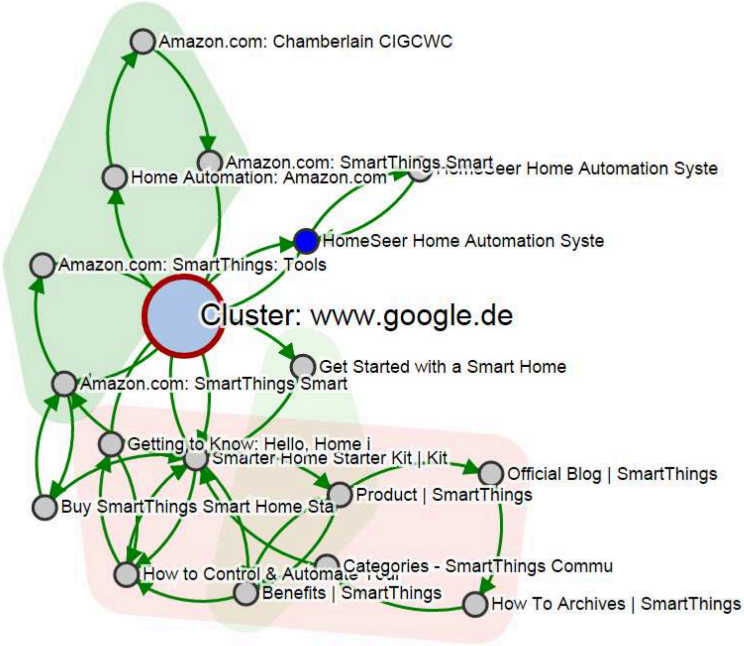


Figure 20: More complex example search graph from the first SearchTrails user study.

Several approaches have been made to calculate a graph's complexity in a mathematical way. One more recent approach in calculating the linear complexity of a graph was made by (Neel & Orrison, 2006), who mainly compute the graph complexity by a graph's adjacency matrix. The authors define graph complexity by the linear complexity of any of the graph's adjacency matrices. As to their definition, 'the linear complexity of a graph is a measure of how hard it is to compute AX, where A is an adjacency matrix of the graph and X is a generic vector of the appropriate size' (Neel & Orrison, 2006, p.4). In practice, the computation of the complexity required finding irreducible graphs by reducing graphs and calculating their complexity recursively, which was not turned into practice in the thesis for practical reasons.

Nevertheless, (Neel & Orrison, 2006) mention a publication by (Minoli, 1975), who computes the complexity of a graph by a formula involving the number of vertices, edges, and proper paths. As the different measures of complexity are not comparable anyways, and the complexity of the graphs in the thesis is always determined by the same measure, I chose to investigate Minoli's measure for my thesis. Minoli formulates four properties his complexity function fulfills (Minoli, 1975, p.651):

1. 'The complexity should be monotonically increasing on the number of vertices of the graph;
2. The complexity should be monotonically increasing on the number of edges of the graph;
3. The complexity should reflect the degree of connectedness of the graph; and
4. The complexity should satisfy our intuitive 'feel' of complexity, assigning a high number to a graph which 'looks' complicated, and vice versa.'

Minoli defines his measure of complexity for normal graphs, which are connected, undirected graphs with no multiple edges, no self-loops, and having more than two nodes. Proper paths between two different nodes are defined as paths that include each of the two nodes exactly once (as the initial and final vertex), and contain any particular edge no more than once (Minoli, 1975, p.652). Although the graphs constructed in the thesis are directed to resemble the user's paths between resources, I chose to omit the direction for the complexity measure, as it is not relevant for the perceived complexity of a graph, therefore fulfilling all of Minoli's criteria.

Based on these prerequisites, Minoli defines his measure of the complexity of a normal graph fulfilling the given properties as:

$$\chi(G) = \frac{ne}{n + e} \sum_{\substack{(v_i, v_j) \\ i>j}} \sum_{k=1}^{e} \gamma_k(i,j)$$

Where

- $\chi(G)$ is the complexity of a graph G = (V, E)
- V = $\{v_i$, i ϵ $\mathbb{N}\}$ is the set of vertices
- E = $\{(v_i, v_j)$, $v_i \epsilon V\}$ is the set of edges
- n = |V|
- e = |E|
- $\gamma_k(i,j)$ is the number of proper paths of length k from nodes v_i to v_j)

Based on this definition, the complexity of the given graphs can be calculated and returns a value of 3,761.3 for the less complex graph (Figure 19) and a value of 86,484.3 for the more complex graph (Figure 20). Obviously, the complexity measure and its respective computing time are influenced by the highly connected parts of the graph, as these parts exponentially increase the number of proper paths between two given vertices. Larger graphs therefore result in far higher complexity values. In order to cope with these exponential slopes of complexity when comparing them to other linear measures, the complexity values need to be logarithmized for such purposes.

2.7 Recent trends

Adding to the overview of related work and concepts, I want to prominently point out some recent research which is connected to the work presented in this thesis and which complements the work presented in the previous subchapters.

Recent publications and workshops show a steady interest in supporting web-based complex search tasks. These resources include a paper on CIKM 2014 (Awadallah et al., 2014), or a workshop on 'Supporting Complex Search Tasks' on the European Conference on Information Retrieval 2015 (Gade et al., 2014). Another workshop with the same title was held on the ACM SIGIR (Belkin et al., 2012). It is remarkable that despite the prominent use of the term 'complex search', only one of the publications takes an effort in defining complex search. (Awadallah et al., 2014) define complex search as 'a multi-aspect or a multi-step information need consisting of a set of related search tasks', thus neglecting the definition of complex search in (Singer et al., 2012a). For a definition of complex search the authors also refer to (Marchionini, 2006) and

(White & Roth, 2009), which both define exploratory search, thus implying a synonymous use of complex and exploratory search. Apparently, the term ‘complex’ is so far understood as a property of a search task, not needing a definition, or as being synonymous to the term exploratory search.

Some work has been done with respect to search trails and their analysis. A very recent example from Microsoft research is (Awadallah et al., 2014). The authors propose suggesting queries to users which belong to thematically related search tasks. In order to achieve this, the authors access the search logs of Microsoft’s search engine Bing. The search logs of single users are determined by their IP addresses, separated into sessions, and analyzed. When users are assumed to be engaged in complex search sessions the related search terms are used to build up a repository of common search tasks. These related search terms are then assembled to form a graph of connected search tasks. Searchers entering terms from these graphs receive recommendations for related search terms. This approach analyzes the trails the user leave on server-side to extract generalized information, and tries to generate added value for future users.

An earlier approach which is concerned with the users’ trails on client side is described in (Singla et al., 2010). It is interesting that the authors find that ‘individual items may be insufficient for vague or complex information needs’, and that there is a ‘value in trails’ (Singla et al., 2010, p.443). The authors from Microsoft research widely distribute an Internet Explorer extension, which captures time stamps and the visited web pages and send this data back to Microsoft. The captured data does not include explicit user feedback, such as specific highlights on pages, or other ratings of the content seen. This data is used to generate trails, which are evaluated with regards to length, breadth, depth, or diversity by a number of different algorithms. These algorithms determine the value of a search trail for a given initial query and try to find the most useful trail based upon these scores. The authors assume ‘that there is a best trail for each query-result pair’ (Singla et al., 2010, p.450). They claim that the paths scoring best ‘outperform the average over all trails followed by users’ (Singla et al., 2010, p.449) and therefore ‘deliver value to users over origins’ (Singla et al., 2010, p.447). It is interesting that the authors can show that search trails can provide value to users, although they do not reveal exactly how they evaluated this added value. The paper suggests that no study has been performed with human participants and that the described analyses are purely theoretical.

The papers presented here show a steady and recent interest in complex and exploratory search since its first definition in 2006 (Marchionini, 2006). The presented approaches are similar in that they gather massive amounts of data on server-side to extract some sort of common knowledge from, which is then again presented to other users. The drawback of these solutions is that they assume that there is one single best-fitting answer for a given search case. This approach provokes the existence of a filter bubble as described in (Pariser, 2011), as recommendations are generated from a dominating set of information. Current developments use search trails for recommendation rather than for collaboration. These approaches are still close to classic information retrieval, in that they try to show a way to reach a final information goal.

2.8 Conclusion and outlook

In this chapter, I derive insights for all parts of this thesis from an analysis of related work. The work in each subchapter supports and influences my own work. Until now, the idea of arbitrary tail through the Web has not yet been realized. I choose the definition of complex search to work with and support search processes modeled by the cyclic model of foraging and sensemaking. My approach values the importance of negative search results and enables the capturing and exchanging of negative search. Study results on the search behavior of real users suggest limiting the technical complexity of search support systems even for experienced users. Several obvious solutions for my approach, like bookmarking or PIM, are not suitable for solving the challenges of capturing search results as a whole, enabling the exchange of search results between collaborators, or resuming interrupted searches.

From the analysis of 11 systems for logging search activities and 16 systems for supporting collaborative search, I develop two comparing tables with self-developed and literature-derived criteria. I evaluate these tables in the following chapter to work out the desired characteristics of the envisioned system. I realize my approach with the help of force-directed graph visualization techniques and use graph complexity measures for the evaluation of search artifacts. Finally, my approach differs from recent trends of capturing and analyzing search trails as I value the individual search trail over generalized recommendations or server-side analyses.

Taking together all insights from this chapter, the main goal of my thesis gets clear: In my thesis, I develop and evaluate a system for supporting discontinuous, asynchronous, collaborative, and complex search. The system relies on the concept of capturing search activities of users and visualizing them as trails, which can be annotated and enriched by metadata. These trails are assumed to provide added value to other users, as they have the power to completely reflect performed search processes and reduce work load for the collaborator.

The insights of this chapter already help delivering parts of the answers to the research questions RQ1.1 ‘Search trails valid?’ and RQ1.2 ‘Which functions?’. My findings on the historical origins of search trails (cf. Chapter 2.1) and the recent developments towards search trails in search support (cf. Chapter 2.7) provide positive indication towards answering research question RQ1.1 ‘Are search trails a means to resemble the course of a user’s web search?’. Similarly, the comparing tables in Chapter 2.5.1.2 and Chapter 2.5.2.2 provide an overview of possible features and approaches for answering RQ1.2 ‘Which functions does a system need to offer to the users to support capturing both the context as well as key findings of a web search?’. These tables are evaluated in the next chapter in more detail.

In Chapter 3, I build upon the related work presented in this chapter and evaluate it. Mainly, the work presented in Chapters 2.1, 2.2, 2.3, 2.5, and 2.6 is evaluated. This way, the related work helps developing requirements and finally results in a rough architecture of the envisioned system.

Chapter 3
Concept and approach

In the first two chapters of my thesis, I provide a motivation and an overview of related work on supporting asynchronous, collaborative, complex search. In this chapter, I draw conclusions from both the motivation in Chapter 1 and the related work in Chapter 2 of the thesis. Based on the motivation and the related work for the thesis, I progress with developing requirements for a system which is able to answer the main research question developed in the Chapter 1: ‘Can search trails provide support for complex web search and how should tool support look like?’. I name the envisioned system ‘**SearchTrails**’, as it is able to support asynchronous, discontinuous, collaborative, complex search tasks by building search trails.

In the motivating scenarios in Chapter 1, I describe three situations in which users face complex search tasks and want to collaborate with other users on the information found. These scenarios do not describe interaction with a concrete system, but make the problems with current systems visible. The envisioned system ‘SearchTrails’ has to ensure that the problems described in the scenarios can be solved with it. The related work in Chapter 2 shows that most current systems cannot provide support in these problem scenarios.

This chapter helps working out how the envisioned system needs to be constructed to achieve its goals. The analysis helps building on best practices and at the same time avoiding errors that have already been made. In Chapter 3.1, I develop high level requirements from the motivating scenarios and abstract requirements from the analysis of the related work, which specify system properties or the system behavior in a non-technical way. In Chapter 3.2, I cluster the abstract requirements from the first subchapter and derive technical requirements from them for the envisioned system. In Chapter 3.3, I develop a concept of SearchTrails in terms of a series of user interfaces with their respective features. In Chapter 3.4, I outline a rough architecture for the envisioned system, while the Chapter 3.5 summarizes the main results.

This chapter as a whole delivers input for answering the research questions RQ1.1 ‘Search trails valid?’, as it develops the core ideas for a system which transforms user interaction into search trails. Chapter 3.2 delivers input for answering research question RQ1.2 ‘Which functions?’ by the technical features developed in this subchapter.

3.1 Deriving abstract requirements for SearchTrails

This subchapter is divided into two sections. In the first section, I summarize and analyze the three motivating scenarios from Chapter 1. I derive a set of features from each scenario, which could help the users in the specific case. I then summarize these features to a set of desired high-level characteristics of the envisioned system. The features are meant to sensitize the reader for the analysis of the related work from Chapter 2, which is done in the second section of this subchapter. I derive abstract technical requirements from the related work and the comparing tables. The abstract technical requirements do not describe detailed technical solutions.

They define a set of desired characteristics the envisioned system SearchTrails needs to fulfill. This ensures that SearchTrails can both provide help in the motivating scenarios and at the same time builds upon best practices of the related work and avoids commonly made errors. In Chapter 3.2, I match the abstract requirements with the technical requirements, which form the base for developing a concrete concept and an architecture of SearchTrails.

3.1.1 Analyzing the motivating scenarios

The historical origins of the information overload problem presented in Chapter 1 reveal that the problem has always been a child of its time, and each generation has found ways to cope with it. In our days, it seems that not the existence of information is questionable, but the ability to find it. Be aware that the information might not be available in one document, but requires the user's individual synthesis of information from several resources, which complicates sharing a set of resources. Logically, researchers demand for 'methods for building on previous reading and experience, so that we are not reduced to searching for everything anew' (Blair, 2010a).

The three scenarios in Chapter 1 each describe a problem which occurs during complex search tasks. Here, I take a glance at the three scenarios with respect to key features a software system like SearchTrails should offer to support the complex search tasks described in the scenarios. These features also cover the basic requirements mentioned in Chapter 1.3 and provide more detail on them.

The first scenario ('Capturing a search process', cf. Chapter 1.1.1) describes how the single user Chris' interest is drawn on the topic of coffee preparation, which triggers him to do an initial complex search on the topic. He gathers a lot of information during the search process on coffee preparation, but the information is of very inconsistent quality. After the initial search session is over, he wants to remember the helpful resources, but is unable to do so. In Chapter 1.2, I derive a challenge from the scenario, which is: How to capture the information within its context going beyond plain bookmarks? And how is it possible to mark the helpful and the less helpful search results? This leads to several individual features which could achieve this (The concept names in brackets allow clustering the high-level requirements at the end of this section):

- Capture the interaction of the user with the Web during complex search tasks. (Capture)
- Build a visual search trail of the user's navigation through the Web. (Visualize)
- Let the user point out which resources are valuable. (Interact)
- Let the user point out which resources do not contain the desired information. (Interact)
- Capture the valuable resources within their context. (Capture)

The second scenario ('Exchanging search results', cf. Chapter 1.1.2) describes how a conversation between Chris and Carol triggers the wish to technically exchange information on a topic. While Carol faces an information need on a certain topic, Chris may help with offering information on the topic. This information has been gathered during a previous search process and contains very valuable resources, but also resources which turned out not to be helpful.

Carol now wants to build upon the information to get into the topic more efficiently. The challenge derived from this scenario is: How is it possible to enable collaborative search, such that workers can start working from other users search results? The following features can provide help in the described situation:

- Store the search trails on a remote server. (Store / Retrieve)
- Remain the unfiltered search results in the search trails. (Capture)
- Provide search trail identification numbers for easy access of the search trails. (Store / Retrieve)
- Allow the extension of another user's search trail. (Extend)

The third scenario ('Continuing a search', cf. Chapter 1.1.3) describes how a biomedical researcher starts conducting a complex biomedical research, but gets disturbed after a while by an urgent experiment. After some time, the researcher wants to continue the previous research. At this time, the cognitive context of the started search task is lost. The data could also be gone completely, as a forced system reboot has triggered cleaning the browser history. The challenge derived from this scenario is: How to resume an interrupted search processes and to update the information on a topic? This leads to the following high-level requirements:

- Store the unfiltered search process of a user as a search trail. (Capture)
- Regularly save the current progress. (Store / Retrieve)
- Allow catching up on search processes by recreating search trails. (Store / Retrieve)
- Allow investigating the detailed search process of collaborators. (Interact)
- Allow highlighting valuable information and marking value-less resources. (Interact)
- Visually organize parts of the search trails to allow easy evaluation. (Interact)
- Allow the extension of an own search trail. (Extend)

The high level requirements derived from the three scenarios in Chapter 1 already provide an idea of the overall functionality of SearchTrails. Clustering the requirements resulting from the scenarios, SearchTrails has to fulfill the following high-level requirements:

- **HL1 'Capture'**
  SearchTrails is a system that captures the course of a user traversing the Web during a complex web search as a search trail.
- **HL2 'Visualize'**
  The actual visual search trail shall be generated automatically based on the interaction of the user with the Web.
- **HL3 'Interact'**
  SearchTrails should offer the users possibilities to interact with the trails. These possibilities include enriching the search trail with valuable information, marking valueless resources, and methods to automatically organize the search trail.
- **HL4 'Store / Retrieve'**
  SearchTrails needs to provide an ID when storing the search trail, with which the respective search trail can be retrieved from the external storage platform. The search trail can be retrieved via the Internet by anyone who has the ID.

- **HL5 ‘Extend’**
  It needs to be possible to extend search trails retrieved from the storage server and to automatically store the extended trails.

I refer to the set of high level requirements (HL1 - HL5) at the end of Chapter 3.2, where the technical requirements are matched with the abstract requirements. At this point, it becomes visible how the envisioned technical solution is able to provide support in all three scenarios. In the next subchapter, I develop abstract requirements for SearchTrails. They do not refer to concrete technical solutions, but build upon the related work and make sure that best practices are followed, while commonly made errors are avoided.

3.1.2 Abstract requirements derived from the related work

In order to achieve the desired functionality of SearchTrails, Chapter 2 of my thesis presents theoretical and practical work related to several aspects of the described concepts of SearchTrails. For shaping the concept of SearchTrails, the subchapters on historical origins (Chapter 2.1), theoretical foundations (Chapter 2.2), related systems (Chapter 2.5), and related work for other parts of the thesis are most relevant (Chapter 2.6).

Considering the historical origins of the idea of SearchTrails shown in Chapter 2.1, the concept of search trails by Vannevar Bush is the most important concept for this thesis and provides the initial idea for realization (Bush, 1945). Bush describes the process of capturing and exchanging individual search trails between users. He leaves open how this idea can be realized and how search trails actually look like. The overview of related work indicates that this approach has not yet been realized. Several researchers have captured web log data to analyze it. A few have even captured the web interaction data to build trails from it and analyze it. However, none of the approaches puts value in individual trails and tries to enable the exchange of these resources. The overall realization of the system SearchTrails brings the idea of capturing user navigation as trails of resources to life. Based on this information, I specify abstract requirement **AR1** as: **SearchTrails should capture user navigation as a trail of resources**.

Chapter 2.2 presents several theoretical foundations for this work. Starting with a distinction of exploratory and complex search, Chapter 2.2 presents an overview of important definitions, search task classifications, and different models of search. An overview of approaches for collaborative search and a view on the value of negative search results closes Chapter 2.2. My thesis relies on the definition of complex search from (Singer, 2012, p.17), as it avoids the cognitively high loaded aspects of Marchionini’s definition of exploratory search (Marchionini, 2006). As such, SearchTrails needs to support all three single aspects of complex search, which are aggregation, discovery, and synthesis. The different approaches of search task classification reveal that informational search tasks as defined by Broder (Broder, 2002) account for a major share of all web-related search activities. These results get confirmed by the studies of (Rose & Levinson, 2004), and (Lewandowski, 2006). Informational search tasks can be seen as the most complex and longest-running search tasks. With supporting those complex informational search tasks, SearchTrails aims for improving the searcher’s capabilities when facing the most complex search tasks. In their model of search processes, (Pirolli & Card, 2005) model the search process as consisting of two loops, the foraging and the sensemaking loop. Within the foraging loop, the searcher collects evidence from external data sources via the Web, processes it, and creates an evidence file from it. Based on the gathered evidence, the sensemaking loop aims for cognitive processing of the data and condensing it for presentation. This model was taken up by (Evans & Chi, 2008a), who claim that social interaction and information exchange happen before, during, and after the search process; but especially during the process of foraging and sensemaking. SearchTrails tries to improve the foraging process by offering support during aggregation and discovery of information. SearchTrails improves sensemaking by supporting the synthesis of information, and enables the exchange of the results of the search process between collaborating users. While aggregation of information can be considered to be achieved by the search trail itself, discovery is achieved by generating relevant keywords from the search context and displaying them to the user during the search. Synthesis of information is improved by a condensed highlight overview of the valuable information which was found during the search. I specify the abstract requirement **AR2** as: **SearchTrails should improve the support for complex search by supporting aggregation, discovery, and synthesis of information**.

In Chapter 2.2.5, I present a chronological overview of approaches for supporting collaborative search processes. Among these approaches are the four dimensions of collaboration, as introduced by (Golovchinsky et al., 2009) and the four basic aspects of collaboration, as introduced by (Kelly & Payne, 2014). Among other self-defined criteria, I take these dimensions and aspects for comparing the presented collaborative search systems. (Garfield, 1970) points out the value of negative results (cf. Chapter 2.2.6). This may be the information that a certain resource does not exist, or that certain information does not exist yet. Especially in scientific contexts, finding no result may be desired. Although support for this concept has not been incorporated a lot in the literature, SearchTrails should be able to capture the search process as a whole, such that users can see where collaborators have been searching, but did not find information. The abstract requirement **AR3** is therefore specified as: **SearchTrails should be able to mark pages where no valuable information was found**.

In Chapter 2.3, I present insights into research results on the way real users conduct their search processes and the problems they face during search. The results presented in this subsection include that users do often have no fixed strategy for searching but instead adapt to the recent situation. The understanding and the use of search engines tends to be faulty, which can be seen by the often faulty use of search operators and the prevalence of short queries. Additionally, the complexity of a search process itself is very hard to estimate for the users, as the difficulty of a search task cannot be judged by all users equally good, and their success cannot be detected by statistical measures. All these results imply that users must not be forced into a strict process, and that individual search approaches need to be supported. For SearchTrails, this means that search support needs to go along with the user-defined process of search and may not enforce specific actions.

Evaluating the comparing table for search logging systems

In Chapter 2.5, I describe a set of 11 search logging systems and 16 collaborative search support systems and compare them in two comparing tables (cf. Chapter 2.5.1.2 for search logging systems and Chapter 2.5.2.2 for collaborative search support systems).

Depending on the characteristics of the presented eleven search logging systems, I develop a set of nine criteria which allow the comparison of the technical abilities of the presented search logging systems (cf. Chapter 2.5.1.2). These criteria reveal a number of technical details which all need to be solved by SearchTrails. The following list explains how SearchTrails is positioned with respect to all nine comparison criteria. I develop abstract requirements for SearchTrails from the best practices on these criteria.

1. **Data collection approach**. This criterion reflects the historical development of the search logging systems. While early approaches made use of self-developed proxies capturing the user's interaction with the Web, later approaches were stand-alone applications, which needed to be installed on a computer. As those systems were installed on the PC, they were able to monitor the users' interaction with the operating system and a web browser. The next approach for developing search logging systems was writing custom browsers, often being based on Microsoft's Internet Explorer. With the development of the Firefox web browser since 2002 and the Google Chrome web browser since 2008 and their possibilities of being enriched by self-written extensions, the development of search logging systems first turned towards Firefox extensions. This approach seems to have become a standard approach for developing search logging systems, which SearchTrails should follow. The abstract requirement **AR4** is specified as: **SearchTrails should be realized as a browser extension**.
2. **Universe of search**. This criterion aims for the data that the searcher is allowed to investigate. It is remarkable that most presented systems are able to capture the users' interaction with the Internet, whereas a few collaborative search support systems rely on predefined databases. It is essential for SearchTrails not to limit the users with respect to the universe of their search. Based on these observations, abstract requirement **AR5** is defined as: **SearchTrails should be able to capture data from all web pages and not limit its users**.
3. **Captured data**. The type of captured data depends on the data collection approach. The more holistic the data collection approach, the more data can possibly be stored by the respective search logging tools. However, only the most important data is stored. SearchTrails wants to capture the user's interaction with the web, which means that not the interaction with the single pages' content, but the interaction with the navigation functions of the browser is relevant. I specify abstract requirement **AR6** as: **SearchTrails should capture the user's interaction with the browser's navigation functions**. In order to allow the users to work with their captured data, I specify abstract requirement **AR7** as: **The user should be able to interact with the generated search trail via mouse and keyboard commands**.
4. **Accessibility of the search logs to the user**. Despite very early approaches, none of the presented search logging systems makes the logs available to the users. The logs seem to be considered secret knowledge the user may not interact with. According to (Singla et al., 2010, p.443), I believe that there is a 'value in trails' for the users when the system allows them to access their logs. Therefore, I specify abstract requirement **AR8** as: **The users should be able to access the search logs interpreted as search trails**.

5. **Forced feedback during search**. Some of the presented systems are intended as research frameworks for discovering factors of web pages that determine the users' interest in single web pages. These systems mostly require the users' feedback for each visited web page. To achieve this, most of these systems automatically present feedback forms to the users. With respect to the navigation behavior, these feedback forms disturb the users' natural browsing behavior and destroy their cognitive context. SearchTrails therefore omits forced feedback during the browsing process. This property is covered by abstract requirement AR9 (cf. next criterion). User studies have to incorporate pre- and post-evaluation questionnaires that determine the users' opinion towards SearchTrails.
6. **Unobtrusiveness of the system**. Several properties of a search logging system can lead to an obtrusive impression of it. Whenever feedback is forced from the users, the cognitive context is disturbed and the system does seem obtrusive. One of the presented search logging systems uses eye tracking devices to prevent interruptions during the search process. However, these devices are obtrusive in that they constantly remind the searcher of the evaluation situation. SearchTrails should not force the user to do any interaction with it, but allows several interaction possibilities. This means that the search process does not get interrupted unless the user actively decides to do so. I therefore specify abstract requirement **AR9** as: **SearchTrails should be unobtrusive**.
7. **Type of data storage**. Early examples of search logging systems stored the data generated during the search processes on the local hard disk. After the experiment, the researchers collected the data files from the respective computers. This way of collecting data is easy as long as the participants work with the system in a lab situation. Whenever the participants should make use of their own computers for the search tasks, it is more convenient to store the data via a web connection in a centralized server. As SearchTrails will be evaluated in a field test, I specify abstract requirement **AR10** as: **The search trail data storage should be realized on a remote server and ensure regular storage intervals**.
8. **Helping the user or just capturing data?** Almost all of the presented search logging systems consider the generated information secret knowledge and do not share it with the users. The only exception is the system presented in (Danilov & Vainikko, 2013), which focuses less on logging search activities, but more on generating recommendations from the user's search behavior, as determined by the context of the opened pages. In order to transport the value of the search trails to the user, the search trails have to be made available to the user to help them. As SearchTrails focuses on supporting asynchronous, collaborative, complex search tasks, it aims for helping the user. This is very similar to abstract requirement AR2.
9. **Purpose of the system**. The presented systems have been developed for a number of different purposes, of which supporting the users is the least prominent. While one presented system aims for the classification of web search moves of the study participants, four other systems aim for detecting explicit and implicit interest indicators on web pages. Other systems are intended as frameworks for conducting information retrieval experiments. In contrast to the properties of many other of the presented systems, I aim for supporting users during asynchronous, discontinuous, collaborative, complex

search tasks and specify abstract requirement **AR11** as: **SearchTrails should enable the users to exchange unfiltered search trails**.

The position of SearchTrails towards the nine criteria can be summed up in the table below (Table 11). The table row shows the same nine criteria as the comparing table in Chapter 2.5.1.2. This table can therefore be extended by the table row and shows how SearchTrails fits into the context of related work with respect to search logging systems.

Table 11: Extension of the search logging systems comparing table with SearchTrails.

	Data collection approach	Universe	Captured data (key, mouse, eye, …)	Logs accessible to users?	Forced feedback during search?	Unobtrusive	Data storage	Help user or just capture?	Purpose / Intention
SearchTrails	Browser extension	Web	Navigation data, text fields, keyboard commands	Yes	No	Yes	Remote server	Help user	Prove the value of exchanging unfiltered search trails

Evaluating the comparing table for collaborative search support systems

Similar to the comparison of the search logging systems, I present a set of 16 systems for supporting collaborative search processes in Chapter 2.5.2.1. From the literature, I derive a set of eight criteria for comparing those systems, namely the four dimensions of collaboration (Golovchinsky et al., 2009) and the four basic aspects of collaboration (Kelly & Payne, 2014). These eight criteria already classify collaborative search support systems by a number of technical factors and factors relevant to collaboration. SearchTrails has to take a position towards each of these eight criteria.

1. **Intent**. Most of the early systems for supporting collaborative search aim towards supporting explicit collaboration. This means that the collaboration between users often happens synchronously during the search process. Starting with the system ResultsSpace, the systems have turned more towards supporting implicit collaboration. In these systems, users no more interact explicitly with known collaborators, but get recommendations based on other users interactions or can build on curated content that has been produced by other unknown users. It seems like the increasing computing capabilities have turned the research focus more on supporting implicit collaboration. SearchTrails aims for supporting asynchronous, collaborative search. Based on this, the intent of SearchTrails is explicit, as the users build upon unfiltered search artifacts, which reveals the possibly changing information need to the collaborator. The intent of SearchTrails is already reflected in abstract requirement AR2 and AR11.
2. **Depth of mediation**. The depth of mediation describes how deep the individual user data is drawn into the overall search system before interaction between users is deter-

mined and evaluated. In shallow systems, the interaction data is processed mainly on the UI-level, which means that collaborators most probably interact directly with other users' data. In deep systems, the user data is distinguished at search engine-level, where for example search recommendations are generated. This implies that users interact with the highly processed data of other users. As SearchTrails tries to show the value of search trails without manipulating search engines, the depth of mediation is shallow. The depth of mediation is not reflected in an abstract requirement, as it is reflected in the general idea of SearchTrails.

3. **Concurrency**. Many of the related systems for collaborative search support follow a synchronous approach of collaboration. These systems often require all collaborators to search actively at the same time. However, synchronous communication requires a coordination of actions, which may be inflexible. SearchTrails therefore works in an asynchronous manner, leaving its users the freedom to choose when they want to work on a search task. SearchTrails does not support synchronous collaboration. I therefore specify abstract requirement **AR12** as: **SearchTrails should allow catching up with previous search processes by recreating and evaluating search trails**.

4. **Location**. Similar to the criterion 'Concurrency', 'Location' determines whether the collaborators need to be co-located or if they may work remotely with a tool. Some of the presented systems aim for solving collaboration issues when several collaborators need to share the same computer for a collaborative search. SearchTrails does not require the collaborators to be at the same place, thus affording remote collaboration. I therefore specify abstract requirement **AR13** as: **SearchTrails should provide a way to allow working on search trails from any place**.

5. **Awareness**. This criterion refers to the awareness of the collaborator's ongoing or past actions. Systems that realize implicit interaction also provide no awareness on the collaborator's actions, as the interaction between the collaborators happens indirectly on processed user actions. One exception is the CIRE system, which does not incorporate the users' actions explicitly into the user interface. This leads to a missing awareness of other searchers, and made most of the users forget about the collaborative functions of the system. As SearchTrails enables explicit, asynchronous interaction, awareness of the collaborator's past actions can be ensured, which is therefore not reflected in a dedicated abstract requirement.

6. **Division of labor**. Several systems supporting synchronous collaborative search offer features for explicit division of labor between collaborators. This is achieved by e.g. different roles during the search process in Cerchiamo (Golovchinsky et al., 2008) or by distributing search results among collaborators in SearchTogether (Morris & Horvitz, 2007). Asynchronous systems cannot easily divide labor between collaborators during the search process, but may allow splitting the work into sessions. Therefore, abstract requirement **AR14** is specified as: **SearchTrails should allow searchers to split their search into single sessions**. This allows collaborators to deduce from the past actions of the searcher which search topics may not yet have been worked on. SearchTrails therefore does not explicitly support division of labor, but as the collaborators can see the unfiltered search trail, they can concentrate on searching in new directions.

7. **Persistence of results**. Most of the presented systems for supporting collaborative search store the generated results. This does not yet reveal how detailed these results are. Systems like SearchTeam or So.cl capture curated content, but not how the users came to it. SearchTrails should store the search results persistently on a remote storage server, but it also stores the way how users came to the search results. This is very similar to abstract requirement AR10.

8. **Sensemaking**. This criterion refers to the support of sensemaking of the found resources or information. This can be achieved by e.g. visualizing search strategies or by publishing artifacts that structure the process of searching, such as the task lists in CoSense (Paul & Ringel Morris, 2009). SearchTrails supports sensemaking by providing an unfiltered way of interaction with the collaborator's search trails. Therefore, I specify abstract requirement **AR15** as: **SearchTrails should support sensemaking by the creation and annotation of the valuable pieces of information (highlights) in a combined overview with their respective sources**. From these search results, users can evaluate the valuable information the collaborator has found as well as the context in which it was found.

A summary of the results for SearchTrails for the first eight criteria is given in the table below (Table 12). The table row shows the same eight criteria as the first comparing table in Chapter 2.5.2.2. This table can therefore be extended by the table row below and shows how SearchTrails fits into the context of related work with respect to collaborative search support systems.

Table 12: Extension of the 8 characteristics of collaborative search support systems with SearchTrails.

	Intent	**Depth of Mediation**	**Concurrency**	**Location**	**Awareness**	**Division of labor**	**Persistence of results**	**Sensemaking**
SearchTrails	Explicit	Shallow	Asynchronous	Remote	Yes	Not explicitly	Yes	Yes

Additionally, I developed a set of nine further criteria to distinguish the most important aspects of collaborative search support systems, which are introduced in Chapter 2.5.2.2. These criteria are generated from key features of the presented systems. Several of the presented systems perform very well with respect to some of the criteria. In order to build upon successful concepts and to avoid duplicating the flaws of existing systems, these criteria help determining which key properties SearchTrails should implement to be a successful approach.

9. **Lightweightness**. According to several authors (e.g. (Ringel Morris, 2013) and (Kelly & Payne, 2014)), the searcher's feeling of lightweightness of the systems is an important property of collaborative search support systems. Systems which are cluttered with functions and do not immediately reveal their feature set to the user have very limited chances to be used frequently. Although lightweightness is hard to measure,

SearchTrails strongly aims for being lightweight in offering a set of easily understandable functions, which is not reflected in a dedicated abstract requirement.

10. **Synchronous communication**. Some of the presented systems support synchronous search processes. Some of these systems integrate means for synchronous communication into their user interface, such as instant messaging functions. Naturally, synchronous communication can only be supported for synchronous collaboration. Therefore, SearchTrails does not support synchronous communication.
11. **Technology**. Similar to the data collection approach of the search logging systems, the presented systems for collaborative search support are realized with various technical approaches. While early approaches have built their own modified web browsers, later approaches relied on client/server architectures. Special solutions have incorporated surface tables or computers with several mice. Later approaches provide views on preprocessed databases via specialized browsers, or are available as online services. As one of the more recent collaborative search support systems, Coagmento (González-Ibáñez & Shah, 2011) is based on a browser extension, may indicate that also collaborative search support systems become more browser based when they focus on supporting individual users. This matches the experiences with search logging tools, such that SearchTrails will be realized as a browser extension. This is very similar to abstract requirement AR4.
12. **Artifact**. This criterion refers to the artifacts which are created during the collaborative search process. The presented systems have created different artifacts. These range from plain ratings in the case of recommendation-focused systems such as ResultsSpace or Querium, to bookmarks in the early approaches CSCW3 and CIRE. Additionally, some systems allowed the creation of comments, categories, to-do-lists, or curated lists of resources as in So.cl. SearchTrails aims for creating trails reflecting the search process. Independent of the internal representation in the storage, the search trails include features like highlights, clusters of nodes from the same host, keywords of web pages, and marked nodes. Therefore, I specify abstract requirement **AR16** as: **The artifacts created by SearchTrails or by the user in should include the search trail, highlights, clusters, keywords, and marked nodes.**
13. **Hardware needed?** Most of the presented systems do not need additional hardware for being used. The only exceptions are CoSearch, which enables a computer to cope with signals from several mice, and WeSearch, which was specially designed for a large surface table. SearchTrails does not require additional or specialized hardware, thus ensuring an uncomplicated setup without any more special components than a web browser.
14. **Unknown cooperation partners?** This criterion refers to the scope of the cooperation partners. While early systems focus more on direct cooperation with co-located or remote known collaborators, later approaches allow cooperation with unknown partners. These include web-based platforms like So.cl, where users can publish collections of resources to the public. SearchTrails offers the possibility to cooperate with known or unknown cooperation partners. When exchanging search trail IDs between collaborators, e.g. via e-mail, the partners are known to each other. It is also possible to create a search trail repository, where anonymous trails can be taken from. I specify abstract re-

quirement **AR17** as: **SearchTrails should allow remote asynchronous collaboration of users by the exchanging of search trails.**

15. **Temporality**. Only a small share of the presented systems preserves the temporal sequence of actions in a way that it can be recovered after the actual collaboration process. The CIRE system stores all user actions as the 'Information retrieval memory', while CoSense features a timeline view of all user actions. Coagmento implements an activity stream of user actions, while Querium offers a history of past queries. SearchTrails goes beyond the purely chronological approaches of the presented systems by connecting the chronological view with the sequence of the visited resources. As SearchTrails records the search process as a trail of visited resources, I specify abstract requirement **AR18** as: **SearchTrails should keep the temporal sequence of user actions**. This way, collaborators can understand how the searcher's results build upon another, and the search for which resources was motivated by visiting which pages.
16. **Universe**. Similar to the evaluation of search logging systems, collaborative search support systems may be based on a limited universe of resources. However, most of the presented systems do not rely on a limited universe, but allow the searchers to interact with the Internet without limitations. Exceptions are the ViGOR system, which was specially designed for a video database, and the systems ResultsSpace and Querium, which focus on document recommendations. Therefore, the universe of these two systems was limited to a database of text documents. Abstract requirement **AR19** is therefore specified as: **SearchTrails should not limit its users to a certain database and allow free interaction with all resources of the Internet**.
17. **Possibility of rating resources?** A major share of the presented systems explicitly allow the rating of resources by dedicated rating options. These rating options may by binary (e.g. like or dislike options) or provide more steps (e.g. by a star rating). Systems offering only a commenting function for the found resources are not counted as offering rating options. However, the availability of these functions does not express the frequency of their usage. It is especially hard to achieve a high number of ratings from a user, as every rating is considered extra effort. SearchTrails offers the possibility of implicit ratings: When a highlight is found on a resource, the resource is automatically considered to be valuable and is highlighted within the search trail as a valuable resource. When a resource turns out to be not valuable, the user has the option to explicitly rate down the respective resource. Abstract requirement **AR20** is therefore defined as: **It should be possible to mark resources in the search trail to reflect high or low value.**

A summary of the results for the criteria 9 to 17 is given in the table below (Table 13). The table row shows the same nine criteria as the second comparing table in Chapter 2.5.2.2. This table can therefore be extended by the table row below and shows how SearchTrails fits into the context of related work with respect to collaborative search support systems.

Table 13: Extension of the 9 characteristics of collaborative search support systems with SearchTrails.

	Lightweight	Synchronous communication	Technology	Artifact	Hardware needed	Unknown part-ners	Temporality	Universe	Rate resources
SearchTrails	Yes	No	Browser extension	Search trails with highlights, clus-ters, keywords, marked nodes, …	No	Yes	Yes	Web	Yes

Visualizing information

Chapter 2.6 presents related work for other parts of the thesis. This includes approaches for visualizing large amounts of data and mathematical measures of graph complexity.

As SearchTrails wants to represent the user's interaction with Internet resources as a trail of visited resources, special care has to be taken with respect to the visualization of this type of information. The historical perspective can give more hints on what type of visualization may best fit this application case. Although Bush (Bush, 1945) does not provide any hints on how he thinks about a search trail in a visual way, illustrations like the one from Bates ((Bates, 1989), cf. Figure 8 in Chapter 2.2.4) suggest that a search trail can be considered a path connecting the visited resources. Technically, this path can be considered as nodes representing the web resources which are connected by directed edges. These edges resemble the chronological traversing of the visited resources.

According to (Eades & Huang, 2000), force-directed graph visualizations are suitable for these purposes. This has the advantage that the graph visualization can dynamically adjust to the number of nodes, and the users can still rearrange the graph during its exploration. Any static method would have to constantly adjust to the number of nodes by scaling in and out, or by forcing the user to scroll or drag a drawing pane. The advantage of force-directed graph visualizations is obvious when it comes to creating edges between nodes at opposite parts of the graph. As the force engine simulates a sort of gravity between nodes, an edge connecting two remote nodes drags these nodes together. This way, highly connected parts of the graph lump together and reflect that the user has jumped between a certain set of resources during the search process. Therefore, I define abstract requirement **AR21** as: **SearchTrails should make use of a force-directed graph visualization for the search trail.**

For complex visualizations, the use of clustering algorithms has been investigated by (Stuckenschmidt et al., 2004) and (Fluit et al., 2006). In these studies, it turns out that clustering is generally appreciated as a means for reducing the visual complexity of the visualization of large data sets. Therefore, I define abstract requirement **AR22** as: **SearchTrails should visually cluster the nodes based on the website hosts and offer the possibility to shrink and expand the clusters**. This makes visible which nodes belong to the same host and may be related to each other, such that e.g. nodes from a certain search engine can be easily identified. It also allows the searchers to actively reduce the complexity of the visualization.

Considering the visualization of information about the nodes, the two possibilities of providing thumbnails or textual descriptions exist. Several studies (e.g. (Czerwinski et al., 1999; Dziadosz & Chandrasekar, 2002)) suggest that textual representations of pages are not inferior to visual representations of pages, for example thumbnails. The results of (Sebrechts et al., 1999; Westerman & Cribbin, 2000) also suggest that a textual representation of search results may be most effective. This holds true for the representation of search results, where users want to find some information and can more easily scan through text than evaluate images. In their study, (Kelly & Payne, 2014) find that participants tend to ask for the type of visualization that is not available: In case of available thumbnails participants ask for textual representations, and in case of available textual representations, they ask for thumbnails. Based on these results, I define abstract requirement **AR23** as: **SearchTrails should enrich the force-directed search trail graph by textual representations of the respective web pages.**

Conclusion

This subchapter presents basic ideas for the envisioned system 'SearchTrails', as it derives abstract requirements from the motivating scenarios and the related work. It therefore helps to frame the picture of how SearchTrails conceptually works, and how SearchTrails builds upon both the motivating scenarios and the lessons learned from the related work.

The analysis of the motivating scenarios in Chapter 3.1.1 is followed by a review of related work on historical, theoretical, and practical approaches towards search logging systems and collaborative search support systems. The overview of existing systems for both search logging and support collaborative search helps drawing conclusions and helps SearchTrails building on existing approaches and improving these approaches. Connecting historical ideas to results on visualizing large amounts of data helps identifying the force-directed graph visualization with text labels and clustering as a proper method for visualizing the search trail in the user interface.

Table 14 gives an overview of the abstract requirements generated in this section.

Table 14: Overview of the abstract requirements for SearchTrails.

Identifier	Abstract requirement
AR1	SearchTrails should capture user navigation as a trail of resources.
AR2	SearchTrails should improve the support for complex search by supporting aggregation, discovery, and synthesis of information.
AR3	SearchTrails should be able to mark pages where no valuable information was found.
AR4	SearchTrails should be realized as a browser extension.
AR5	SearchTrails should be able to capture data from all web pages and not limit its users.
AR6	SearchTrails should capture the user's interaction with the browser's navigation functions.
AR7	The user should be able to interact with the generated search trail via mouse and keyboard commands.
AR8	The users should be able to access the search logs interpreted as search trails.
AR9	SearchTrails should be unobtrusive.
AR10	The search trail data storage should be realized on a remote server and ensure regular storage intervals.

AR11	SearchTrails should enable the users to exchange unfiltered search trails.
AR12	SearchTrails should allow catching up with previous search processes by recreating and evaluating search trails.
AR13	SearchTrails should provide a way to allow working on search trails from any place.
AR14	SearchTrails should allow single searchers to split their search into single sessions.
AR15	SearchTrails should support sensemaking by the creation and annotation of the valuable pieces of information (highlights) in a combined overview with their respective sources.
AR16	The artifacts created by SearchTrails or by the user in should include the search trail, highlights, clusters, keywords, and marked nodes.
AR17	SearchTrails should allow remote asynchronous collaboration of users by the exchanging of search trails.
AR18	SearchTrails should keep the temporal sequence of user actions.
AR19	SearchTrails should not limit its users to a certain database and allow free interaction with all resources of the Internet.
AR20	It should be possible to mark resources in the search trail to reflect high or low value.
AR21	SearchTrails should make use of a force-directed graph visualization for the search trail.
AR22	SearchTrails should visually cluster the nodes based on the website hosts and offer the possibility to shrink and expand the clusters.
AR23	SearchTrails should enrich the force-directed search trail graph by textual representations of the respective web pages.

3.2 Deriving technical features from the abstract requirements

In the previous subchapter, I evaluate the motivating scenarios and the related work and develop a set of high level and abstract requirements for the envisioned system SearchTrails. In this subchapter, I cluster and evaluate the abstract requirements from the last subchapter and generate a set of technical features from them, which are realized in SearchTrails. The features describe technical solutions for achieving desired properties of SearchTrails. I describe the concrete technical realization of the features in Chapter 4.

This subchapter is divided into three subsections, each comprising a set of requirements on a functional aspect of SearchTrails. The first section develops the features which are required for realizing the support of all three aspects of complex search. The second subsection develops features regarding the scope and the features of SearchTrails. In the third section, I develop features which enable the user's interaction with SearchTrails. This subchapter provides an overview of how the features make sure that all abstract requirements are fulfilled, and how I cover the high level requirements from the motivating scenarios. At the end of this subchapter, I provide a concluding table, which summarizes the connection between the abstract requirements, the technical features, and the high level requirements.

Supporting complex search

Generally, SearchTrails aims for supporting asynchronous, discontinuous, collaborative, complex search tasks. As complex search consists of aggregation, discovery, and synthesis, each single component needs to be supported. Aggregation is supported by creating a search trail (Abstract requirements AR1, AR2, and AR18), which captures the user's interaction with the browser's navigation functions (AR6). Discovery is supported by extracting keywords which appear in the browsing context and displaying them to the searcher (AR2). Synthesis is supported by enabling the user to collect valuable pieces of information in an overview (AR15), including valuable resources (AR2), and valueless resources (AR3, AR20).

The following list illustrates the planned functionality of SearchTrails by a set of features supporting the three different aspects of complex search. I first describe the feature to be implemented and then mention which abstract requirements are covered by the functionality described.

- **Feature F1: Transform user navigation into a search trail**
  Each visit of a web page is transformed into a node in the search trail. Following a link from one page to another page results in another node being created, which is connected by an edge from the node of the previously visited URL to the new node (Figure 21). The node corresponding to the recently visible URL is indicated by an additional red circle.
  Feature F1 covers the following abstract requirements: Capture user navigation as a trail of resources (AR1), improve the support for complex search by supporting aggregation of information (a part of AR2), and keep the temporal sequence of user actions (AR18).

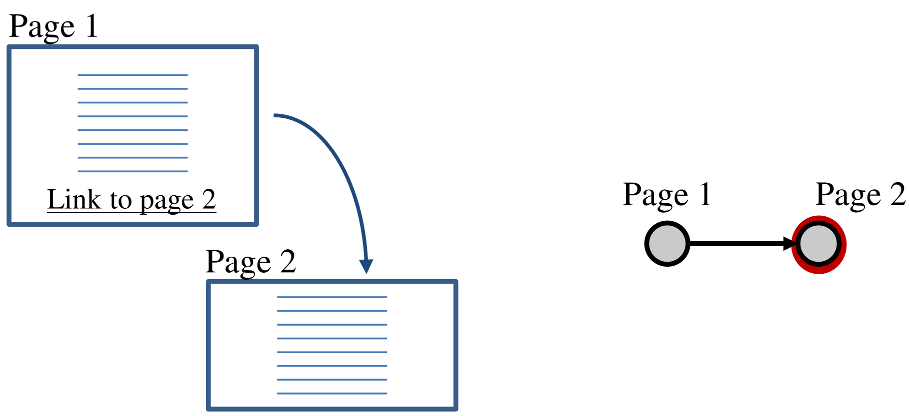


Figure 21: Technical sketch for transforming user navigation into a search trail (Feature F1).

Each node can only occur once in the search trail. This implies that a second visit to a web page results in following existing edges. To avoid cluttering the visualization, reused edges are not specially indicated (Figure 22).

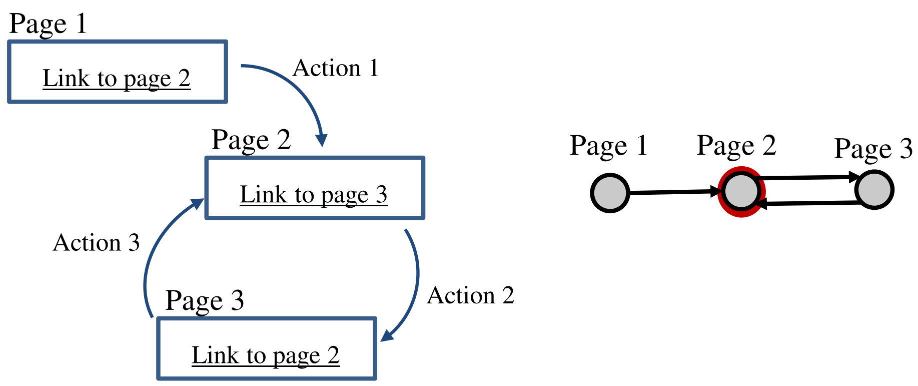


Figure 22: Technical sketch for following existing edges (Feature F2).

- **Feature F2: Capture unfiltered data**
  Independent of the importance of the visited web pages, all URLs are transformed into nodes within the search trail (Figure 23).
  Feature F2 covers the following abstract requirement: SearchTrails captures the unfiltered navigation through the Internet by capturing the user's interaction with the browser's navigation functions (AR6).

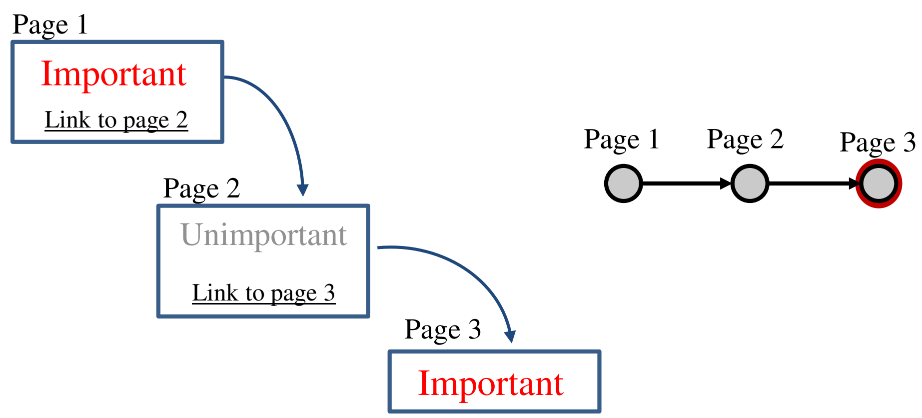


Figure 23: Technical sketch for capturing unfiltered data (Feature F2).

- **Feature F3: Extract keywords from visited web pages**
  A keyword extraction engine scans the contents of the visited pages and extracts relevant keywords from the visited pages. Whenever a keyword appears on e.g. three or more pages, this is considered to happen intentionally and the keyword is added to a list (Figure 24). Keywords in the keyword list may be selected by clicking and highlight the respective nodes in the search trail. Additionally, selected keywords may be used as search terms for starting new web searches.
  Feature F3 covers the following abstract requirement: SearchTrails should improve the support for complex search by supporting the discovery of information (a part of AR2).

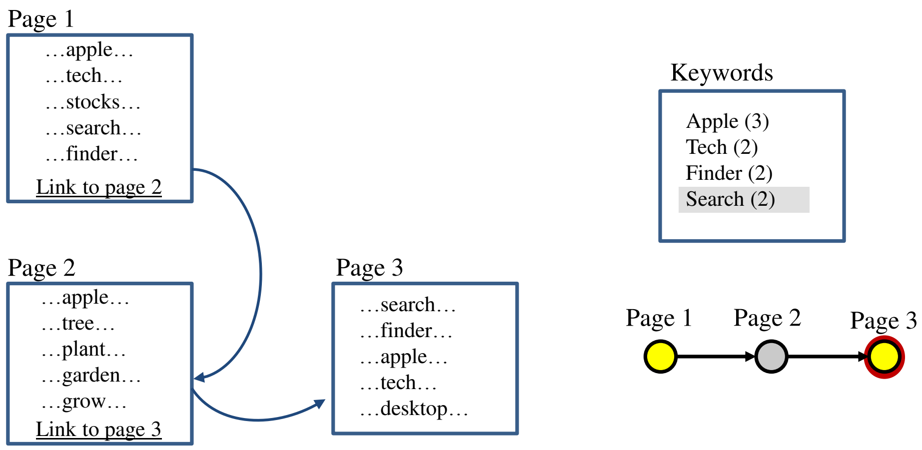


Figure 24: Technical sketch for extracting keywords (Feature F3).

- **Feature F4: Provide an editable highlight overview**
  An overview of all valuable pieces of information and the user's own comments is provided in the highlights view. When a highlight is set on a web page, the respective node is marked blue in the search trail, and the highlight is stored in the highlight overview, together with a link to the source URL. Web pages with explicitly valueless resources may be marked by the searcher when visiting the web pages. They are indicated by a red node in the search trail (Figure 25). Additionally, it is possible to enter manual highlights into the highlight overview.
  Feature F4 covers the following abstract requirements: SearchTrails should support sensemaking by the creation and annotation of the valuable pieces of information (highlights) in a combined overview with their respective sources (AR15). This provides support for complex search by supporting the synthesis of information (a part of AR2). Furthermore, SearchTrails should be able to mark pages where no valuable information was found (AR3), which makes it possible to mark resources in the search trail to reflect high or low value (AR20). Additionally, it should be possible to enter own comments into the highlight overview.

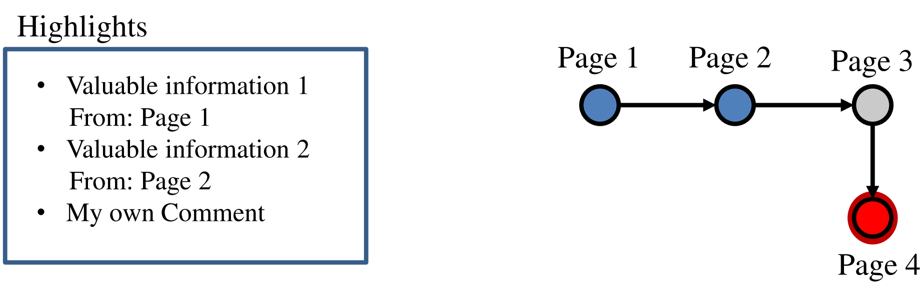


Figure 25: Technical sketch for providing a highlight overview (Feature F4).

Scope and features of SearchTrails

The universe in which SearchTrails users search for results should be the Internet without limitations (AR19) and SearchTrails should be a browser extension (AR4). The browser extension shall run as a separate tab in the web browser, which is unobtrusive because it does not require the user's interaction and allows normal interaction with the browser (AR9). SearchTrails needs to generate the search trail from the users' interaction with the browser and the web pages (AR5), and records a set of data from this interaction (AR16). The users should have the possibility to access the search trail (AR8) and to interact with it (AR7).

The following list illustrates the planned functionality of SearchTrails with respect to the requirements for the scope and features of SearchTrails.

- **Feature F5: SearchTrails as a web browser extension**
  SearchTrails exists as a browser extension, which allows SearchTrails to access the user's interactions with the browser and therefore the Internet. The universe of SearchTrails is not limited. SearchTrails interprets the user's interaction with the standard functions of the web browser (Figure 26).
  Feature F5 covers the following abstract requirements: SearchTrails should be realized as a browser extension (AR4) which does not limit its users to a certain database and allows free interaction with all resources of the Internet (AR19).

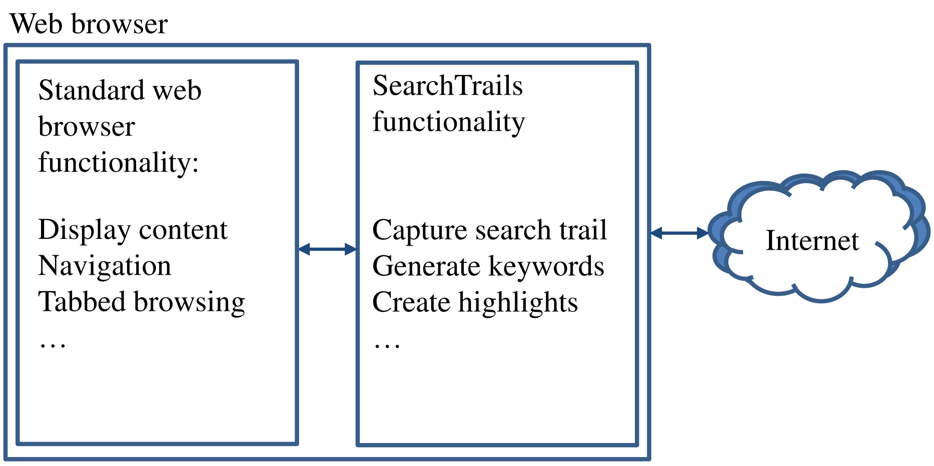


Figure 26: Technical sketch for realizing SearchTrails as a web browser extension (Feature F5).

- **Feature F6: Do not force user interaction**
  SearchTrails does not force the user to do any interaction after having started the system. Additionally, SearchTrails runs as a separate tab in the browser. This allows the users to follow their normal search behavior, with the difference that only one more tab in the browser is open (Figure 27).
  Feature F6 covers the following abstract requirement: SearchTrails needs to be unobtrusive (AR9).

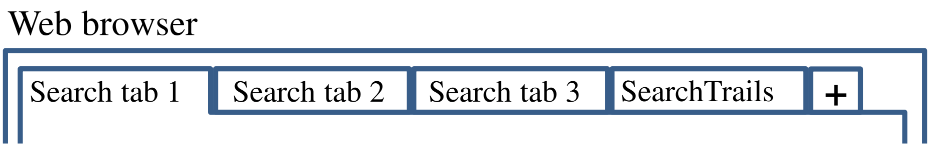


Figure 27: Technical sketch for ensuring unobtrusiveness (Feature F6).

- **Feature F7: Generate the search trail from the user's interaction with the entire Web**

  The search trail is created from every change of the visible content of the browser, no matter where it stems from. This implies that switching between browser tabs results in creating edges between the nodes resembling the two web pages, or the creation of a new node in case it has not been visited before. As a click on the browser's back button results in the changing of the visible URL, this results in creating an edge to the previously visited URL (Figure 28). Therefore, SearchTrails does not limit its users to specific databases.

  Feature F7 covers the following abstract requirement: SearchTrails will be able to capture data from all web pages and not limit its users (AR5).

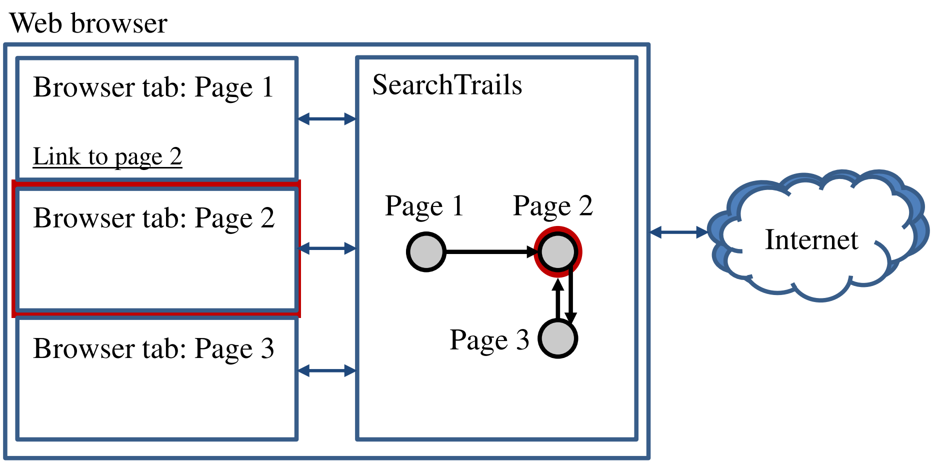


Figure 28: Technical sketch for generating search trails from user interaction (Feature F7).

- **Feature F8: Generate data, but require only little user interaction**

  SearchTrails generates as much data as possible without requiring user interaction, to be as unobtrusive as possible (Figure 29). Additionally, the user has the possibility to set highlights, mark valueless resources, or add own comments to the highlights list.

  Feature F8 covers the following abstract requirement: The artifacts created by SearchTrails or by the user should include the search trail, highlights, clusters, keywords, and marked nodes (AR16).

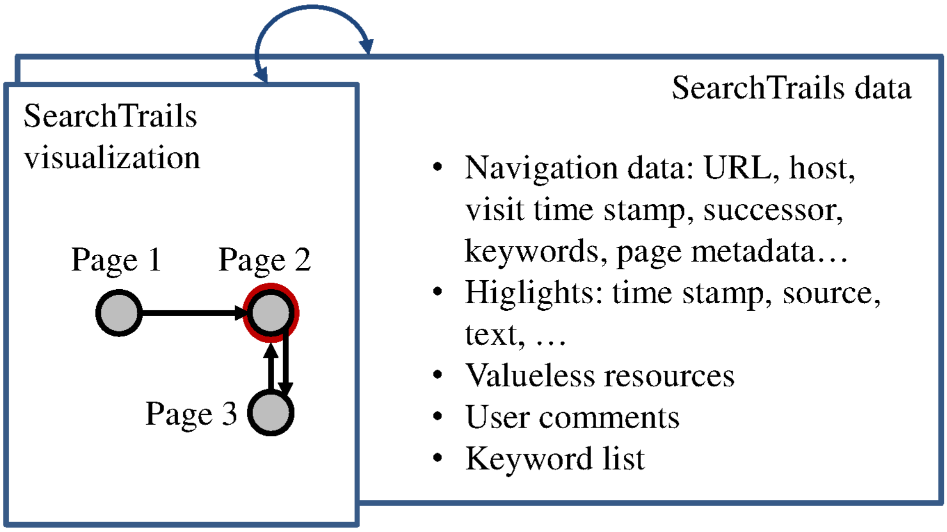


Figure 29: Technical sketch for generating data with limited user actions (Feature F8).

- **Feature F9: Users may interact with the search trail**

  The users have the possibility to access and evaluate the search trail by mouse interaction, e.g. by hovering over the nodes and consuming the information provided by additional pop-ups, consisting of the title of the page, a link to the corresponding web page, and an overview of the highlights attached to the node, if existing. Users also have the possibility to attach hand written comments to the highlights of single nodes. Additionally, users may delete nodes from the graph, if they find a node inappropriate. This only works only when a node does not have multiple in- and outgoing edges (Figure 30).
  Feature F9 covers the following abstract requirements: The users should be able to access the search logs interpreted as search trails (AR8) and should be able to interact with the generated search trail via mouse and keyboard commands (AR7).

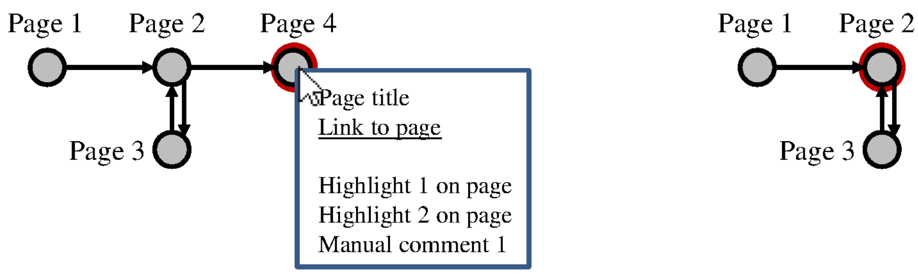


Figure 30: Technical sketch for user interaction with the search trail (Feature F9).

Interaction with search trails

The search trail itself is a force-directed graph (AR21), which is enriched by textual annotations (AR23) and allows clustering (AR22). The search trail is stored remotely (AR10), may be exchanged between users (AR11), may be continued in several sessions (AR14), and SearchTrails provides a means to identify and work on an existing search trail (AR13). This enables users to recover trails from the remote storage server, therefore enabling the recreation of remote trails (AR12), and allowing remote collaboration (AR17) on search trails.

The following list illustrates the planned interaction functionality with the search trails with respect to the requirements of SearchTrails:

- **Feature F10: Visualize the search trail as a force-directed graph**
  The search trail grows during the search process. In order to avoid constant rearranging in a rigid layout, a force-directed layout is chosen to realize the search trail visualization. The force-directed graph dynamically rearranges in a sort of simulated gravity, which makes densely connected parts of the graph lump together (Figure 31). Even if being laid out automatically, the force-directed graph can still be dragged around and be manually rearranged by the user.
  Feature F10 covers the following abstract requirement: SearchTrails should make use of a force-directed graph visualization for the search trail (AR21).

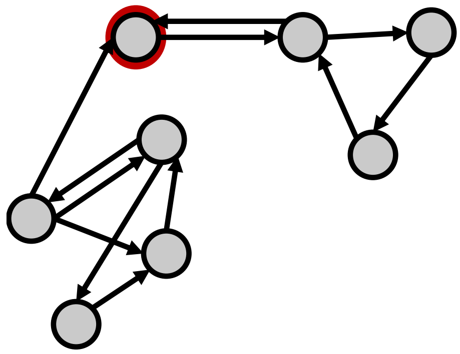


Figure 31: Technical sketch for realizing the search trail as force-directed graph (Feature F10).

- **Feature F11: Enrich the search trail nodes by textual annotations**
  The title of the node is constantly attached to the node in the force-directed visualization. Upon hovering over a node, a pop-up appears, revealing the page title with an underlying link to the corresponding URL and an overview of the highlights attached to the node (Figure 32).
  Feature F11 covers the following abstract requirement: SearchTrails should enrich the force-directed search trail graph by textual representations of the respective web pages (AR23).

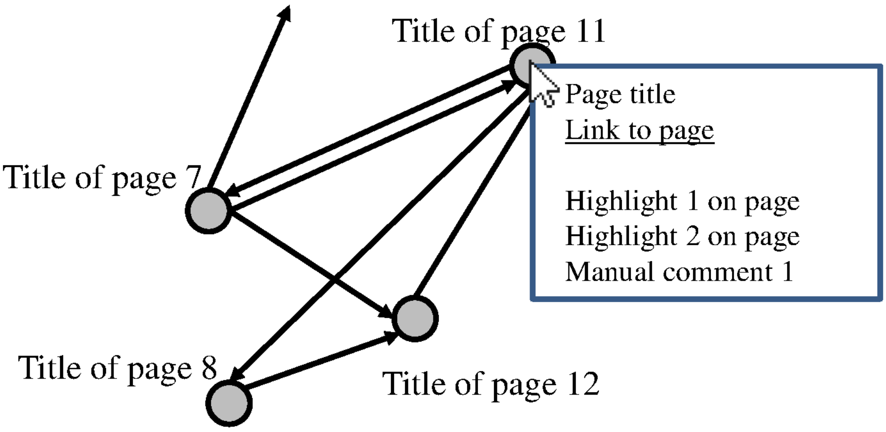


Figure 32: Technical sketch for enriching the search trail by textual annotations (Feature F11).

- **Feature F12: Use clustering for structuring the search trail**
  Clusters in the force-directed visualization comprise the nodes originating from the same host. The clusters are indicated by colored hulls around sets of nodes. Double-clicking a colored hull results in shrinking the cluster to a single node, remaining all in- and outgoing edges. Another double-click on the cluster node expands the node again (Figure 33).
  Feature F12 covers the following abstract requirement: SearchTrails should visually cluster the nodes based on the website hosts and offer the possibility to shrink and expand the clusters (AR22).

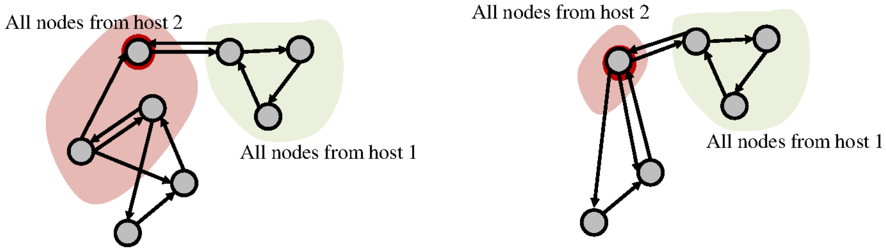


Figure 33: Technical sketch for clustering the search trail (Feature F12).

- **Feature F13: Store the search trail in regular intervals**
  To avoid storing the search trail locally, SearchTrails stores all session data in regular intervals on a remote storage server. For this purpose, the search trail data object which comprises all data of the search session is sent to a remote log storage server (Figure 34).
  Feature F13 covers the following abstract requirement: The search trail data storage should be realized on a remote server and ensure regular storage intervals (AR10). Furthermore, this enables the exchange of unfiltered search trails (AR11).

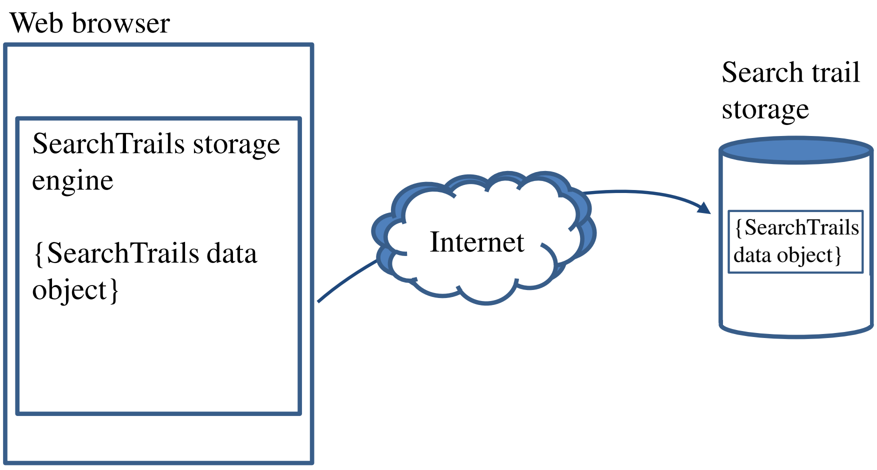


Figure 34: Technical sketch for regular storage of the search trail (Feature F13).

- **Feature F14: Generate a search trail ID for continuing search trails**
  In order to identify the stored search trail on the remote log storage server, a unique search trail ID is generated by SearchTrails. This search trail ID can be used to retrieve the search trail from the server, which is then recreated in SearchTrails (Figure 35). Feature F14 covers the following abstract requirements: SearchTrails should provide a way to allow working on search trails from any place (AR13), which allows continuing previous search processes by recreating and evaluating search trails (AR12).

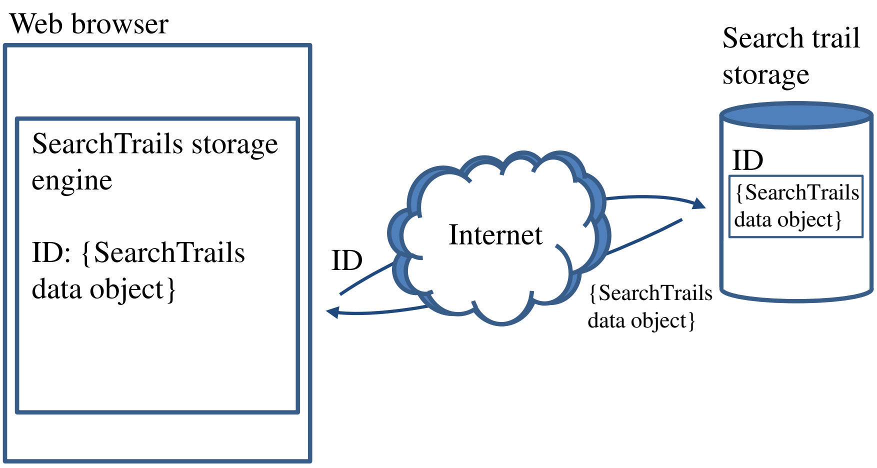


Figure 35: Technical sketch for generating search trail IDs (Feature F14).

- **Feature F15: Allow continuing and extending the search trail**
  When a retrieved search trail is continued by the user, the search trail data object gets extended. The extended data object gets stored regularly on the storage server, each overwriting the older search trail data object (Figure 36), therefore allowing to extend a search process over time and building upon earlier search results.

Feature F15 covers the following abstract requirement: SearchTrails should allow single searchers to split their search into single sessions (AR14).

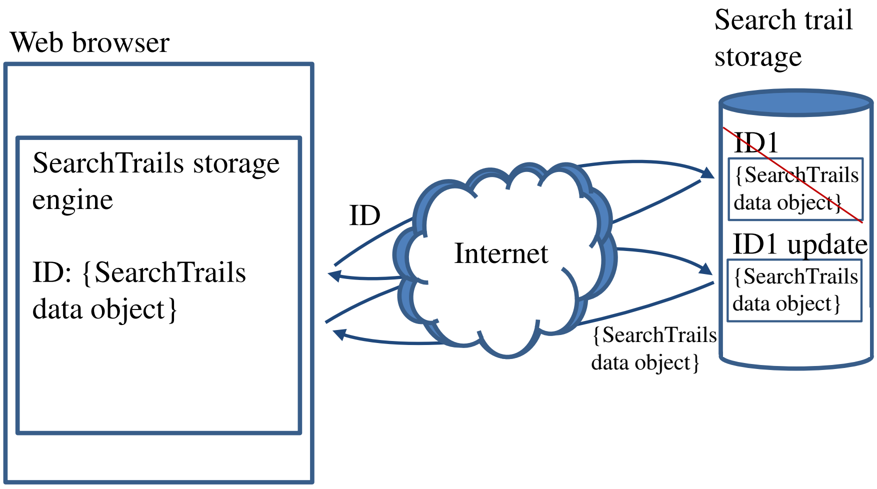


Figure 36: Technical sketch for continuing search trails (Feature F15).

- **Feature F16: Enable asynchronous collaboration**
  As the search trail ID is the only way to identify a search trail once it is stored on the remote log storage server, exchanging the search trail ID grants other users access to the own search trail. The trail can be retrieved from the search trail storage server and gets recreated by SearchTrails, where it can be extended (Figure 37).
  Feature F16 covers the following abstract requirement: SearchTrails should allow remote asynchronous collaboration of users by the exchanging of search trails (AR17).

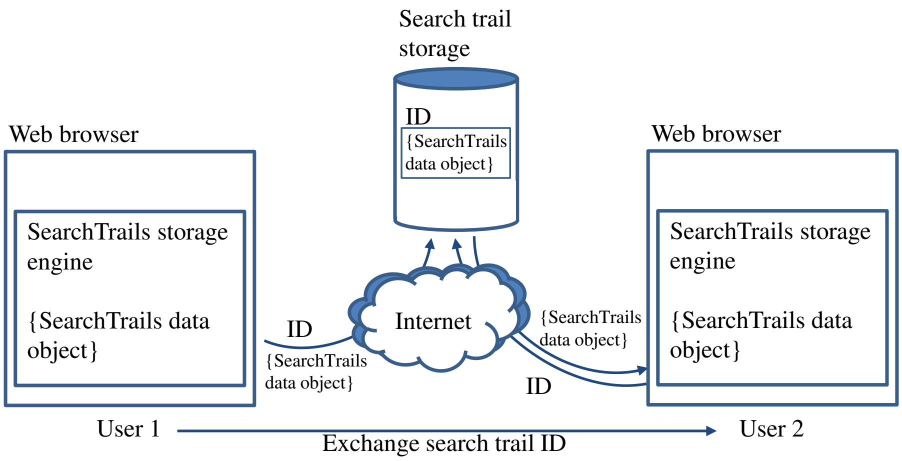


Figure 37: Technical sketch for asynchronous collaboration with SearchTrails (Feature F16).

This subchapter has two main purposes: It first clusters the requirements AR1 to AR23 from Chapter 3.1 and it second maps the clustered requirements to the technical features F1 to F16. This set of functions forms the technical frame of SearchTrails and already defines a number of possible user actions. In the next section, I provide an overview of the different features, abstract requirements, and high level requirements.

Conclusion

In this subchapter, I develop a set of 16 features, which cover all abstract requirements. Additionally, the features can also be mapped to the high level requirements, which I derived from the three motivating scenarios. Table 15 shows the mapping between the features to be implemented, the abstract requirements that are covered by SearchTrails' features, and the mapping of the features and the abstract requirements to the high level requirements from Chapter 3.1.1.

Table 15: Concluding table, mapping features to abstract requirements and high-level requirements.

Section	**Feature** (cf. Chapter 3.2)	**Abstract requirement** (cf. Chapter 3.1.2)	**High level requirement** (cf. Chapter 3.1.1)
Supporting complex search	**F1**: Transform user navigation into a search trail	**AR1**: SearchTrails should capture user navigation as a trail of resources. **AR2**: SearchTrails should improve the support for complex search by supporting aggregation, discovery, and synthesis of information. **AR18**: SearchTrails should keep the temporal sequence of user actions.	**HL2** 'Visualize'
	F2: Capture unfiltered data	**AR6**: SearchTrails should capture the user's interaction with the browser's navigation functions.	**HL1** 'Capture'
	F3: Extract keywords from visited web pages	**AR2**: SearchTrails should improve the support for complex search by supporting aggregation, discovery, and synthesis of information.	**HL1** 'Capture'
	F4: Provide an editable highlight overview	**AR15**: SearchTrails should support sensemaking by the creation and annotation of the valuable pieces of information (highlights) in a combined overview with their respective sources. **AR2**: SearchTrails should improve the support for complex search by supporting aggregation, discovery, and synthesis of information. **AR3**: SearchTrails should be able to mark pages where no valuable information was found. **AR10**: The search trail data storage should be realized on a remote server and ensure regular storage intervals.	**HL3** 'Interact'
SearchTrails	**F5**: SearchTrails as a web browser extension	**AR4**: SearchTrails should be realized as a browser extension. **AR19**: SearchTrails should not limit its users to a certain database and allows free interaction with all resources of the Internet.	**HL3** 'Interact'
	F6: Do not force user interaction	**AR9**: SearchTrails should be unobtrusive.	**HL3** 'Interact'

Scope and features of	**F7**: Generate the trail from the user's interaction with the entire Web	**AR5**: SearchTrails should be able to capture data from all web pages and not limit its users.	**HL1** 'Capture'
	F8: Generate data, but require only little user interaction	**AR16**: The artifacts created by SearchTrails or by the user in should include the search trail, highlights, clusters, keywords, and marked nodes.	**HL3** 'Interact'
	F9: Users may interact with the search trail	**AR8**: The users should be able to access the search logs interpreted as search trails. **AR7**: The user should be able to interact with the generated search trail via mouse and keyboard commands.	**HL3** 'Interact'
Interaction with search trails	**F10**: Visualize the search trail as a force-directed graph	**AR21**: SearchTrails should make use of a force-directed graph visualization for the search trail.	**HL2** 'Visualize'
	F11: Enrich the search trail nodes by textual annotations	**AR23**: SearchTrails should enrich the force-directed search trail graph by textual representations of the respective web pages.	**HL5** 'Extend'
	F12: Use clustering for structuring the search trail	**AR22**: SearchTrails should visually cluster the nodes based on the website hosts and offer the possibility to shrink and expand the clusters.	**HL3** 'Interact'
	F13: Store the search trail in regular intervals	**AR10**: The search trail data storage should be realized on a remote server and ensure regular storage intervals. **AR11**: SearchTrails should enable the users to exchange unfiltered search trails.	**HL4** 'Store / Retrieve'
	F14: Generate a search trail ID for continuing search trails	**AR13**: SearchTrails should provide a way to allow working on search trails from any place. **AR12**: SearchTrails should allow catching up with previous search processes by recreating and evaluating search trails.	**HL4** 'Store / Retrieve'
	F15: Allow continuing and extending the search trail	**AR14**: SearchTrails should allow single searchers to split their search into single sessions.	**HL5** 'Extend'
	F16: Enable asynchronous collaboration	**AR17**: SearchTrails should allow remote asynchronous collaboration of users by the exchanging of search trails.	**HL5** 'Extend'

In the following subchapter, I develop the concept of the software system SearchTrails. I describe SearchTrails as a series of screens, each offering certain possibilities for user interaction.

3.3 Concept of SearchTrails

In Chapter 3.1, I work out 23 abstract requirements for SearchTrails based on the conclusions drawn from the motivating scenarios and the related work. Chapter 3.2 clusters the abstract requirements and derives a set of 16 technical features from them. In this subchapter, I unite the set of the distinct technical features F1 to F16 into an overall high-level mockup of the SearchTrails system.

As SearchTrails is realized as a web browser extension, Feature F5 is ensured. The realization of SearchTrails as a web browser extension enables fulfilling a set of features: It enables unlimited user interaction with the Internet (Feature F2). The extension has access to the user's interaction with the browser and generates the search trail from this (Feature F7). Besides the user interaction, the search trail can automatically be enriched by information captured from the visited web pages (Feature F8). Realizing SearchTrails as a web browser extension allows running a storage mechanism in the background for storing the search trails on a remote log storage server (Feature F13). Figure 38 shows how SearchTrails exists besides the normal web browser functionality and uses the Internet connection for storing the search trails remotely.

Figure 38: SearchTrails exists besides the normal web browser functionality.

Internally, the browser extension SearchTrails runs as a separate tab besides the normal browser tabs, which does not force user interaction and allows users to still follow their usual practices of tabbed browsing (Feature F6, Figure 39), but allows also returning to the SearchTrails tab during the search process and to check the current status of the search trail and its information content.

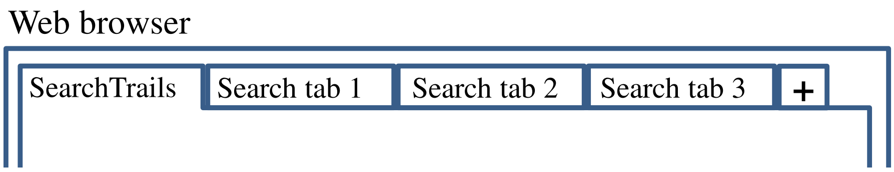


Figure 39: SearchTrails as a separate browser tab.

In order to allow the user to interrupt the work with SearchTrails without having to stop SearchTrails completely, SearchTrails provides a button besides the browser address bar, which changes its mode depending on the state SearchTrails is currently in. When SearchTrails is inactive, a click on the button starts SearchTrails and puts it into the 'Active' mode. A click on the button in the 'Active' mode offers the option to 'Pause' SearchTrails or to 'Stop' SearchTrails.

Stopping SearchTrails puts SearchTrails back into the ‘Inactive’ mode. Pausing SearchTrails interrupts the logging, but leaves the SearchTrails tab open. A click on the SearchTrails button during the ‘Pause’ mode brings SearchTrails back to the ‘Active’ mode again (Figure 40).

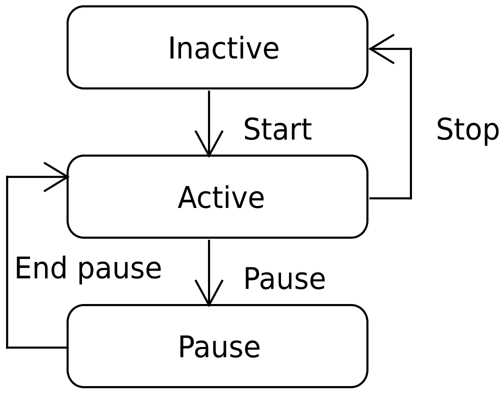


Figure 40: UML diagram of SearchTrails’ states and their transitions.

Within the SearchTrails browser tab, the described features are separated into a number of subpages to avoid cluttering the visualization with too many features. Most important is the standard ‘Home’ page, showing the main part of the visualization. The ‘Home’ page is followed by the ‘Highlights’ page, showing the highlights overview. A ‘Settings’ page supplies the user with important information and with options for configuring the search trail. The ‘Continue a search trail’ page offers the possibility to retrieve and visualize an existing search trail for evaluation and / or extension. The following sections explain which functions are visible on what page of SearchTrails.

‘Home’

The ‘Home’ page is the most important page of SearchTrails. This page mainly features the search trail visualization and the keywords overview. The search trail visualization realizes a set of the aforementioned features, which are the following:

- Feature F1: Transform user navigation into a search trail.
- Feature F9: Users may interact with the search trail.
- Feature F10: Visualize the search trail as a force-directed.
- Feature F11: Enrich the search trail nodes by textual annotations.
- Feature F12: Use clustering for structuring the search trail.

The panel on the right side of the ‘Home’ page shows the list of keywords, which are extracted from the visited web pages (Feature F3). The keywords are extracted from the visited pages at the runtime of the system. Keywords may be clicked by the user. This highlights the clicked keyword in the list and highlights all nodes with the corresponding keyword in the search trail (Figure 41, with the search trail nodes corresponding to the keywords ‘Apple’ or ‘Finder’ highlighted in green). This way, users have the chance to quickly discover all pages related to a certain keyword. Additionally, the ‘Home’ page features a search button, which automatically opens a new browser tab with a web search for the recently highlighted keywords. Therefore, the keyword list can be used to trigger searches on previously unknown keywords.

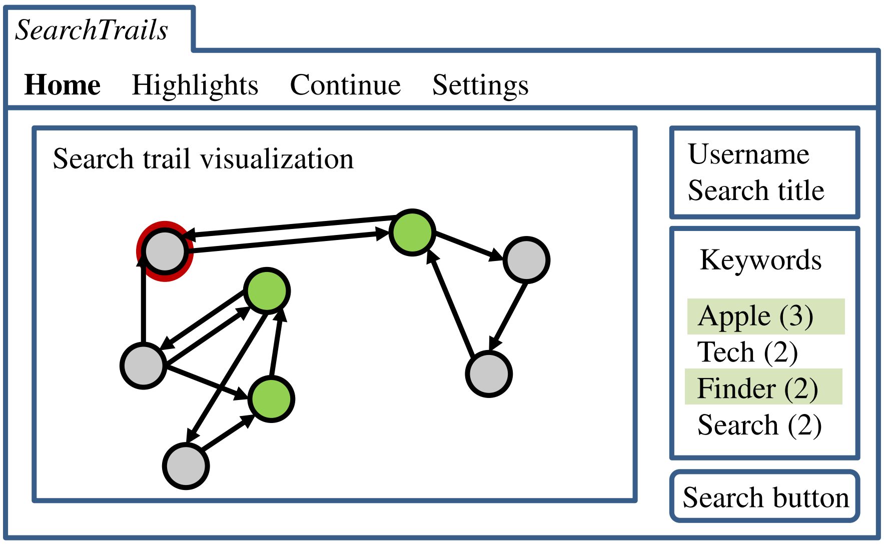


Figure 41: Schematic view of SearchTrails’ home page.

‘Highlights’

The ‘Highlights’ page collects all valuable resources found during the search process. This page is essential for the synthesis of the information found. The highlights can be included into the highlights overview by highlighting the according text on the web page and pressing a specially defined button. This copies the selected text into the highlights overview, and also marks the according node in the search trail. Figure 42 shows how the selected text from web page ‘Page 1’ gets included into the highlights overview. The highlights overview shows for each web page the collected highlights (allowing the collection of multiple highlights from the same page), the host of the page, and features a direct link to the original page. Additionally, buttons for removing highlights are included as well as buttons for manually adding comments to highlights of a certain web page, allowing adding comments for each web page where a highlight was found.

The highlights overview is ordered chronologically and is displayed with the search trail still being visible below the highlights view. Similarly, the keywords are still visible when the highlights are displayed. This page realizes feature F4.

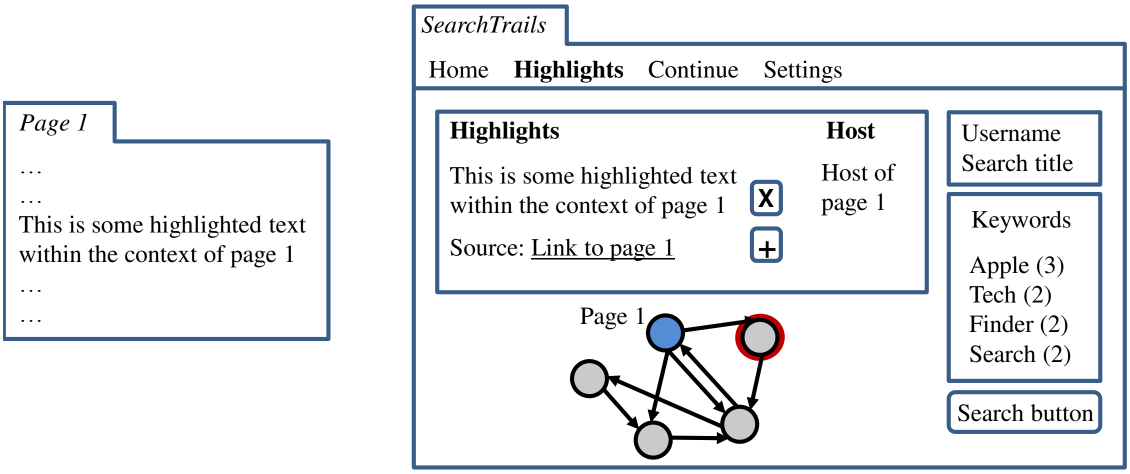


Figure 42: Schematic view of SearchTrails’ highlights page.

‘Settings’

The ‘Settings’ page combines all important settings for the use of SearchTrails. It first displays the unique search trail ID, which is generated at every start of SearchTrails, before the search trail gets stored. This ID can be used by the searcher to recreate a search trail after a period of time (Feature F14) or for forwarding the ID to another user. On the ‘Settings’ page, the searcher can enter a username and a title for the search. This may help other searchers to identify the broad topic of the search trail more easily. The ‘Settings’ page furthermore features a checkbox, which enables or disables the remote storage mechanism. If the search trail is not stored remotely, it needs to be possible to store the search trail locally (cf. ‘Other pages’ below). The ‘Settings’ page still shows the search trail and the keyword list, but also displays the username and the title of the search (cf. Figure 43).

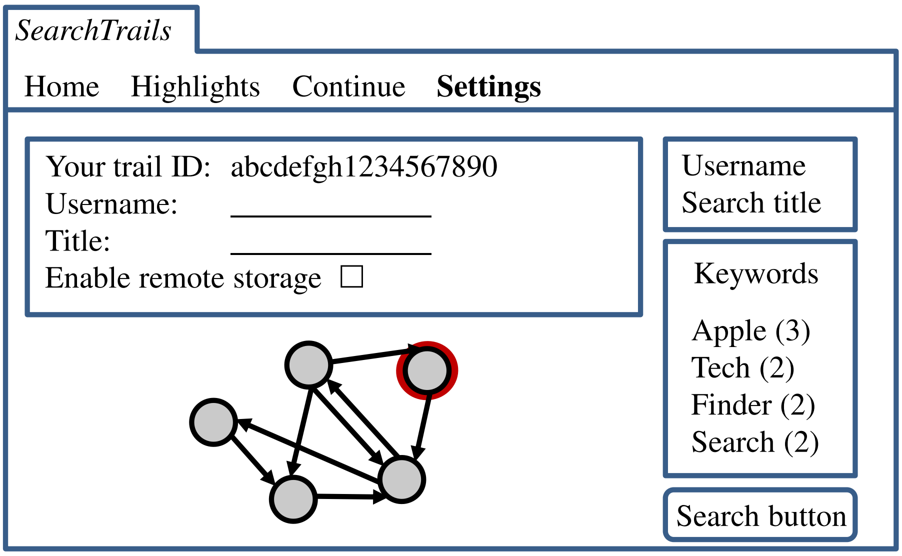


Figure 43: Schematic view of SearchTrails’ settings page.

‘Continue a search trail’

The ‘Continue’ page offers the users the possibility to continue a search trail by both a given search trail ID or a search trail data object. In case a search trail ID is entered, the according search trail is requested from the remote log storage server. The retrieved search trail is rendered by SearchTrails, and all keywords and highlights are recreated. The user can then continue and extend the search trail (Feature F15). This page is the basis for enabling asynchronous collaborative search (Feature F16).

When the search trail data object was stored locally by the user, the search trail does not need to be fetched from the storage server. Instead, the search trail data object is loaded and rendered by SearchTrails. The ‘Continue’ page additionally shows the search trail and the keywords, such that the user can easily check whether the trail was loaded correctly (cf. Figure 44).

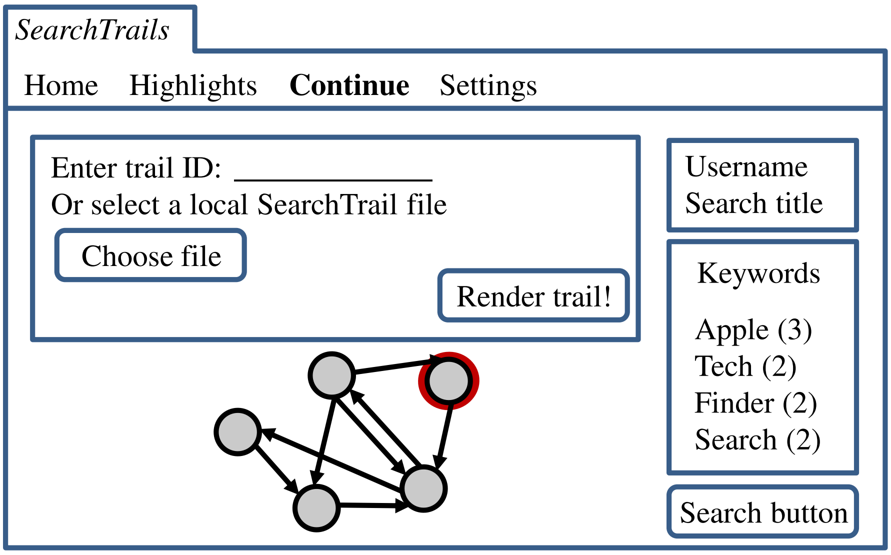


Figure 44: Schematic view of SearchTrails’ continuation page.

Other pages

It may be necessary to include other pages into the SearchTrails user interface. These pages can be explicitly intended for the users of SearchTrails, or they can be intended for the developer of SearchTrails.

For supporting the users of SearchTrails, a 'Help' page is needed. The help page explains the overall functionality of SearchTrails and its different features. The help page explains the color coding of the search trail nodes and the keyboard commands for creating a highlight or for marking a page as valueless. Furthermore, the page explains which nodes may be deleted and which not. For developing and testing purposes, a 'Show trail data' page may be needed. The page shows the underlying data from which the search trail is created. In case a user does not want to store the search trail remotely, the search trail data can be downloaded from that page to the local hard disk. Recreating the search trail from the trail data is possible by loading the search trail data on the 'Continue' page.

In this subchapter, I build upon the technical features developed in the previous subchapter and integrate these features into a prospective user interface of SearchTrails. The subchapter therefore provides a first picture of the functions of and the possible user interactions with SearchTrails. In the next subchapter, propose a rough architecture of SearchTrails by identifying its logical components.

3.4 Rough architecture of SearchTrails

In the previous subchapters of this chapter, I more and more consolidated the different insights gathered in Chapter 1 and Chapter 2 of this thesis. First, I derive requirements from the related work. Second, the requirements are clustered to form certain features of the envisioned system SearchTrails. Third, the developed features are transformed into a high-level concept of the SearchTrails user interface. In this subchapter, I develop a rough architecture from the user interface mockup. The architecture can be considered rough, as the technical realization may have to be more complex to achieve the proposed functionality.

Figure 45 shows the rough architecture of SearchTrails. This view already reveals the logic behind the user interface. While the user interface is divided into the four main pages ('Home', 'Highlights', 'Continue a search', and 'Settings') and two supplementary pages ('Show search trail data' and 'Help'), the inner logic of SearchTrails is divided into three components, with one additional component residing in the normal browser functionality. The three components are the following:

- **Logging engine**
  The logging engine captures information about the user's interaction with the browser. It gets informed about all changes of opened or closed tabs, changes of the displayed URL, or the status of the visited web pages with respect to the loading of pages.
  The logging engine interprets the browser events and creates the data object underlying each search trail, containing information about the nodes, their metadata, their successors, and their follower nodes. The logging engine furthermore retrieves information

about the page's keywords and stores this information in the node's data object. Similarly, the web page highlights are retrieved by the logging engine and stored with the corresponding node in the search trail data object.

- **Visualization engine**
  While the logging engine is responsible for creating und updating the search trail data object, the visualization engine is responsible for all user interface functionality. As such, it builds the user interface consisting of the four main pages and the two supplementary pages. It is also responsible for creating the start/stop/pause buttons of SearchTrails, which have to be integrated into the overall browser structure.
  The visualization engine builds all SearchTrails page content from the information in the SearchTrails data object. This data object is served by the logging engine upon every change of the data object. Any change of the search trail that is made in the visualization is fed back to the logging engine and results in a change of the data object.

- **Storage engine**
  The storage engine is responsible for the regular saving of the search trail data object. This is achieved by a customized data storage infrastructure. When a user enables the remote storing of the search trail, the search trail data object is sent with its newly created, unique ID to the storage server, where it is stored under the respective ID. In regular time intervals and when SearchTrails is ended by the user, the storage engine requests the recent copy of the search trail data object from the logging engine and stores it on the storage server. In case the new search trail data object differs from the previously stored version, the old version gets updated by the new one.
  When a user requests a search trail by its ID, the storage engine contacts the storage server and retrieve the data object from it. In this case, the given search trail ID is also used for storing the updated data object, which results in updating the retrieved data object on the search trail storage server.

As an additional component, the page data retrieval engine is needed. This component is used for extracting data from all visited web pages and sending this data to the SearchTrails logging engine. In order to generate the keyword table, SearchTrails needs to evaluate the content of all visited web pages to extract the keywords from these pages. To achieve this, every page has to be evaluated for its keywords. Technically, this has to be done by code that is executed for every single web page in every browser tab. Similarly, the code in the data retrieval engine enables the pages to send the highlights with their respective metadata back to the logging engine. The data retrieval engine code executed for every visited web page enables SearchTrails to catch the user's highlighted text upon a special key combination and to store this data within the search trail. The data retrieval engine is a component that is injected into each web page and is therefore more a part of the web browser functionality.

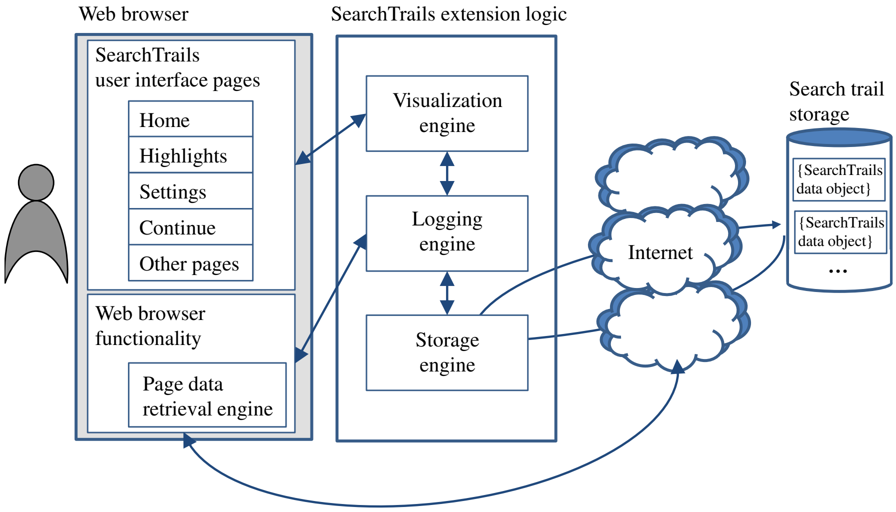


Figure 45: SearchTrails' rough architecture.

3.5 Conclusion and outlook

This chapter connects the related work and the actual implementation of SearchTrails. A set of abstract requirements from the motivation and the evaluation of the concluding tables frame the scope of the SearchTrails system. SearchTrails is implemented as a web browser extension features a force-directed graph visualization. A set of technical features ensures the support for complex search support and the interaction with search trails. A set of four high-level user interface views demonstrates the core elements of the technical realization of SearchTrails, which are the search trail visualization, the highlight overview, basic settings, and the possibility to continue a search trail. The architectural view incorporates both the user interface and the logical background of SearchTrails.

Considering the research questions developed in Chapter 1, this chapter can add information to answering research question RQ1.1 'Search trails valid?' and RQ1.2 'Which functions?'. For answering RQ1.1 'Are search trails a means to resemble the course of a user's web search?', this and the next chapter develop the core ideas for a system to transform user interaction into search trails for supporting collaborative search processes. Chapter 3.2 presents a set of functional requirements for answering RQ1.2 'Which functions does a system need to offer to the users to support capturing both the context as well as key findings of a web search?'.

In the next chapter, I describe the process of the technical implementation of SearchTrails. Beginning with the first versions, which are not fully functional, the Chapter 4 describes the iterative process of developing technical solutions which finally enable the realization of SearchTrails.

Chapter 4
Implementation

The first three chapters of my thesis paved the way for the technical realization of SearchTrails, which I describe in this chapter. In Chapter 1, I motivate the idea of SearchTrails with the help of motivating scenarios and work out the problems to overcome during single user and collaborative complex search processes. Chapter 2 evaluates a range of related work, from historical foundations and theoretical approaches to an overview of systems for logging search processes and supporting collaborative search processes. Chapter 3 builds upon the first two chapters by evaluating the motivation and the related work. I evaluate the comparing tables of the related work and develop a set of abstract requirements for implementing SearchTrails. These abstract requirements are transformed into a set of technical features for SearchTrails. These technical features lead to a more detailed concept of SearchTrails, which is described as a series of user interface screens. Finally, Chapter 3.4 presents a rough architecture of SearchTrails.

In this chapter, I describe the implementation process of SearchTrails, starting from the abstract architecture and the more detailed technical features of the previous chapter. This chapter is divided into four subchapters. In Chapter 4.1, I explain the final architecture of SearchTrails and its internal components in more detail. In Chapter 4.2, I describe the final implementation of SearchTrails, explain technical design decisions and introduce a set of frameworks and tools which are used for the implementation of SearchTrails. Then I present details of the search trail data object and the function of the search trail remote storage mechanism.

In Chapter 4.3, I present the design iterations made with SearchTrails. I first present paper prototypes that have led towards the development of concepts for SearchTrails. I then introduce an initial prototype I developed at a very early stage of the thesis which has some substantial differences compared to SearchTrails. I then explain the progress made during the implementation of SearchTrails during the first and the second iteration of the development cycle. Especially the Chapter 4.3.3 and 4.3.4 can provide hints towards answering research question RQ1.2 ‘Which functions?’, as they describe the implemented features of the implemented versions of SearchTrails. The last subchapter concludes this chapter.

4.1 SearchTrails architecture

In this subchapter, I present the final architecture of SearchTrails. In contrast to the rough architecture presented at the end of Chapter 3, this version of the architecture includes details on the respective frameworks that are responsible for achieving the desired functionality. Before explaining the components that contribute to the architecture of SearchTrails, I describe the architecture itself.

The overall architecture of SearchTrails is split into server-side and client-side components. On server side, there exist the Node.js-based search trail data storage service deployed at the

Platform-as-a-Service provider Heroku and the BSCW workspace server which serves as the search trail storage. A more detailed overview of the integrated services and frameworks can be found in Chapter 4.2.2. The search trail data objects are stored to the BSCW as described in Chapter 4.2.4. All communication between the server-side components is made via HTTP requests (cf. the black arrows in Figure 46).

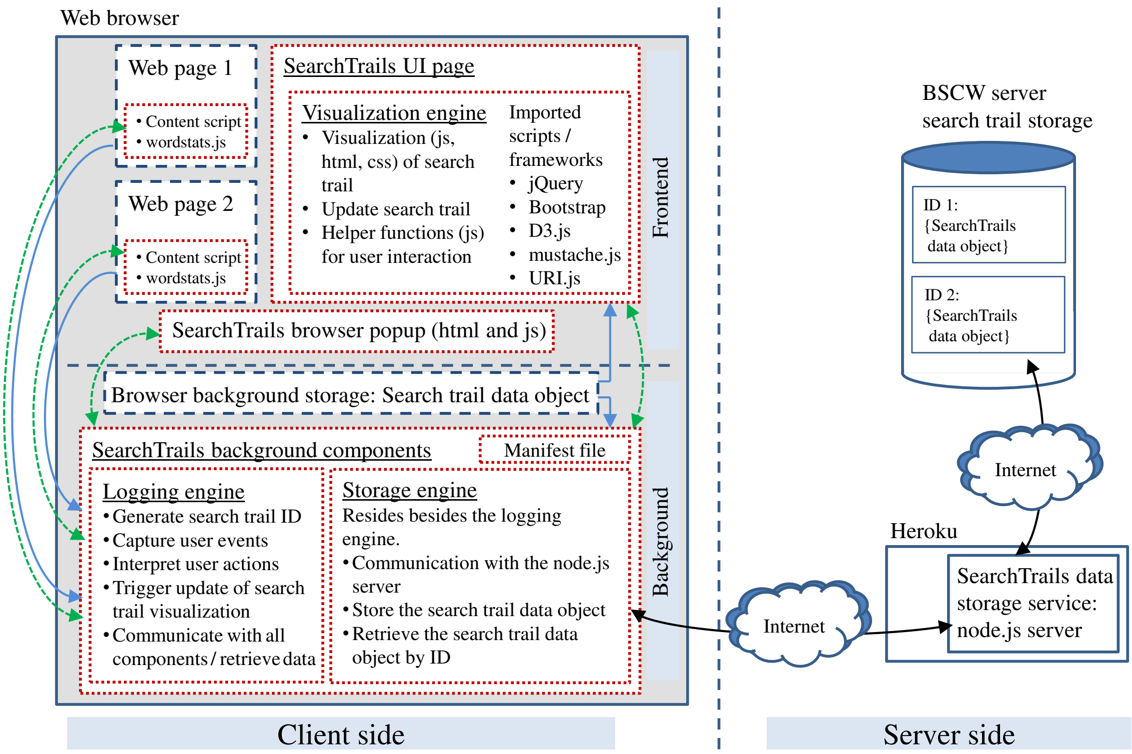


Figure 46: Detailed architecture diagram of SearchTrails.

On client side, several components exist in each SearchTrails instance. A manifest file serves as the connection between the browser and the extension itself. It contains data about which components belong to the extension, which permissions the extension requests from the user, and defines some standard behavior for the extension. The most important components for SearchTrails are the logging engine, the visualization engine, and the storage engine. Additional components are the content script and both the HTML file and the according JavaScript files for the SearchTrails browser pop-up.

At the start of SearchTrails, a new tab is opened containing the visualization engine. Simultaneously, the logging and the integrated storage engine are started as SearchTrails extension background scripts.

Therefore, SearchTrails consists of several different client-side components interacting with each other. These components are the following (cf. Figure 46. The components with the red dotted frame are created or delivered by SearchTrails, while the components with the blue dashed frames are native browser components):

- The SearchTrails manifest file, which declares SearchTrails' components to the web browser.

- The SearchTrails browser pop-up, which offers controls to start, pause, or stop SearchTrails.
- The SearchTrails visualization, running in a separate browser tab. The visualization page contains the visualization engine.
- The SearchTrails background components, consisting of the logging engine with the integrated storage engine. The logging engine transforms the user's interaction with the browser into a search trail, which is stored on the remote search trail storage server by the storage engine.
- A number of browser web pages which are each opened in separate tabs of the browser. Each tab contains a content script that extracts the keywords from the visited web pages.
- The browser background storage, which keeps the search trail data object and serves as a link between the visualization engine and the SearchTrails background components.

All these components technically exist within different processes of the operating system, such that several means of communication between these components are necessary. Interaction between the client-side components of SearchTrails happens either by registering listeners for certain events or by message passing between SearchTrails components. Browser extensions can register listeners for browser events, such as the interaction with browser functions, interaction with browser tabs, or for the change of storage objects (cf. the blue arrows in Figure 46). After registering a function to a listener, the respective function is called each time the event happens. SearchTrails registers listeners for the following purposes:

- The logging engine registers listeners to get informed about all changes of browser tabs, their creation, change of URL, switching between tabs, and the closing of tabs.
- Both the logging engine and the visualization engine register listeners to get informed when the respective other component has changed the search trail data object in the browser's background storage.

When the implementation of listeners is not feasible, information is shared between the different parts of SearchTrails by message passing. This mechanism is offered by the browser to enable interaction between components that do not reside in the same logical space. As several components of SearchTrails exist in different logical spaces, message passing is the solution that allows interaction between these components (cf. the green dashed arrows in Figure 46). Message passing is needed between the following components:

- Interaction between the pop-up script and the SearchTrails background components.
  The pop-up script informs the logging engine whether SearchTrails was started, paused, or closed.
- Interaction between the web page tabs and the SearchTrails background components.
  The content script injected into all web pages sends back the web page's keywords to SearchTrails' logging engine.
- Interaction between the SearchTrails background components and the SearchTrails visualization.
  When a user changes parts of the search trail, these changes are passed towards the SearchTrails logging and storage engine.

This combination of communication mechanisms enables the interplay of the different components of SearchTrails. The following sections explain all SearchTrails specific components and their functionality.

SearchTrails browser pop-up

The SearchTrails browser pop-up is registered in the manifest file of the SearchTrails extension. It is responsible for creating a SearchTrails icon next to the browser address bar, where a click triggers the start of SearchTrails. Additional clicks on the SearchTrails icon offer the possibility to pause or end a pause of SearchTrails or to close the extension, according to the state diagram from Chapter 3.3, extended by a graphical state indicator (cf. Figure 48). Figure 47 shows the pop-up when SearchTrails is active and offers the options to pause or end SearchTrails.

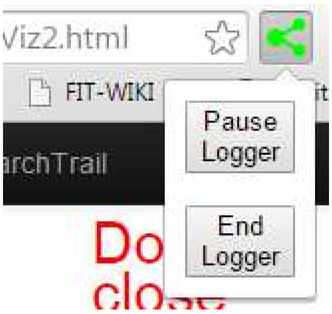


Figure 47: SearchTrails browser pop-up and user options.

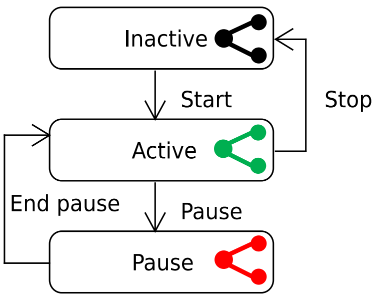


Figure 48: SearchTrails' states and icon color coding.

Content script

The content script gets injected into every tab the browser is maintaining. For each web page, the content script is executed after loading. By making use of wordstats.js, the content script extracts the five most important keywords of every visited page. The content script sends the five most important keywords via message passing to the SearchTrails logging engine, where the keywords get attached to the node representation in the search trail data object and get integrated into the search trail visualization and the keyword table.

Logging engine

The central part of SearchTrails is the logging engine, which is started from the SearchTrails extension manifest file and runs in the background in an own process of the operating system. At the start of SearchTrails, the logging engine registers a number of listeners for several browser events, such as the opening, updating, and closing of new browser tabs, and the activation of tabs. This way, the logging engine gets informed about all user interaction with the browser tabs. The logging engine receives the corresponding events and forwards them to the interpretation logic, where the user interaction gets translated into the search trail data object. User events that result in a visible change of the browser content, such as the opening of a new tab, the switching between tabs, or following a link trigger a change of the search trail. In these cases, the search trail data object gets updated. However, the description and the title of a page are frequently not present during the loading of a site, thus the new node gets created without this information in these cases. This helps making SearchTrails reactive for the user, as new nodes are created immediately upon visiting a new web page. Upon the update event of a browser tab, the newly contained information gets extracted and the search trail data object gets updated. Depending on the web page visited, these events can occur multiple times and need to be filtered out. Each time the search trail data object gets changed, it is written to the browser's background storage.

Background storage

The background storage manages data objects from all extensions in a sandbox and serves as a link between the SearchTrails background components and the visualization. Whenever the search trail data object is changed by the logging engine, it is stored to the background storage. The listeners registered by the visualization engine are triggered at any change of the background storage. When the change event's name implies a change of the visualization, the visualization engine gets active and updates the visualization.

Similarly, the visualization engine may also trigger changes of the search trail force-directed graph visualization, e.g. when a user deletes a highlight or a node. In these cases, the visualization engine takes the search trail data object from the background storage, and adds a remark that a user has changed the search trail data object, which only affects the visualization. Upon this remark, the visualization gets changed but the basic search trail data object stays the same. Still, the logging engine gets informed to ensure that the most recent copy of the search trails data object gets stored by the storage engine.

Visualization engine

The visualization engine manages the transformation of the search trail data object into the force-directed graph visualization and handles all user interaction with the visualization. The visualization engine registers a listener on changes of the browser's local storage. Upon a change of the search trail data object in the browser's local storage, the visualization gets updated. Technically, the visualization is done by the D3.js framework, which generates a force-directed graph that is fully interactive for the user. The force-directed graph allows hovering over nodes and creates pop-ups on the hovered nodes; it enables dragging the search trail and rearranging it dynamically. The SearchTrails visualization also enables double-clicking on clusters, which results in closing and expanding the clusters. The user's interaction only triggers changes to the visualization of SearchTrails, but no structural changes to the underlying search trail data object. For example, when a user closes a cluster, a new cluster node replaces all other nodes from within the cluster. When the user visits more pages within the cluster, the search trail data object still gets updated and new nodes are created within the closed cluster. Figure 49 shows a cluster of web pages from the New York Times (left), which gets closed by the user (middle). As the user visits a new page from the same host, one new node is created within the cluster. This node gets visible upon expanding the cluster. The new node has led the user to a page outside the cluster (right).

Figure 49: Search trail with cluster indication (left), closed cluster (middle), and reopened cluster after ongoing browsing (right).

For reasons of data consistency, the logging engine gets informed about all changes of the visualization of the search trail data object via message passing.

Storage engine

The storage engine technically lives in the same process as the logging engine. When the user activates the remote storage mechanism, the storage engine stores the search trail on the BSCW server by sending it to the search trail data storage engine, and updates the stored search trail data object at least every five minutes to avoid losing much data. When the storage process is started, the search trail data object is taken from the background storage and the search trail data object is sent to the Node.js server together with the unique search trail ID. The search trail ID consists of a 20 digit random alphanumerical code together with the UNIX timestamp of the creation of the ID. The Node.js server then stores the search trail data object on the BSCW search trail storage server under the respective ID. Changes of the search trail data object are automatically detected by the BSCW server and result in replacement of the according file.

When a search trail is requested from the storage server by its ID, the storage engine sends the ID to the search trail data storage engine, which retrieves the according file from the BSCW search trail storage and sends it back to the storage engine with the help of a polling mechanism. Upon the reception of a search trail data object, the data object is loaded into the logging engine and the visualization is started.

This subchapter presents the architecture of SearchTrails with all its technical components, the interplay of server-side and client-side components and the communication within SearchTrails. The next subchapter explains the technical details of the integrated frameworks.

4.2 Technical implementation decisions and integrated services and frameworks

This subchapter presents details on the implementation of SearchTrails. The most important tool for developing SearchTrails is the Google Chrome browser, which serves as the container for the browser extension. I first explain why I chose this browser to develop Search-

Trails for. In order to realize the functionality described in the previous chapters, Google Chrome supports a number of tools and frameworks. I explain which tools and frameworks I chose for the implementation. After that, I introduce the search trails data object and explain which frameworks and mechanisms contribute to the remote search trail storage mechanism.

4.2.1 Selected browser and browser extension details

My analysis of the related work shows that all recent search logging systems rely on a web browser extension. Therefore, I chose to realize also SearchTrails as a web browser extension. I decided to choose a web browser that is common among the potential users of SearchTrails and that is uncomplicated to handle for development. Figure 50 shows the distribution of market shares of the five major web browsers Google Chrome, Microsoft Internet Explorer, Firefox, Safari, and Opera. The data are taken from the logs of the w3schools website[59]. This chart can explain nearly 100% of the different browser types (sum line in Figure 50).

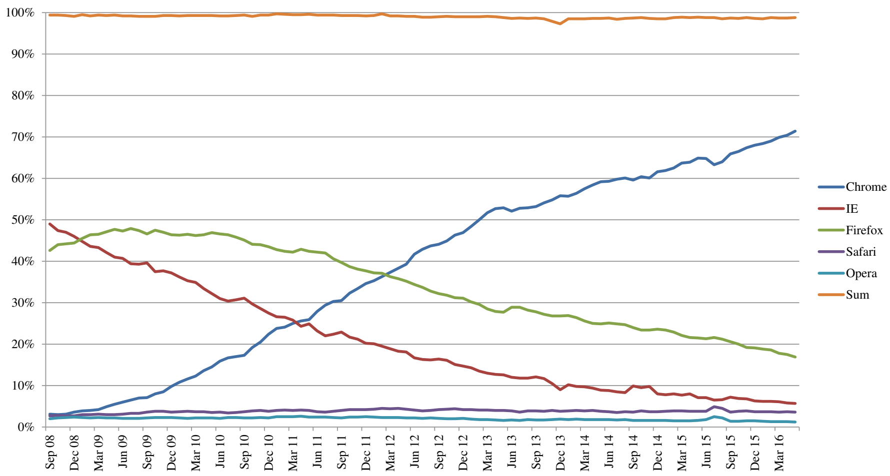


Figure 50: Overview of the leadings browser's market shares[60].

It turns out that Google Chrome possesses a leading market share since March 2012, and crossed the 50% market share threshold in February 2013. As of May 2016, Google Chrome's market share has increased to 71%, with its closest contestant Mozilla Firefox having a 17% market share. For the development of SearchTrails, I compared the possibilities of developing extensions for both the Firefox and the Google Chrome web browser, and finally decided to implement SearchTrails as a Google Chrome browser extension. This is due to the major market share and the less complex development for Google Chrome compared to Mozilla Firefox. For Mozilla Firefox, the extension's manifest file needs to be updated for each new version of the browser, which makes the extension refuse working when a new version of Firefox gets released. In contrast, Google Chrome does not require specifying a certain maximal browser version up to which the extension will be supported. Additionally, the internal application structure of a Mozilla Firefox extension is more tedious than the internal structure of a Google Chrome extension.

---

[59] Retrieved June 20, 2016 from http://www.w3schools.com/browsers/browsers_stats.asp
[60] Own visualization, data retrieved June 20, 2016 from http://www.w3schools.com/browsers/browsers_stats.asp

For the creation of a Google Chrome extension, some files need to be created. It would be only of limited value to elaborate on all files that contribute to the SearchTrails extension. Therefore, I just mention the ones that are mandatory for the creation of the browser extension. These files are the following:

- A manifest file named ‘manifest.json’. This file keeps basic information, is used to request permissions for accessing specific data from the user and defines a set of resources to be loaded for the extension. Basic information consists of the name and the description of the extension, and the current version of it.

  The extension needs to ask permission from the user to be able to access certain data. Examples for such permissions are ‘history’, when accessing the browser’s history; ‘tabs’, when accessing the browser’s open tabs and their metadata, or ‘storage’ when the browser’s local storage shall be used. This list also includes permissions for the extension in the form of URLs for establishing connections to other servers. In case of SearchTrails, this list includes a permission to access the remote search trail data storage engine, which handles the storage of search trails on the BSCW server.

  The resources to be loaded are the links to the content scripts, which are injected into all visited web pages, and the background scripts, which are executed while the browser extension is active.

- An icon, which is displayed as the extension logo in the browser bar. Normally, this file is called ‘icon.png’, and can be located in the extension’s root folder, but also in subfolders, if the path to the resource is set in the manifest file.

- A ‘popup.html’ file, which contains the content which is shown when clicking on the extension’s logo in the browser bar.

- All other files are optional and may be included in an extension-specific file structure. Additionally, web pages being created in an extension may include other online resources, such as links to commonly used programming frameworks.

This section explains why I chose Google Chrome as the browser to implement SearchTrails for. The section furthermore explains some necessary components for setting up the browser extension.

4.2.2 Integrated services and frameworks

This section describes the services and frameworks that are integrated into SearchTrails and which contribute to its overall functionality and appearance. The services and frameworks reach from services for storing data with the help of cloud-based Platform-as-a-Service providers to standard frameworks for elaborated JavaScript functionality and layout frameworks. More specialized frameworks are integrated for data visualization and word frequency extraction. The integrated services and frameworks are the following:

- **BSCW – Basic Support for Cooperative Work**[61]
  BSCW is a web-based shared workspace system, offering a folder structure with elaborate functions for team collaboration. I chose to work with BSCW as it is uncomplicated to access by the BSCW API and allows convenient user access to the stored files via a comfortable web user interface. BSCW has the advantage that it can be configured to overwrite existing objects and allows the remote creation of objects via its API. BSCW makes it especially easy to manage large sets of files, e.g. the storing and securing of the search trails generated during the user studies (cf. Figure 55). The search trail data objects are stored on a German server and are password protected, thus ensuring privacy of the stored data.

- **Heroku**[62]
  Heroku is a cloud application platform, which is a Platform-as-a-Service (PaaS) provider. Heroku offers support for a range of languages and features support for a large number of frameworks, including Node.js. Alternatives to Heroku would have been setting up an own server, or relying on other PaaS services, such as Amazon Web Services[63] or the Google Cloud Platform[64]. I chose to use Heroku, as it turned out to be uncomplicated and free for the limited computing power needed for the SearchTrails data storage service.

- **Node.js**[65]
  Node.js is a server-side runtime environment, which can be deployed on a number of different platforms. It is open-source and allows the easy creation of server-side applications. The name already suggests that all applications are written in JavaScript. For reasons of cross-origin resource sharing (CORS), Node.js is used to build the server that receives the search trails from the storage engine and manages the storage on the BSCW server. Furthermore, this server also manages fetching search trails by their ID.

- **jQuery**[66]
  jQuery is a cross-platform JavaScript library for client-side use. jQuery eases the access to and manipulation of the HTML document object model (DOM) elements. Furthermore, jQuery offers easy methods for starting asynchronous JavaScript and XML (AJAX) calls. jQuery is integrated in most modern websites[67] and is required by a number of other JavaScript libraries. SearchTrails needs jQuery as a prerequisite for other frameworks (such as Bootstrap), and also for a number of internal functions.

- **Bootstrap**[68]
  Bootstrap is a framework for web page user interface design. Bootstrap comes with a large number of features and components, which are not all used within SearchTrails. Most important for SearchTrails are the Bootstrap style information (coming as Cascading Style Sheet (CSS) information) which allows the easy construction of Search-

[61] Retrieved August 27, 2015 from http://www.bscw.de/english/
[62] Retrieved August 27, 2015 from https://www.heroku.com/
[63] Retrieved August 26, 2015 from http://aws.amazon.com/elasticbeanstalk/
[64] Retrieved August 26, 2015 from https://cloud.google.com/
[65] Retrieved August 27, 2015 from https://nodejs.org/en/about/
[66] Retrieved August 26, 2015 from https://jquery.com/
[67] Retrieved August 26, 2015 from http://w3techs.com/technologies/overview/javascript_library/all
[68] Retrieved August 26, 2015 from http://getbootstrap.com/

Trails' pages, and the additional Bootstrap JavaScript jQuery plug-ins, enabling basic user interface functions, e.g. pop-ups.

- **Data-driven documents (D3.js)**[69]
  D3.js is a JavaScript library for generating dynamic data visualizations in the user interface of web pages. The visualizations created by D3.js are dynamic and interactive, such that users can interact with the generated visualizations. This is achieved by D3.js relying on HTML5, Scalable Vector Graphic (SVG), and CSS standards. Furthermore, D3.js contains methods for generating convex hulls around nodes, or for creating arrays of canonical colors. D3.js allows control over a large number of properties of each visualization element.
  SearchTrails uses D3.js for generating the force-directed graph visualization. Several properties of the nodes are controlled by SearchTrails. The clusters around nodes of the same host are computed by the convex hulls around the nodes of the force-directed graph. As D3.js makes use of SVG objects, all parts of the search trail are individual SVG elements, where data can be attached or events can be generated. This allows freely configurable user interaction with the search trail itself.

- **Mustache.js**[70]
  Mustache is a templating engine, which features implementations for all major coding languages. Mustache.js is the JavaScript implementation of mustache which allows using any string with placeholders as a template to be filled with content. SearchTrails needs predefined HTML fragments within the JavaScript code and uses Mustache for replacing placeholders in the HTML code by actual content.

- **URI.js**[71]
  URI.js allows convenient access to all possible properties of URIs, as defined in the RFC 3986[72] standard from January 2005. URI.js can be both Uniform Resource Names (URNs) and Uniform Resource Locators (URLs). While URNs provide a persistent unique name to a resource, URLs locate given resources and therefore enable lookup of the resources. In case of SearchTrails, URLs are most relevant. RFC 3986 defines the structure of URLs in Section 3 as follows (cf. Figure 51):

```
foo://example.com:8042/over/there?name=ferret#nose
\_/   \______________/\_________/ \_________/ \__/
 |           |            |            |        |
scheme     authority       path        query   fragment
```

Figure 51: Structure of a URL according to RFC 3986.

In order to avoid reimplementing parsing mechanisms for URLs, URI.js was integrated into SearchTrails. SearchTrails uses URI.js for parsing URLs, extracting details like the hostname, or for removing fragment links from URLs.

[69] Retrieved August 26, 2015 from http://d3js.org/
[70] Retrieved August 26, 2015 from https://github.com/janl/mustache.js/
[71] Retrieved August 26, 2015 from http://medialize.github.io/URI.js/
[72] Retrieved August 27, 2015 from https://tools.ietf.org/html/rfc3986

- **Wordstats**[73]
  For extracting keywords from web pages, a mechanism had to be found that evaluates websites and derives keywords from the websites. WordStats achieves this by counting the number of words on a web page and by weighting the words based on the location in the HTML DOM. WordStats first parses the website and eliminates all words that can be found in a self-extended list of 1101 common terms in German and English (e.g. other, our, their, them,…). These words are considered as stop words and therefore excluded from the calculations. All meaningful words are stored in an array. Based on a word's location on the web page, a value is assigned to each occurrence of the word. For example, the occurrence of a word in an h1-heading is weighted with a factor of 15, while the occurrence in a bold or italic part of the text is weighted with a factor of 3. For each new occurrence of the word, the value is added to the array, from which the most relevant keywords are extracted after parsing the web page completely.
  SearchTrails injects the WordStats script into all visited pages, evaluates the keywords, and takes the five highest ranked words as the keywords for a page.

This section mentions all external libraries, frameworks, and tools that were used for realizing SearchTrails and explains their use in SearchTrails.

4.2.3 Generated data

In this subsection, I introduce the data generated by SearchTrails and collected in the search trail data object during the use of the final version of SearchTrails.

The search trail data object is stored as a JavaScript Object Notation (JSON) object. JSON is specified under two standards, namely RFC 7159[74] and ECMA-404[75]. While ECMA-404 specifies JSON in a very technical way, the RFC 7159 standard includes some additional suggestions for semantics and security of the data structure. Recently, JSON has become an almost standard approach for the lightweight exchange of information, due to its low formal overhead, especially when compared to XML. For this reason, JSON is used frequently for transferring data from representational state transfer (REST) web services.

The search trail data object contains on its first layer five different types of information (Figure 52). These refer to the formal definition of a search trail as set S = {U, T, I, N, R} (cf. Chapter 2.2.2). The data object S includes the username U and the title T of the search, which can be set individually by the searcher, and the search trail ID (here: 'logid'). The unique search trail ID (I) is generated upon the start of SearchTrails, based on a 20-digit alphanumeric random code, the letter 'd' and the UNIX date stamp of the time of creation. While these three properties of the JSON object are technically strings, the two properties 'nodes' (N) and 'recentChangeLog' (R) are more complex data structures (abbreviated in Figure 52). The 'nodes'-object is a

[73] Jean Francois Hovinne (2007). Wordstats. Retrieved August 26, 2015 from http://hovinne.com/articles/jquery-wordstats-plugin
[74] Retrieved September 01, 2015 from https://tools.ietf.org/html/rfc7159
[75] Retrieved September 01, 2015 from http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

dictionary, while the 'recentChangeLog'-object is an array. These two fields contain the information about the nodes in the graph, and the sequence of actions that lead to the search trail.

```
{
  username: "user_5",
  title: "3D-Printing",
  logid: "9sfj1hxqv95abbq6xwmrd1422310649086",
  nodes: {...},
  recentChangeLog: [...]
}
```

Figure 52: High level structure of the search trail data object.

Figure 53 shows an exemplary excerpt for a single node from a search trail data object. This data structure matches the definition of a node as a set of tuples as a set of tuples N = {(u, ti, Hi, Su, V, h)} from Chapter 2.2.2. Within the 'nodes'-dictionary, each node gets identified by its own URL. This has the advantage that for the creation of each node, it only needs to be checked whether an entry with the recent URL already exists in the 'nodes'-dictionary. Additionally, each search trail node contains the following information:

- **URL (u)**
  This is the node's own URL, stored for better accessibility.
- **Tab ID**
  This is the internal web browser tab ID, in which the page is opened. This is used to detect whether the user switches between different tabs.
- **Title (ti)**
  This is the page's title, as set by the creator of the page. The title is displayed in the search trail visualization next to the node.
- **Highlights (Hi)**
  The highlights are an array containing all user defined highlights on a certain web page. The array itself is sorted by the time of the creation of the highlights.
- **LastHighlightSet**
  This is the UNIX timestamp from when the last highlight was set. This information is used for chronological ordering of the highlight overview.
- **Successors (Su)**
  Successors are the nodes that are connected to a node by an outgoing edge. The nodes are referenced by their ID, which is their URL.
- **Visits (V)**
  These are the timestamps of the visits to this node. A visit-timestamp is created upon opening a URL.
- **Hostname (h)**
  This is the hostname as derived from the node's URL. It is used to generate the clusters in the search trail visualization.
- **Status**
  This is the status of the web page. It is used to determine whether the node's information still needs to be updated. Upon creation, the node has no status indicator. When all information is set after updating the node, the flag 'status: complete' is set.

- **favIconUrl**
  This is the URL of the page's favicon. It was planned to be integrated into the visualization, but turned out to clutter the visualization, especially when pages do not have a favicon or use a standard one.
- **Keywords**
  These are the five most relevant keywords of the corresponding web page. They are derived by the wordstats.js script which into each visited web page and are used to create the keyword table in SearchTrails.
- **MetaJSON**
  The MetaJSON contains all meta information provided by HTML meta tags at the beginning of the page. This information was planned to be integrated into the visualization. Due to a low level of consistency, this information turned out not to be helpful.


```
{ username: "user_5",
  title: "3D-Printing",
  logid: "9sfj1hxqv95abbq6xwmrd1422310649086",
  nodes: {
    http://localmotors.com/3d-printed-car/: {
      url: "http://localmotors.com/3d-printed-car/",
      tabId: 92,
      title: "Local Motors - 3d Printed Car",
      highlights: [
        "The 3D-printed car is made from ABS plastic that has been rein-
forced with carbon fiber. Material for these experiments has been donated
to Local Motors, Oak Ridge National Laboratory, and Cincinnati Incorporated
by SABIC.",
        "Is the entire car 3D printed? Everything on the car that could be
integrated into a single material piece has been printed. This includes the
chassis/frame, exterior body, and some interior features."
      ],
      lastHighlightSet: 1422311680712,
      succ: [
        "http://www.google.de/?gfe_rd=cr&ei=df&gws_rd=ssl",
        "http://en.wikipedia.org/wiki/3D_printing",
        "http://www.inside3dp.com/3d-printing-automotive-industry/"
      ],
      visits: [1422311378695, 1422312717600],
      hostname: "localmotors.com",
      status: "complete",
      favIconUrl: "https://daksb2z.cloudfront.net/545562d/favicon.ico",
      keywords: ["car", "local", "motors", "printed", "revolution"],
      metaJSON: {
        description: "The 3D Printed Car premiers at Detroit. Perhaps the
best example of the power of co-creation and micro-manufacturing, the Local
Motors Strati struts its stuff on the NAIAS floor from January 12th - 25th.
If you are at Detroit, you can watch a Strati being made or win a 3D Print-
ed ride on the track.",
        generator: "Webflow"
      }
    },
    http://en.wikipedia.org/wiki/3D_printing: {...},
  },
  recentChangeLog: [...]
}
```


Figure 53: Excerpt of the search trail data object for a single node with highlights.

Even if the ‘nodes’-object describes the nodes for the search trail and would be able to recreate the force-directed graph, it does not contain all information about the temporality of the user’s actions. The recentChangeLog data object contains information about the sequence in which the search trail has evolved. Figure 54 shows some exemplary events that get captured in the recentChangeLog. The recentChangeLog has been defined in Chapter 2.2.2 as the tuple R = (n, t, e), where n is the node represented by its URL as the identifier, t is the timestamp of the event, and e is the name of the event. As the recentChangeLog is an array of dictionaries, all events are ordered chronologically and contain a UNIX timestamp.


```
{
  username: "user_5",
  title: "3D-Printing",
  logid: "9sfj1hxqv95abbq6xwmrd1422310649086",
  nodes: {...},
  recentChangeLog: [
    {
      time: 1422310653909,
      event: "newNode",
      url: "http://videoag.fsmpi.rwth-aachen.de/vpnonline/13ws-1407.mp4"
    },
    {...},
    {
      time: 1422310660643,
      event: "updateNode",
      url: "http://videoag.fsmpi.rwth-aachen.de/vpnonline/13ws-1407.mp4"
    },
    {...},
    {
      time: 1422311335254,
      event: "createEdge",
      url: "http://www.google.de/webhp?sourceid=chrome-ins&ie=UTF-8",
      fromUrl: "http://www.autonews.com/article/20141027/auto-industry-
uses-3-d-printing-heavily-in-product-development"
    },
  ]
}
```


Figure 54: Excerpt of the recent change log of the search trail data object.

Overall, ten different types of events exist, which are reflected in the recentChangeLog. Table 16 gives an overview of the information stored with each type of event. While all events contain a UNIX timestamp, most events only contain their own name and the URL of the corresponding node. Exceptions are the events that affect the status of the logger, i.e. the pausing and continuing of a search, which carry no additional URL information. Another exception is the ‘createEdge’-event, which contains the start and the end of a directed edge between two nodes with it. These events are stored in a chronological order in the recentChangeLog, and therefore ensure that the search trail can be recreated. The ‘deleteNode’-event makes obvious that the deletion of nodes is used for the modification of the search trail visualization. After a search trail is fully recreated, the recentChangeLog is parsed for the ‘deleteNode’-event, which triggers SearchTrails to remove the respective nodes in a chronological order.

Table 16: Overview of the possible events in SearchTrails and information stored with each event.

Event	Event name	Other information
Create a new node	newNode	The URL of the corresponding node.
Update a node	UpdateNode	The URL of the corresponding node.
Create an Edge	createEdge	url: The URL of the corresponding node. fromUrl: The node from which the edge is created.
Last visit	lastVisit	The URL of the corresponding node.
Set a highlight	highlightSet	The URL of the corresponding node.
Add keywords	addKeywords	The URL of the corresponding node.
Pause SearchTrails	pauseLogger	None
Continue SearchTrails	endPauseLogger	None
Delete a node from the visualization	deleteNode	The URL of the corresponding node.
Mark a page as valueless	minusPressed	The URL of the corresponding node.

The recentChangeLog is especially helpful for extracting the number of interruptions of the search process. Interruptions can be deduced from the number of pause / continue events, but also from large gaps between consecutive timestamps. In case a user turned the attention towards another task, a long break between two single events in the log would appear, which can be detected and provide a hint towards the idle times of the user. Another advantage of the recentChangeLog is that it contains changes that were made in the user interface. When users decide to delete a node from the visualization, this event gets captured in the log. At the recreation of the search trail, the nodes are deleted after the full recreation of the graph. This approach conserves the original search process, and has the technical advantage that the information about successors of nodes does not need to be changed when a node is deleted.

This section clarifies technical details of the data behind SearchTrails. The next section explains how the search trail data object is stored on the remote storage server.

4.2.4 Remote search trails storage

One important feature of SearchTrails is the remote storage of the search trail data object in regular intervals on a remote log storage server. Due to technical limitations, a dedicated mechanism had to be implemented to enable this storage. For easy access to the stored logs, BSCW was chosen to serve as the container to which the logs were sent. Figure 55 shows the logs as they are represented in a BSCW folder.

For storing the logs in BSCW, a way of contacting the BSCW API from the SearchTrails browser extension had to be found. As SearchTrails technically lives in the browser, the search trail data object is only present in the user's web browser instance. It is not possible to contact the BSCW API from a browser instance due to Cross-Origin Resource Sharing (CORS) restrictions. The BSCW API does not allow requests from other domains for security reasons. Therefore, the BSCW API could only be contacted from a remote server, but not from a client browser. To achieve this, I created a search trail data storage service based on Node.js[76], which receives the search trail data object from the browser instance and takes care of storing it on BSCW via the BSCW API, making use of a secure HTTPS connection.

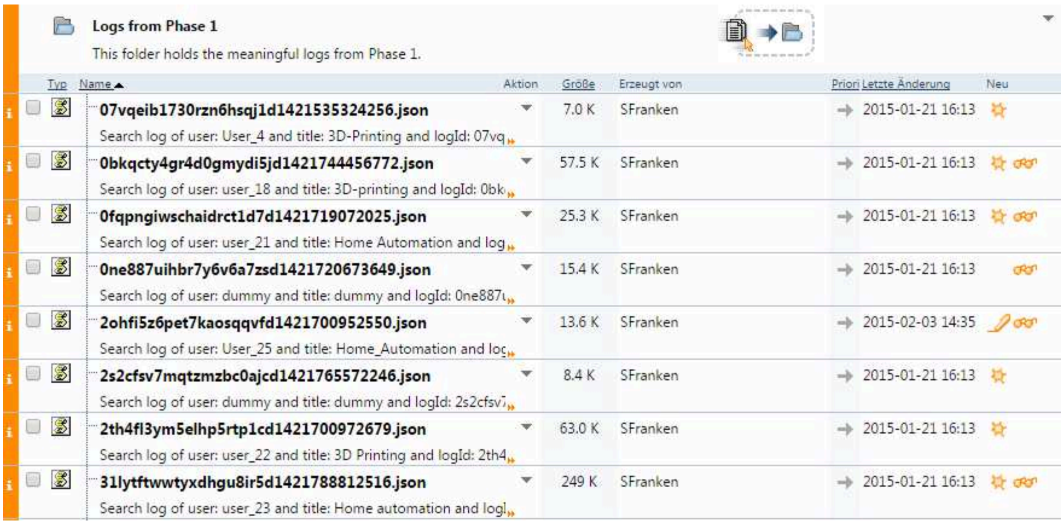


Figure 55: Screenshot of the search trail storage folder in BSCW.

The SearchTrails Node.js server is hosted at the Heroku PaaS provider and manages storing the search trail data objects to BSCW via the BSCW API. It accepts requests from the Search-Trails instance in the browser and receives the search trail data object. The server then opens a secure connection to the BSCW server, contacts the BSCW API, and initiates storing the search trail on BSCW, where a search trail file is created or overwritten if it already exists. Figure 56 depicts SearchTrails' storage process.

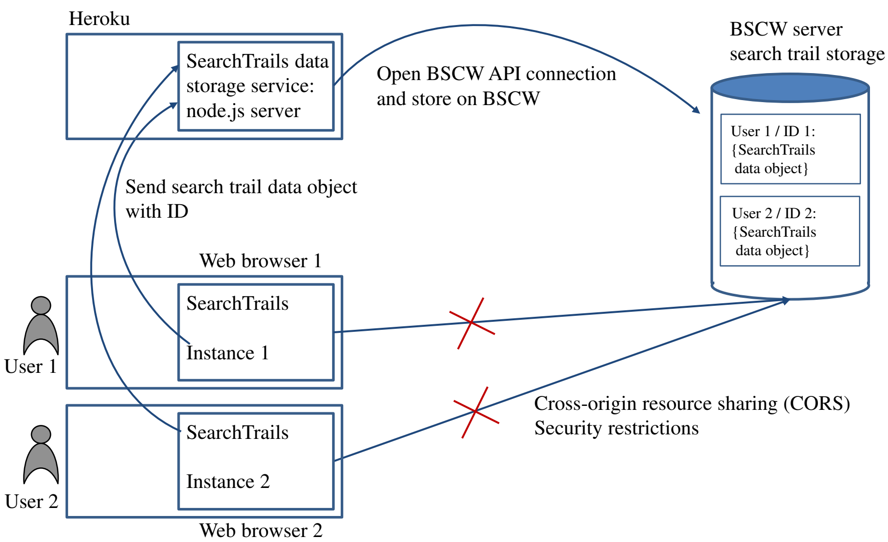


Figure 56: Conceptual view of the search trail storage process.

Similarly, search trail data objects can be retrieved from BSCW (Figure 57). After sending the search trail ID to the Node.js server, the server opens the API connection to BSCW and

[76] Retrieved August 27, 2015 from https://nodejs.org/

fetches the metadata of all content of the general search trail data object storage folder. Within the metadata, the server application searches for the search trail with the matching search trail ID and stores the corresponding BSCW object ID. Then the Node.js server application requests the file with this BSCW ID from the BSCW server via the BSCW API, and the file is delivered by BSCW. During this process, SearchTrails does a limited number of requests to the Node.js server to poll for the result. When the BSCW finally delivers the search trail data object, this is delivered to the SearchTrails instance as the response for one of the polling requests and the search trail is recreated in SearchTrails.

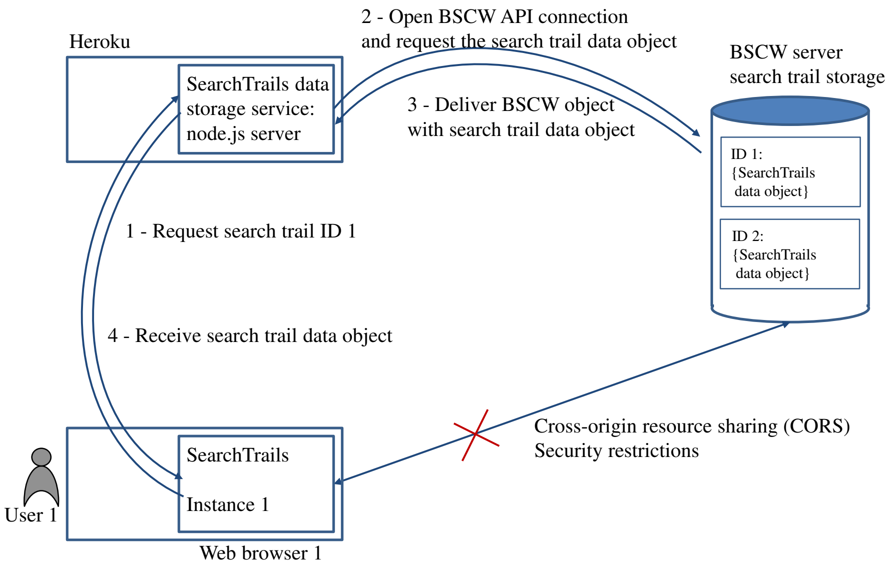


Figure 57: Conceptual view of the retrieval of a search trail data object.

This subchapter explains important details of the implementation of SearchTrails. It first explains why I choose Google Chrome as a platform for implementing SearchTrails and which external frameworks and libraries are used to achieve the final feature set of SearchTrails. I then explain the generated data and how the search trail data objects are stored on the BSCW server.

4.3 Design iterations

Chapter 4.1 and Chapter 4.2 help introducing the architecture of SearchTrails and the implementation details. In this subchapter, I describe the design iterations which lead to the final version of SearchTrails. I start with a set of paper prototypes which help to develop the general idea of SearchTrails. I then introduce an early prototype of SearchTrails. This prototype called ‘SemanticClipboard’ was developed before the work on SearchTrails started. SemanticClipboard helped developing first ideas for SearchTrails, but was based on a different platform with a different data collection approach.

In the last two sections of this subchapter I introduce the two versions of SearchTrails, which I implemented in two development cycles. Here I explain the set of technical features for both versions of SearchTrails. I evaluate each of the two versions of SearchTrails in a user study and describe both user studies and their results in Chapter 5.

4.3.1 Paper prototypes

The work on SearchTrails starts with early paper prototypes, in which I develop the idea of transforming the user's navigation in the Internet into a search trail. Several of these paper prototypes provide an insight into the early development phases of SearchTrails. One main idea in this process is breaking up the linearity of the chronological browser history. SearchTrails should provide a more-dimensional look onto the navigation process, which can be considered a more associative, thus more natural view on the user's own browsing history.

The first section presents a set of sketches of search graphs and valuable information, which inspired the development of SearchTrails. Their key idea was imitating the behavior of SearchTrails during search processes by hand. The second section presents sketches that depict a more technical view on the creation of search trails, by evaluating which information is needed to build the search trail.

4.3.1.1 Search graphs

In this section, I present paper prototypes that convey the overall idea of SearchTrails. These paper prototypes visualize the envisioned behavior on a high level. They illustrate the overall idea of constructing a network of information generated during a complex search process with the valuable pieces of information being highlighted.

Figure 58 shows an example information network on the topic of a certain type of artistic glass, so-called 'Myra glass'. The network was constructed by simulating the envisioned behavior of SearchTrails by writing down the navigation during a search session on the given topic. Even if the sketch in Figure 58 is not readable in every detail, it shows the main question of the search trail as the central starting point. In opposition to the final SearchTrails implementation, the drawn network features the central information need as the starting node and not an actual web page. Another difference to SearchTrails is that new web searches are connected to the central node instead of being inserted into their logical position in the search trail. A number of enumerated edges to different resources start at the central node. The first edge leads to the results of a Google query, from which several results are visited. While some results lead to further interesting pages (e.g. edge 2 leads to another resource via edge 3), other result pages do not lead to interesting resources. Sometimes the found information triggers a new search, which is attached to the central point as a new sub-trail (e.g. the biography in node 3 triggers the search 4). This way, the network emerges during the search, and the single search results influence each other. This paper prototype especially illustrates that a search process is not a linear cognitive process, but that the new information influences the searchers and their actions. Highlighted in red are the central question and all pages where valuable information have been found.

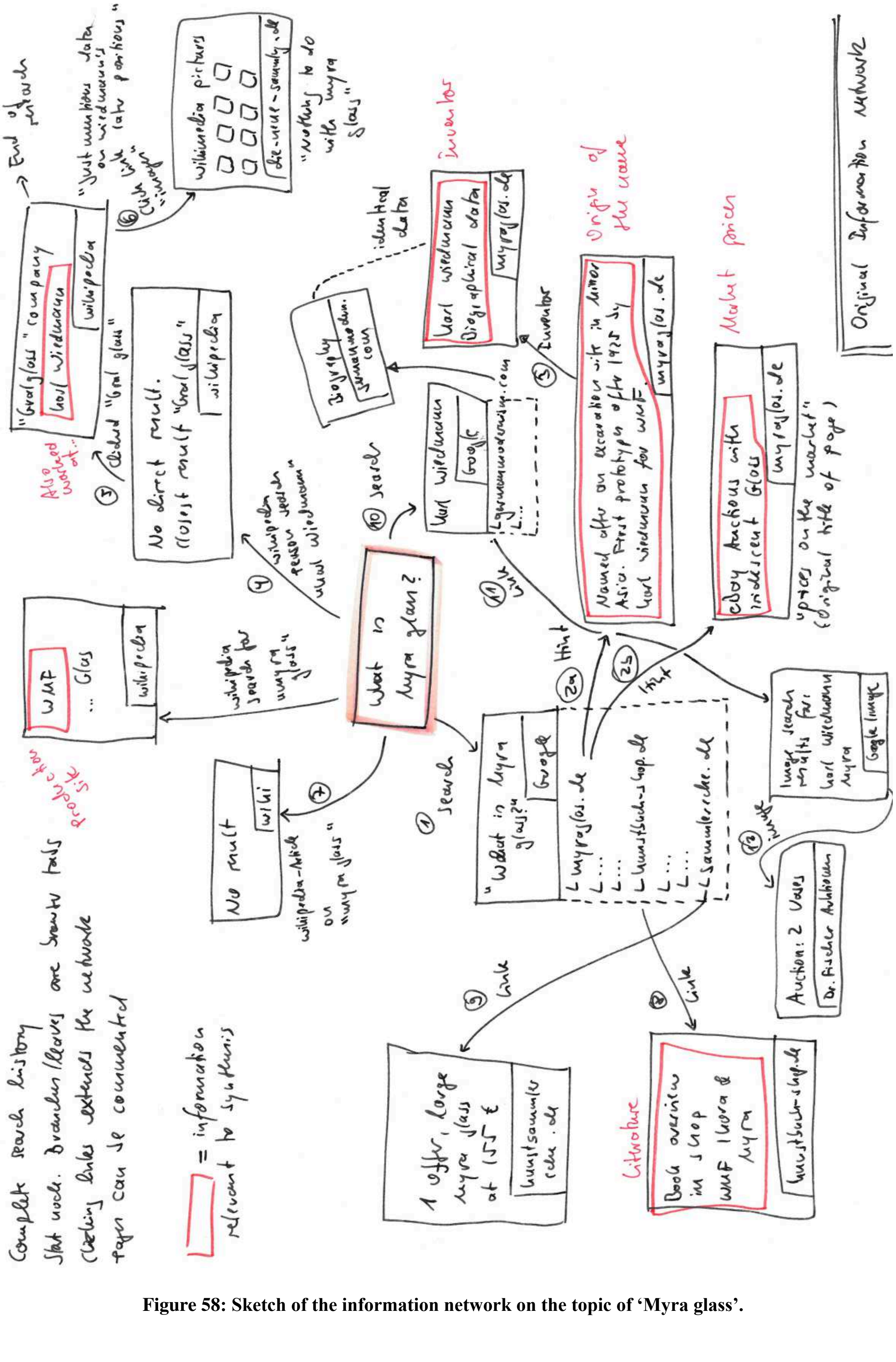


Figure 58: Sketch of the information network on the topic of 'Myra glass'.

Figure 59 shows a second paper prototype on the same search topic. It illustrates how the valuable pieces of information from the information network may be condensed into a network of valuable resources. The network only contains the valuable pieces of information, which are found on various pages, the so-called ‘highlights’. The network now only shows the resources containing valuable information and their relation among each other. This idea has not been integrated into the final implementation of SearchTrails. Instead, I realized that valuable resources are highlighted within the search trail and show the respective highlights in a pop-up upon hovering over the search trail’s nodes. This way, the highlights are always shown within their original context.

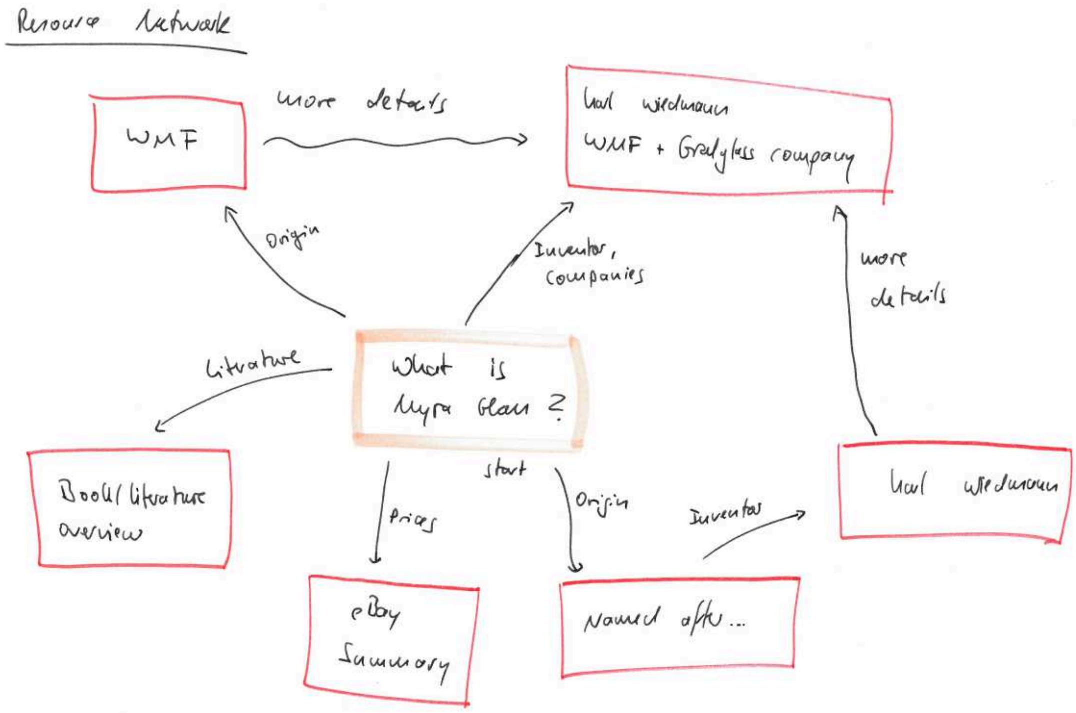


Figure 59: Sketch of the valuable resources on ‘Myra glass’ from the information network.

Finally, the information network with its highlights can be condensed into a fact sheet, which contains the found information in a linear fashion. The sketch in Figure 60 shows this fact sheet on the selected topic. It presents the important information found during the search process, and highlights the facts from the information network by red frames. The non-highlighted information was added manually, without a reference to a network node. Partially, because this information was too general to be related to just one node; but the information would still be helpful for a person who is unfamiliar with this topic. All highlighted pieces of information also contain the reference number of the element in the network, such that a tight connection between the fact sheet and the information network can be ensured. The fact sheet view in this paper prototype is very similar to the described highlight overview of SearchTrails, as it condenses and linearizes the information found.

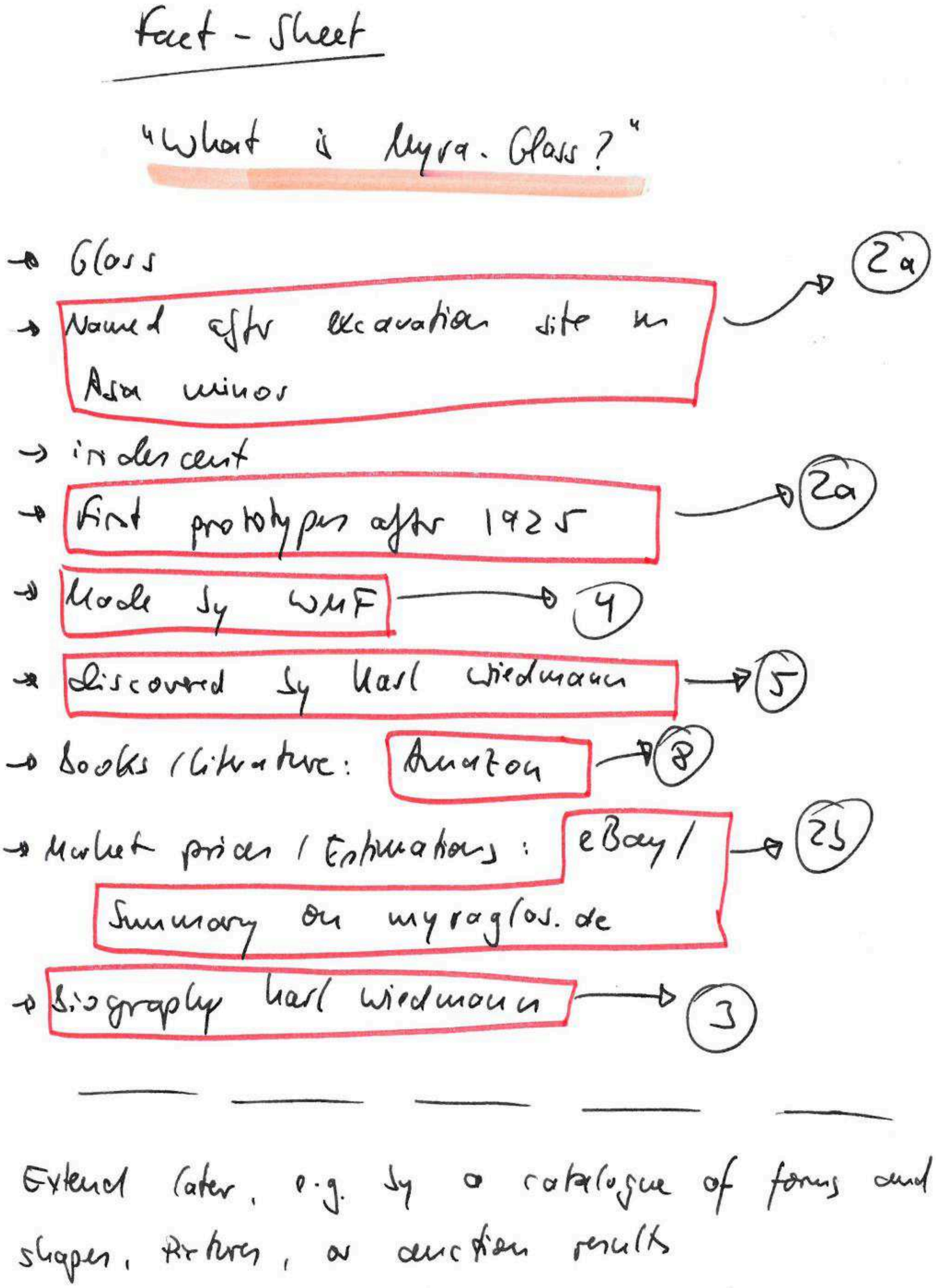


Figure 60: Fact sheet on the topic 'Myra glass', derived from the information network.

This section presents early approaches towards the realization of SearchTrails. These approaches are both technical and logical in nature. A first set of paper prototypes describes technical thoughts for essential features of SearchTrails, such as the creation of the information network itself. The second set of paper prototypes provides an insight into the basic ideas that influence the development of SearchTrails.

4.3.1.2 Creating search trails

The sketches in this section build upon the previous, high level sketches. They provide a glance on the technical thoughts behind the development of SearchTrails, the idea of using a force-directed graph for visualization, and the information necessary to build the search trail.

Figure 61 shows an early drawing of a search trail, indicating several key features of SearchTrails. These features include a flexible layout, directed edges between the nodes, a non-linear approach of browsing, and highlighted nodes. The highlighted nodes indicate web pages with valuable information.

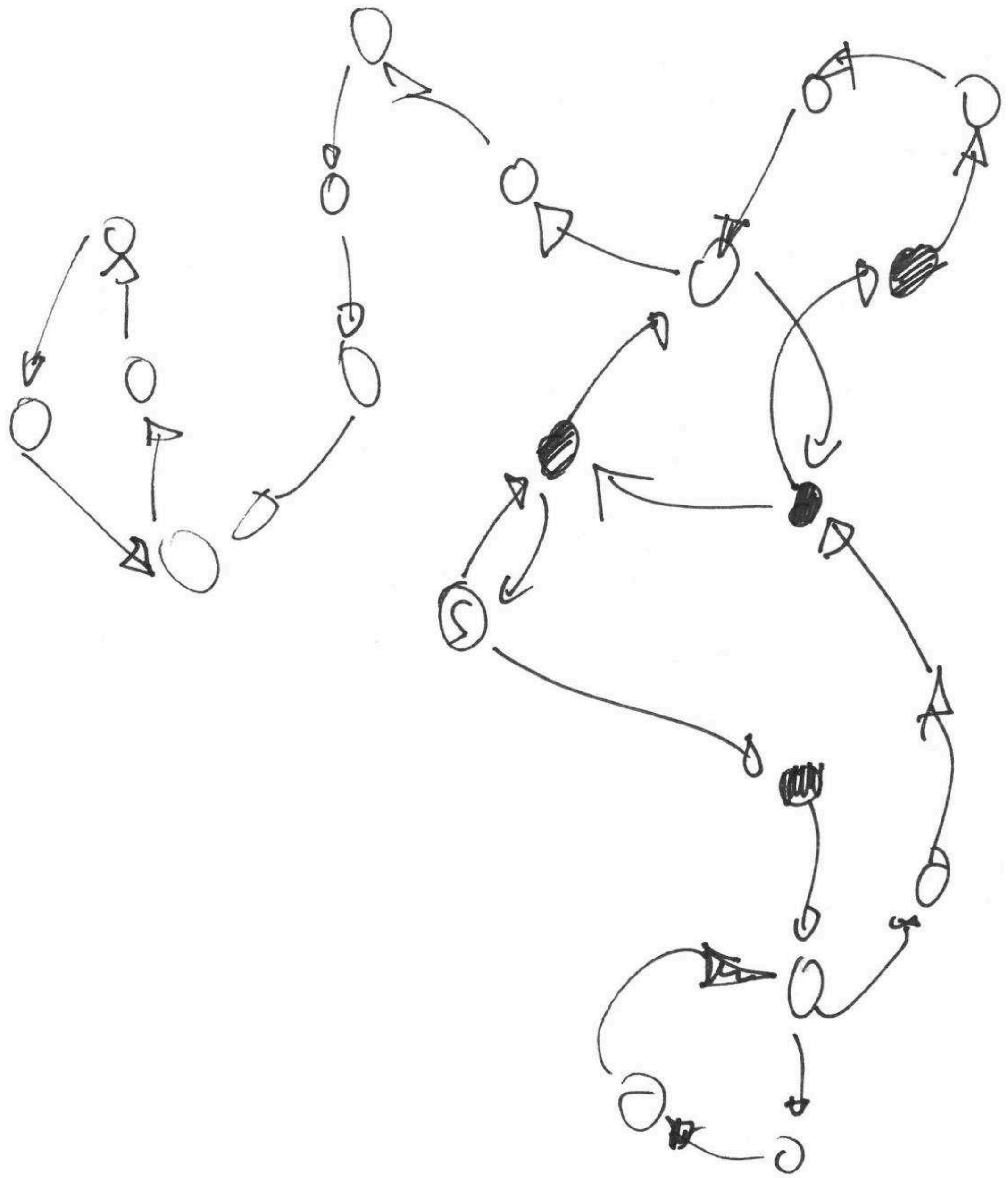


Figure 61: Sketch of the search trail as a force-directed graph.

In order to build such a search trail, certain information is needed. First, it needs to be clarified which information is necessary for building the search trail. This includes information like the time of the visit, the predecessor node, the number of visits, the ID of the search trail and the node, the URL itself, or the status of the tab. Accordingly, it needs to be clarified where the information originates and how it is possible to access it. Figure 62 shows an early sketch of the information required to realize the envisioned approach and the possible sources of this information. On the left side, the sketch names the desired information and on the right side the sketch names the sources of this information. The chronology of the user's interaction and the predecessor web page can both be retrieved from evaluating the browser status. The number of visits to a web page within a session needs to be counted by SearchTrails, while the ID of a search trail is created by the extension itself. The URL of a web page and the status of the respective browser tab may again be detected by evaluating the browser status.

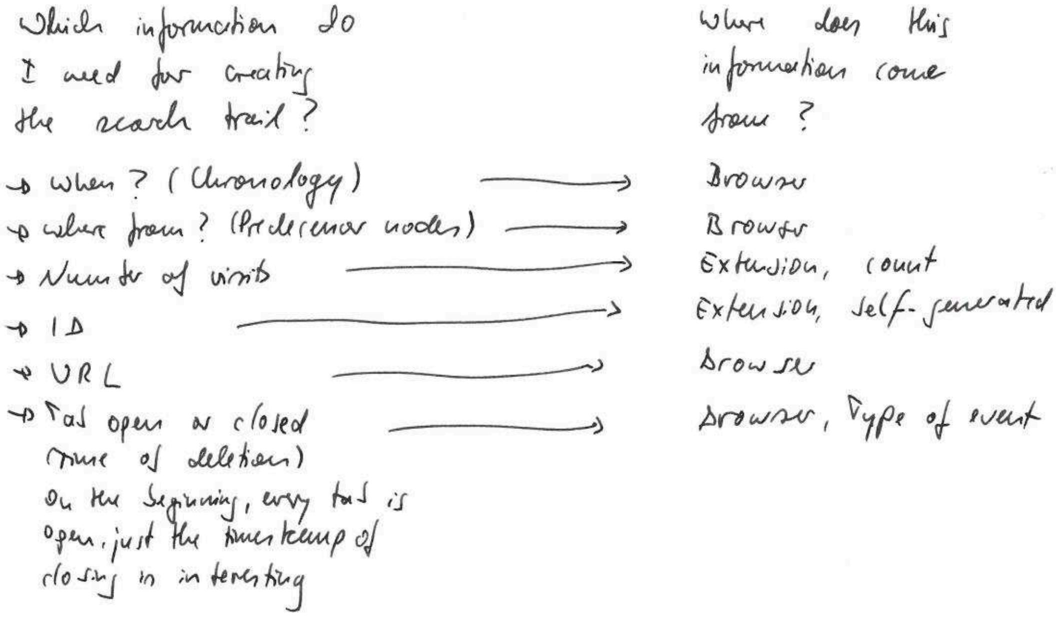


Figure 62: Sketch of the information sources for relevant node information.

Similarly, each node in the search trail needs a set of information that provides value for the user and that includes technical information for building the search trail. Figure 63 shows an early sketch for the minimal information set of a node. This information contains the ID of the node, the timestamps of creation and visits of the node, the URL, its highlights, and the title of the node.

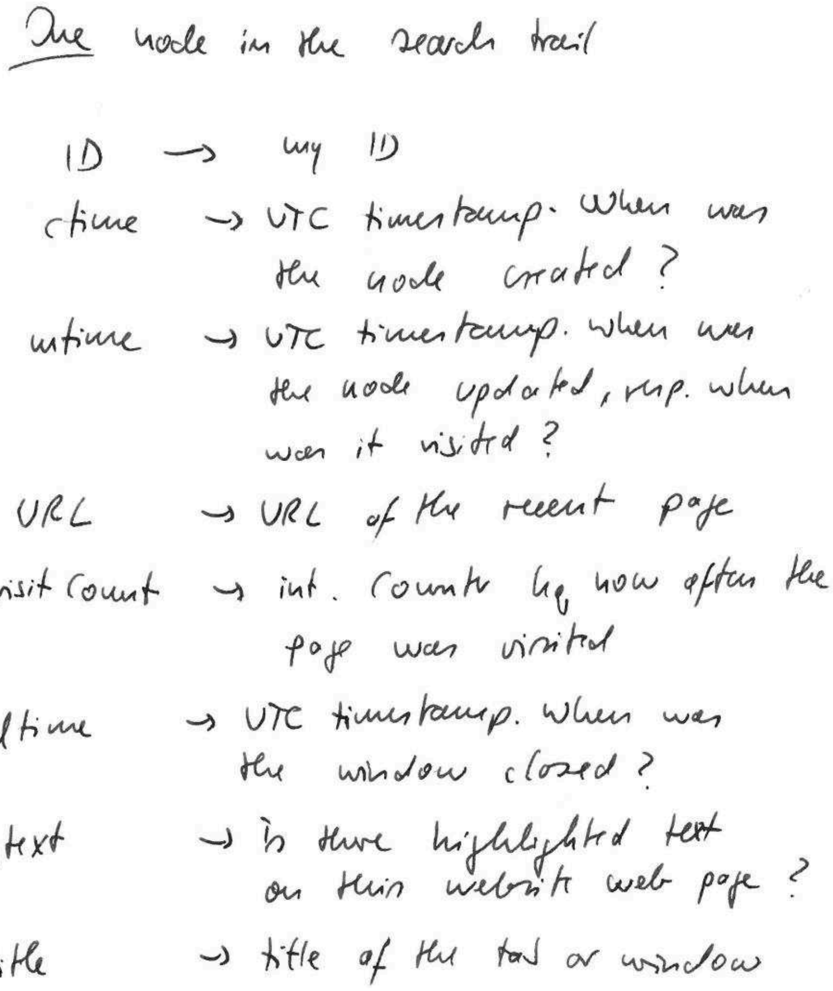


Figure 63: Sketch of necessary information for a single node.

All this information can be collected from the various information sources to a web browser extension. It then needs to be clarified how this information is translated into the actual search trail. Figure 64 shows a sketch which relates the user's navigation behavior to the development of a search trail. Even if the sketch is not readable in every detail, it describes some core principles for SearchTrails. The left column shows the content of the web browser, while the second column describes the user's actions. The third column shows the search trail, while the last column shows the information about the recent node (R), the last visited node (L), and the edge created in this step of SearchTrails in a small table. The sequence of action describes how the user starts from an empty search page (node S), visits three new pages after another (nodes 1, 2, 3). From the third visited page, the user navigates back to the first visited page by pressing the 'Back' button twice. The user enters a new URL and a new node gets created (node 4). After navigating to another web page from the new node (create node 5), the user closes the tab with the web page corresponding to node 4 still left open, which causes no change in the search trail.

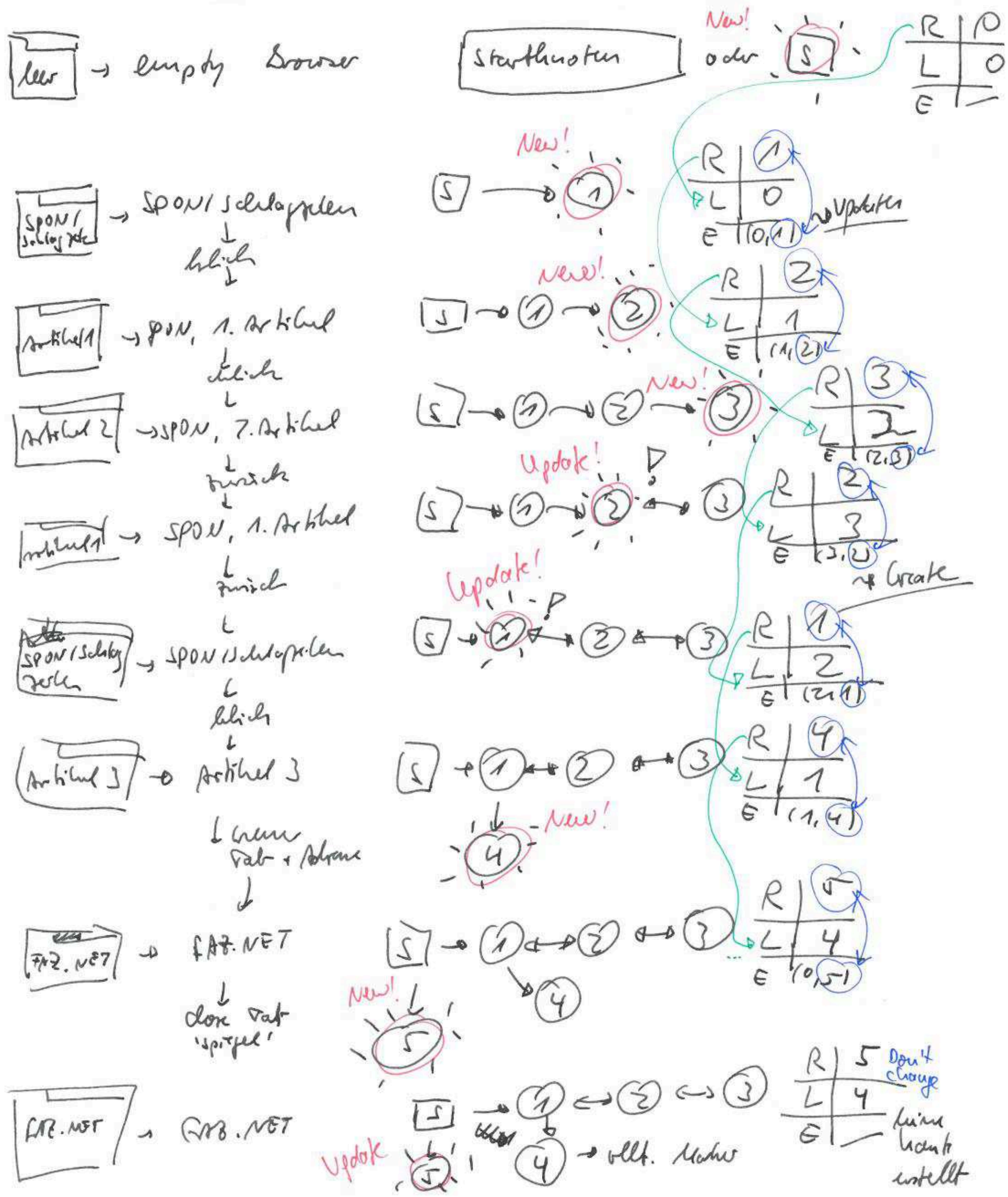


Figure 64: Detailed sketch, depicting the transformation of user navigation into a search trail.

4.3.2 SemanticClipboard: A first prototype

Based on the ideas on search networks from an early stage of my thesis, I implemented a prototype called ‘SemanticClipboard’. This prototype intends to capture important information found during search processes, and allows the user to put the pieces of information into relation.

SemanticClipboard offers a desktop, on which information elements can be collected. The users can give a name to the desktop, describing what the information collection is about. New pieces of information can be created by pasting the most relevant information into a specific template. This template offers the possibility to include five types of information. First, the title of the information and second the valuable information itself can be collected. Third, the origin of the information can be entered, e.g. the URL of a web resource, or the location of a file, but also the date and title of a newspaper article. Fourth and fifth, the author of the information can be entered by the user as well as the creation date of the piece of information (Figure 65).

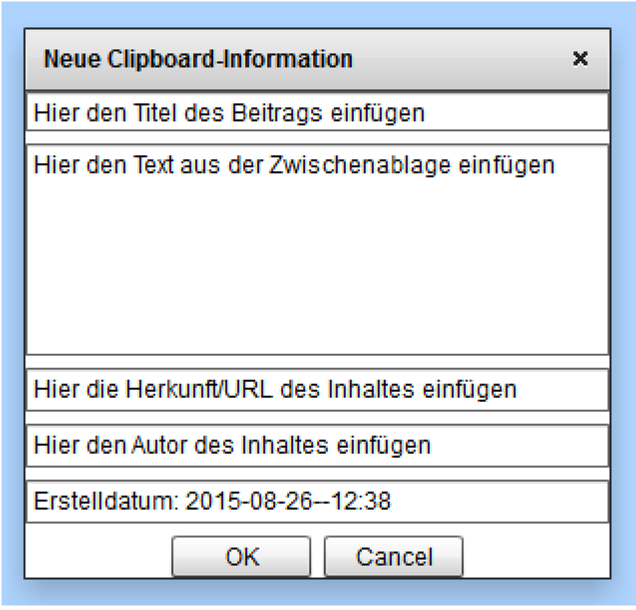


Figure 65: Creating a clipboard element in SemanticClipboard.

After filling out a template the piece of information is created on the desktop. Additionally, all pieces of information offer an extra field for adding comments. The view of each piece of information can be switched between the content view, displaying the actual content of the piece of information together with the comments, and the metadata view. The metadata view displays the author of the information, the location (URL or other), and the creation date. The different pieces of information can be rearranged by the user on the SemanticClipboard desktop and they can be minimized or deleted. This allows the users to arrange the desktop content logically and to structure all content based on subtopics of the larger topic.

Figure 66 shows a SemanticClipboard desktop, containing information on organizing a conference attendance. The different clusters of pieces of information show information about the conference itself, its location, hotels in the city of Stuttgart, open tasks related to the conference, and some information on sights and culture. This structure can be recreated by everyone having access to the underlying XML data structure, and can be individually rearranged to fit the searcher's needs.

SemanticClipboard is realized as an Adobe AIR application[77]. This allows SemanticClipboard to access the local hard drive, and to store and retrieve files from it. The content of a search session can be stored on the local hard drive as an XML file. SemanticClipboard allows recreating XML files from other SemanticClipboard instances on its desktop, such that a search session can be resumed after interruption. Additionally, SemanticClipboard offers access to a BSCW server instance from which blog articles can be integrated into the pieces of information.

Overall, SemanticClipboard served as a first prototype for extracting, arranging, and storing valuable information persistently. It helped to develop and clarify first ideas for the thesis and to identify open questions with respect to capturing and exchanging valuable information. SearchTrails did not yet allow the automatic or manual connection of pieces of information. As it turned out that the tool developed in my thesis should reflect the user's interaction with the Internet, the work on SemanticClipboard was not continued. This was due to its missing integration with local web browsers, which turned out to be technically problematic.

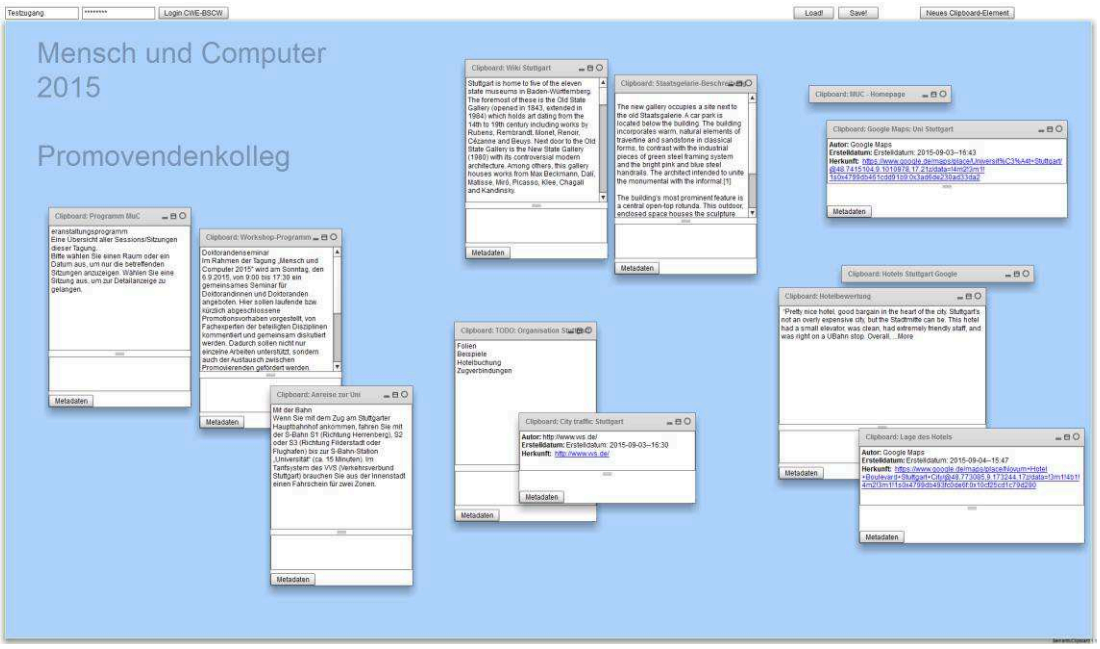


Figure 66: Example desk of SemanticClipboard for organizing a trip to a conference.

At an early point of the thesis, it became apparent that SemanticClipboard served as a generator of ideas on the envisioned possibilities of SearchTrails, but could not be the environment to develop SearchTrails in. Instead, the programming approach of SearchTrails was switched from a standalone desktop application to a web browser extension, which I describe in the next section.

[77] Retrieved August 26, 2015 from http://www.adobe.com/products/air.html

4.3.3 First evaluated version of SearchTrails

In this section, I describe the first version of SearchTrails. This version was evaluated during the first user study in February and March 2014. The version of SearchTrails that was used for this user study was named ‘SearchLogger’. This version offered a limited set of functions compared to the second evaluated version, as it was the outcome of the first iteration of the iterative development process.

Figure 67 shows the first evaluated version of SearchTrails in a screenshot. In this version, SearchTrails consists of a single page, showing all interaction possibilities. The main part of the page is reserved for the search trail visualization, while the right column shows the keyword table. The toolbar on the bottom of the page offers some interaction possibilities to the users.

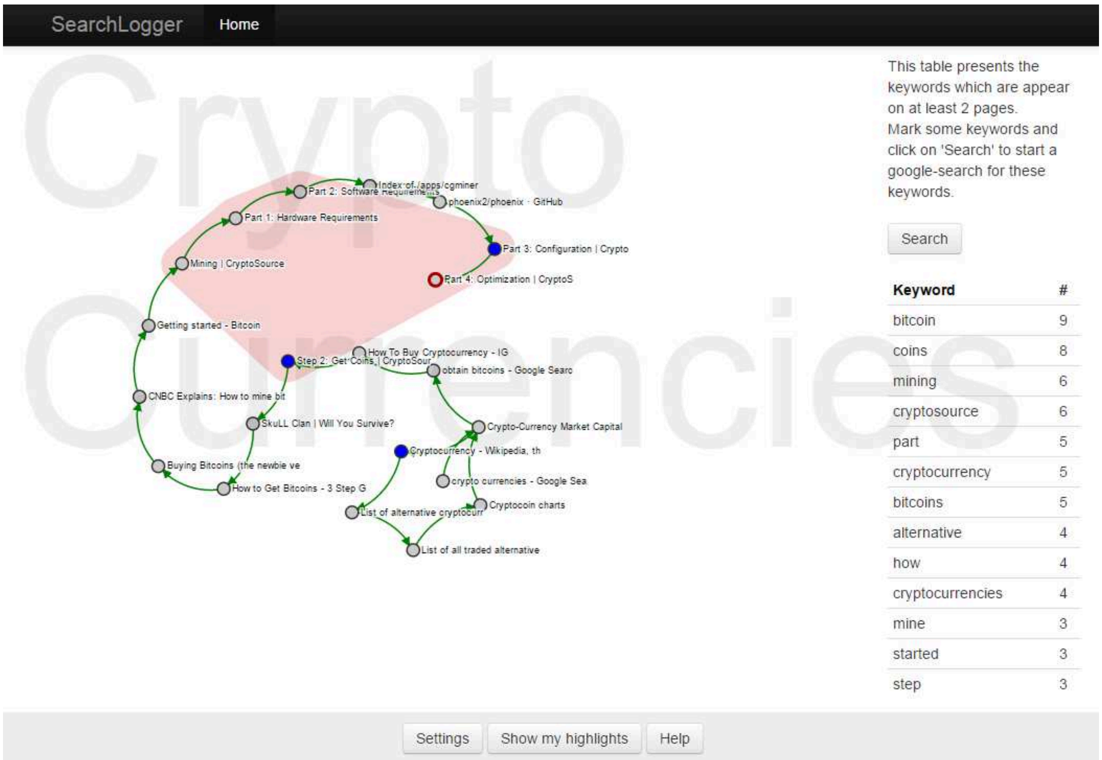


Figure 67: Screenshot of the home screen of the first evaluated version of SearchTrails.

The search trail visualization of the first evaluated version of SearchTrails is already very similar to the second version, offering the central features developed in Chapter 3. These features are the following:

- **Force-directed graph visualization**
  The search trail is visualized as a force-directed graph, which is rendered by SearchTrails’ visualization engine. Due to its composition of individually rendered SVG elements; the search trail can be manipulated by the users. The search trail allows being dragged and rearranged within the borders of the force-directed layout. As such, it is possible to turn the graph, and to disentangle nodes, but it is not possible to attach nodes to each other or to fix the layout.

- **Textual descriptions of the nodes**
  Each node is described by the title of its corresponding web page, as given by the web page creator. In order to avoid excessively long titles, the titles are abridged after 25 characters. This is sufficient for the most essential information, as the naming convention of web pages seems to be the individual name of the page, followed by the host of the page. This way, the title can be abridged without losing too much information. Examples for this are 'Alternative currency - Wikipedia, the free encyclopedia' or 'Bitcoin Estimated Transaction Volume (BCHAIN) - Data and Charts from Quandl'.

- **Pop-ups on the nodes**
  When hovering over a node, a pop-up is displayed, showing the unabridged title of the web page as a link to the corresponding web page, as well as the keywords of the web page. If the user has attached highlights to a node, these are also displayed in the pop-up box.

- **Clustering nodes**
  The search trail visualization features the clustering of nodes. For this purpose, convex, colored hulls are created around all nodes from the same host. As one node can only belong to one host, clusters cannot intersect. Double-clicking into the cluster hull triggers the contraction of the cluster. The set of all nodes within the cluster is replaced by one large cluster node, and only the in- and outgoing edges to and from the cluster remain. Even when the cluster is collapsed to a single node, the underlying search trail stays fully responsive to the user's actions and creates new nodes within the cluster if necessary.

- **Highlights**
  Whenever the user sets a highlight by selecting text on a page and presses the 'Control'-key, the selected text is added to the highlights of the corresponding node and appears in the table of highlights. The node itself gets colored blue and the node's pop-up shows the node's highlights. By coloring the nodes, the valuable web pages are easily identifiable for the user and makes visible where highlights have been found.

- **Marking the recent node**
  In order to provide some orientation within the force-directed graph, the node corresponding to the lastly visited web page is highlighted by a red circle around it.

The keyword table is also very similar to the second version of SearchTrails. After extracting the five most relevant keywords from each visited web page, the keywords get attached to the node data in the search trail data object. Upon every change of the search trail data object, also the keyword table is updated with the new keywords. The keyword table contains all keywords that appear on at least three different pages. It is ordered by the frequency of the keywords, showing the most frequently found keywords on top. Clicking on a keyword highlights the according keyword table row, and marks all nodes in the search trail which contain the selected keyword. Selecting multiple keywords in the keyword table is possible and results in marking all nodes in the search trail which contain at least one of the selected keywords. The search button on top of the keyword table offers the possibility to search for the selected keywords by opening a new tab with a Google search on all selected keywords.

The toolbar on the bottom of the page offers some interaction possibilities to the users, i.e. changing settings of the graph, accessing the underlying search trail data object, visualizing the highlights, and accessing a help page. All information is shown as visual overlays over the search trail visualization.

- **Settings**
  Clicking the settings button results in displaying two input fields, where the username and the title of the search can be entered. Both pieces of information are stored in the search trail data object, and the title is shown in the background of the search trail visualization, similar to the SemanticClipboard prototype.
  Additionally, a button is offered which toggles showing the search trail data object to the user. This button was used during the study to access the JSON search trail data object when I asked the users to send me the JSON object via e-mail. Figure 68 shows the settings of the first version of SearchTrails and an excerpt of the JSON visualization.

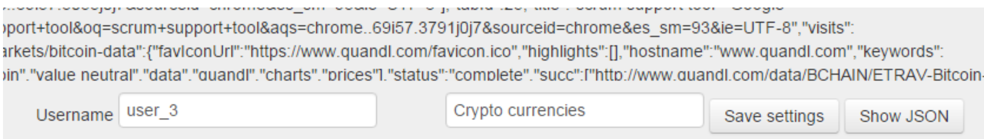


Figure 68: Possible settings in the first version of SearchTrails.

- **Showing the highlights**
  This button toggles the display of the table of highlights as an overlay to the search trail visualization. Figure 69 shows an example highlight overview. Every web page where highlights are taken from is displayed in its own table row. All highlights from the same page are displayed as a list. Each highlight may be deleted individually by the user. The table furthermore displays the host where the web page stems from, and features a link to the source web page where the highlight stems from. In the example, two web pages from the Wikipedia contribute highlights to the overview, with one of the two pages contributing two highlights to the overview.

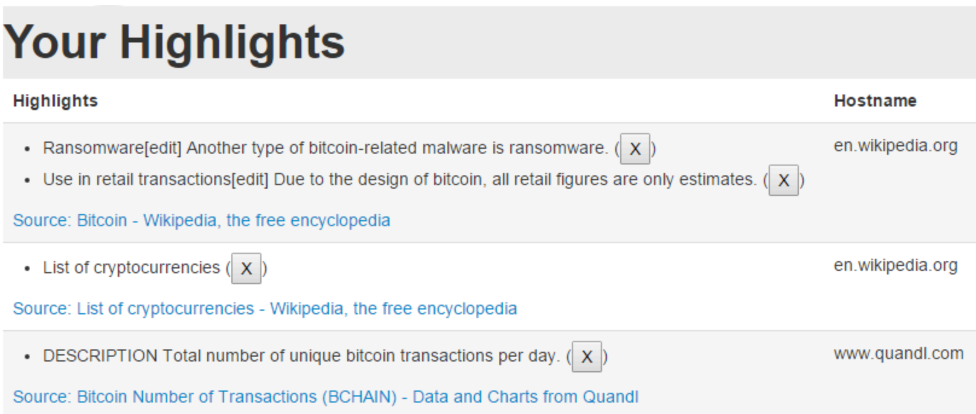
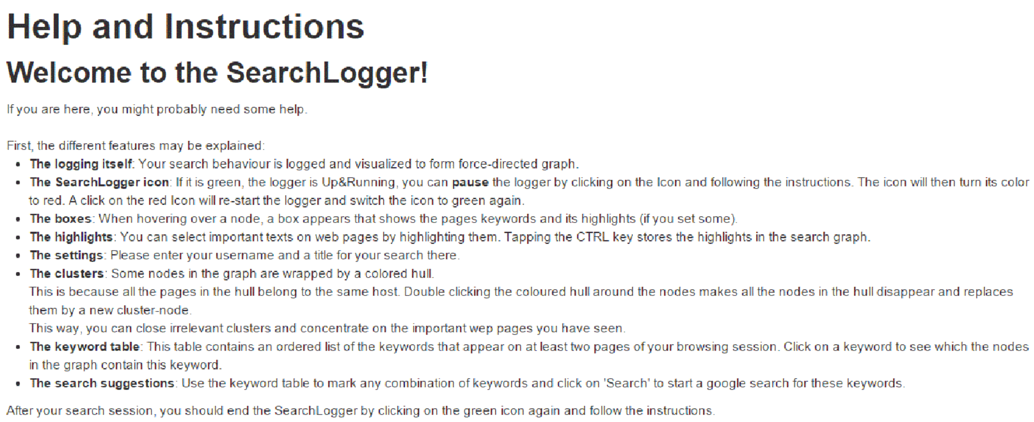


Figure 69: Highlight overview in the first version of SearchTrails.

- **Help**
  The help button toggles the display of a small help page, containing the most important information on the states of SearchTrails, the search trail visualization, the keywords, the clusters, the interaction options, and the settings (Figure 70).


Help and Instructions
Welcome to the SearchLogger!

If you are here, you might probably need some help.

First, the different features may be explained:

- **The logging itself**: Your search behaviour is logged and visualized to form force-directed graph.
- **The SearchLogger icon**: If it is green, the logger is Up&Running, you can **pause** the logger by clicking on the Icon and following the instructions. The icon will then turn its color to red. A click on the red Icon will re-start the logger and switch the icon to green again.
- **The boxes**: When hovering over a node, a box appears that shows the pages keywords and its highlights (if you set some).
- **The highlights**: You can select important texts on web pages by highlighting them. Tapping the CTRL key stores the highlights in the search graph.
- **The settings**: Please enter your username and a title for your search there.
- **The clusters**: Some nodes in the graph are wrapped by a colored hull.
  This is because all the pages in the hull belong to the same host. Double clicking the coloured hull around the nodes makes all the nodes in the hull disappear and replaces them by a new cluster-node.
  This way, you can close irrelevant clusters and concentrate on the important wep pages you have seen.
- **The keyword table**: This table contains an ordered list of the keywords that appear on at least two pages of your browsing session. Click on a keyword to see which the nodes in the graph contain this keyword.
- **The search suggestions**: Use the keyword table to mark any combination of keywords and click on 'Search' to start a google search for these keywords.

After your search session, you should end the SearchLogger by clicking on the green icon again and follow the instructions.

Figure 70: Help screen in the first version of SearchTrails.

This first version of SearchTrails was still named SearchLogger, and was used during the first qualitative evaluation of SearchTrails. I describe the results of this evaluation in Chapter 5.1. The feedback collected during this evaluation was used to substantially improve Search-Trails for the second version.

4.3.4 Second evaluated version of SearchTrails

After the first evaluation, I evaluated the feedback gathered to improve SearchTrails and develop new features. Even if the basic functionality stayed the same, I implemented several improvements with respect to the user experience. Most important, I removed the settings bar at the bottom of the page and split up the SearchTrails user interface into six distinct pages: ‘Home’, ‘Your highlights’, ‘Show JSON’, ‘Continue a search trail’, ‘Settings’, and ‘Help’. Figure 71 shows the home screen of SearchTrails with a search trail from the second evaluation on the topic ‘3D-printing’.

The following sections each describe one of the six distinct pages of SearchTrails.

Home screen

The 'Home' screen is the main visualization of SearchTrails, and it is comparable to the earlier version of SearchTrails. Some improvements have been developed compared to the first version:

- The layout has been changed slightly: The title of the search is no more displayed in the background of the visualization. It is now displayed on top of the keyword table in the right column. It is visible together with the user name and the request to not close the tab. This request is necessary for the user study, as in this case the system process SearchTrails runs in would be terminated and information could be lost.
- It is now possible to mark and delete nodes. Clicking a node marks the node by high-lighting it orange. Only one node can be marked at a time. When a node does not have multiple in- and outgoing edges, it may be deleted from the search trail by pressing the 'Delete' key. This change only affects the search trail, but not the underlying search trail data object.
- Irrelevant nodes can be marked by pressing the 'Minus' key on the corresponding web page. This marks the corresponding node in the search trail in the color pink (cf. Figure 71). This way, the user can easily detect nodes with valueless information.
- The stop word list has been extended by several web search related entries, like 'www', 'google', or 'encyclopedia'. These words turned out to be not relevant in the keyword table.

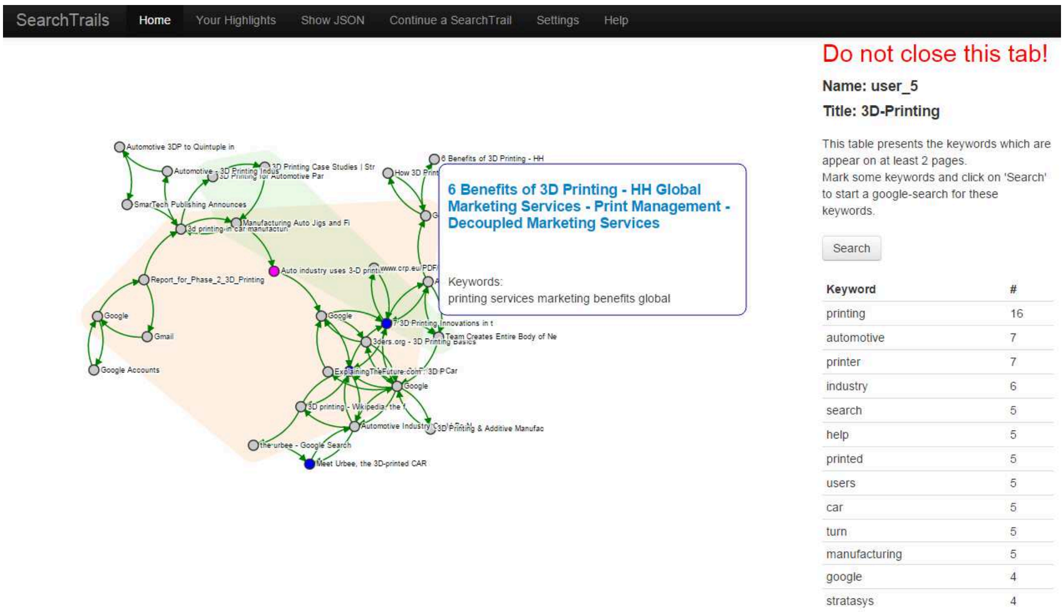


Figure 71: Home screen of the final version of SearchTrails.

Highlights overview

The highlights overview is contained in a new page now (Figure 72). I implemented several changes with respect to functionality and user experience. First, it is now possible to add manual highlights per web page. It is therefore possible to manually summarize or add to the highlights found during the search process and to contribute to the given overview. Furthermore, highlights are now ordered by their time of creation. All highlights of the web page where the most recent highlight has been created are shown on top of the list. This way, the most recent information can easily be accessed, and the highlight list is ordered from late to early findings during the search process.

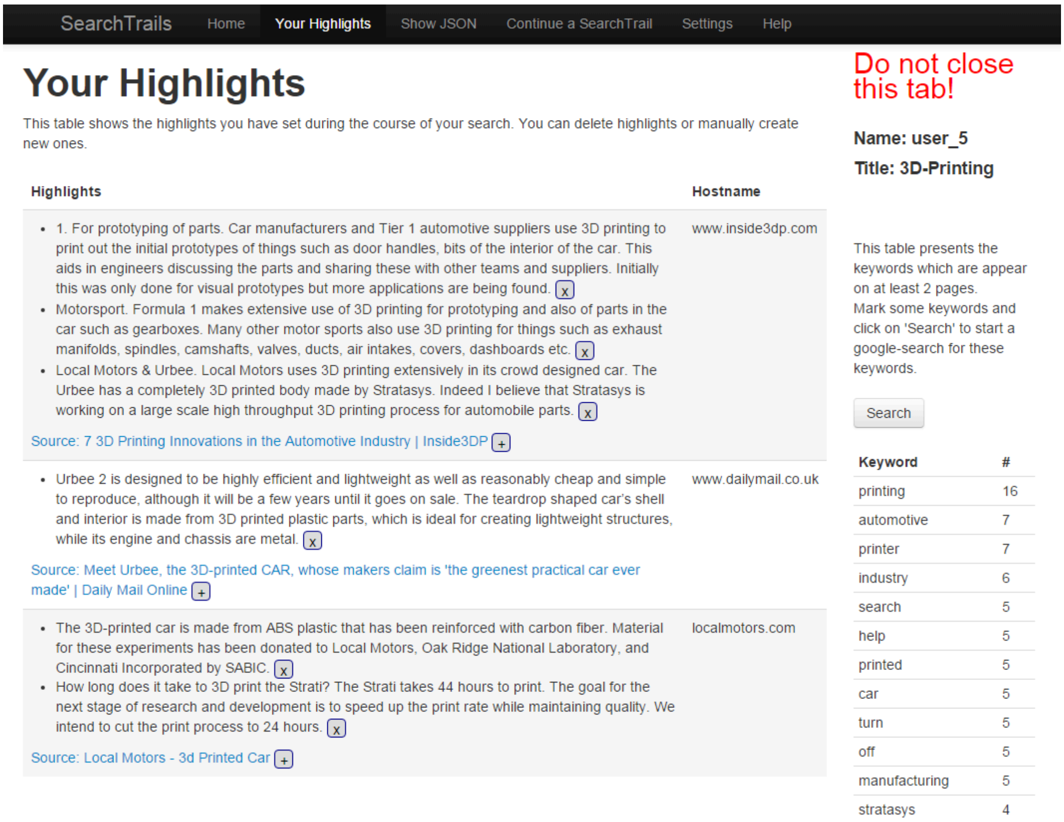


Figure 72: Highlight overview in the final version of SearchTrails.

Show JSON

For technical purposes, the JSON search trail data object can be accessed via the 'Show JSON' page (cf. Figure 73). This page displays the JSON as a plain text, but also contains a button that automatically downloads the search trail data object as a .json-file to the browser's download folder. The file name is created by appending the string 'SearchTrail' with the date and the time of the creation of the file. This file stores the search trail on the user's hard disk. During the second user study, this page was hidden from the participants.

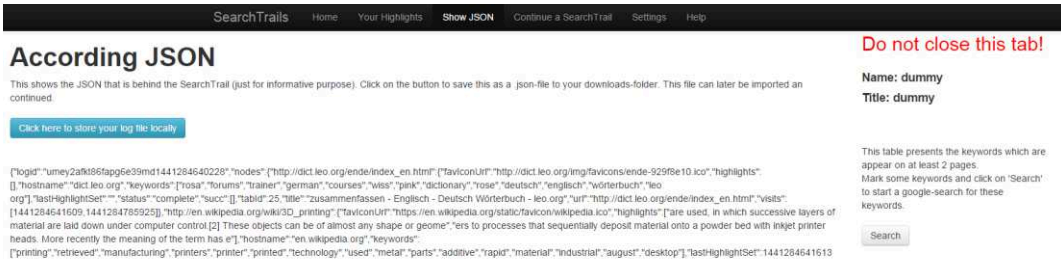


Figure 73: Show JSON view in the final version of SearchTrails.

Continue a search trail

In its first version, SearchTrails was not able to store and reload a search trail data object and to recreate the according search trail from it. I implemented this for the second version, together with the remote search trail storage function. A search trail can be retrieved in two ways: by fetching it from the remote search trail storage server, or by loading the according JSON object from the hard disk. Both ways are supported by the second version of SearchTrails (cf. Figure 74). In the Figure, one example ID is entered into the input field, which is retrieved from the search trail storage server. Alternatively, a file selector is offered which allows selecting the according JSON on the user's hard drive. The selected JSON data is shown in the textbox and the search trail is recreated.

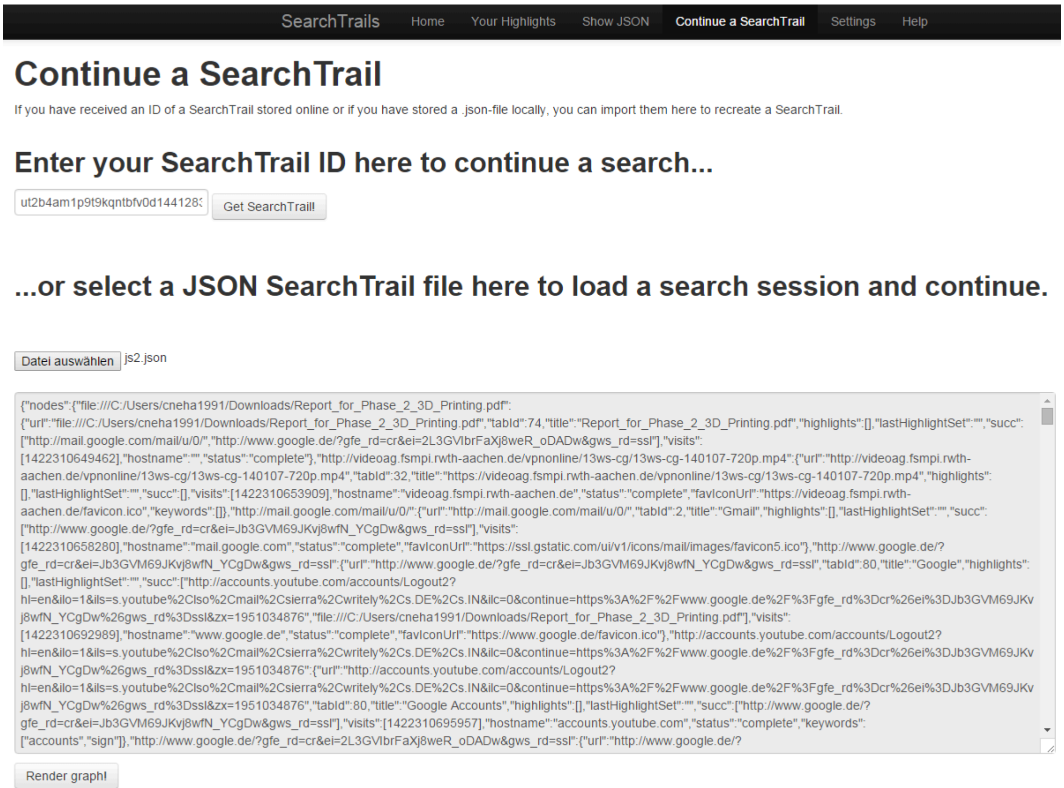


Figure 74: Continue a search trail view in the final version of SearchTrails.

Settings

For the second version of SearchTrails, I structured the ‘Settings’ page to make it more clear (Figure 76). The page now displays the search trail’s individual ID, and offers input fields for the username and the title of the search. When this information is entered and confirmed by pressing the ‘Save settings’ button, it is stored automatically as a text file in the browser’s downloads folder. The text file contains the following information (Figure 75).

```
Your search for your username: "user_5" and with the title:
"3D-Printing" is stored with the SearchTrails ID:
9sfj1hxqv95abbq6xwmrd1422310649086
```

Figure 75: Search trail data stored on the user’s local hard disk.

Furthermore, this text file contains the search trail ID to help the users remembering the unique search trail ID for recovering and continuing their search processes. Upon saving the settings, the username and the title of the search get displayed above the keyword table (cf.Figure 71).

Additionally, a check box is provided which allows starting and stopping the remote storage mechanism. For development purposes, not all trails need to be stored remotely, such that this option helps preventing unnecessary operations. For the user study, this option was not visible and the remote storage was enabled by default.

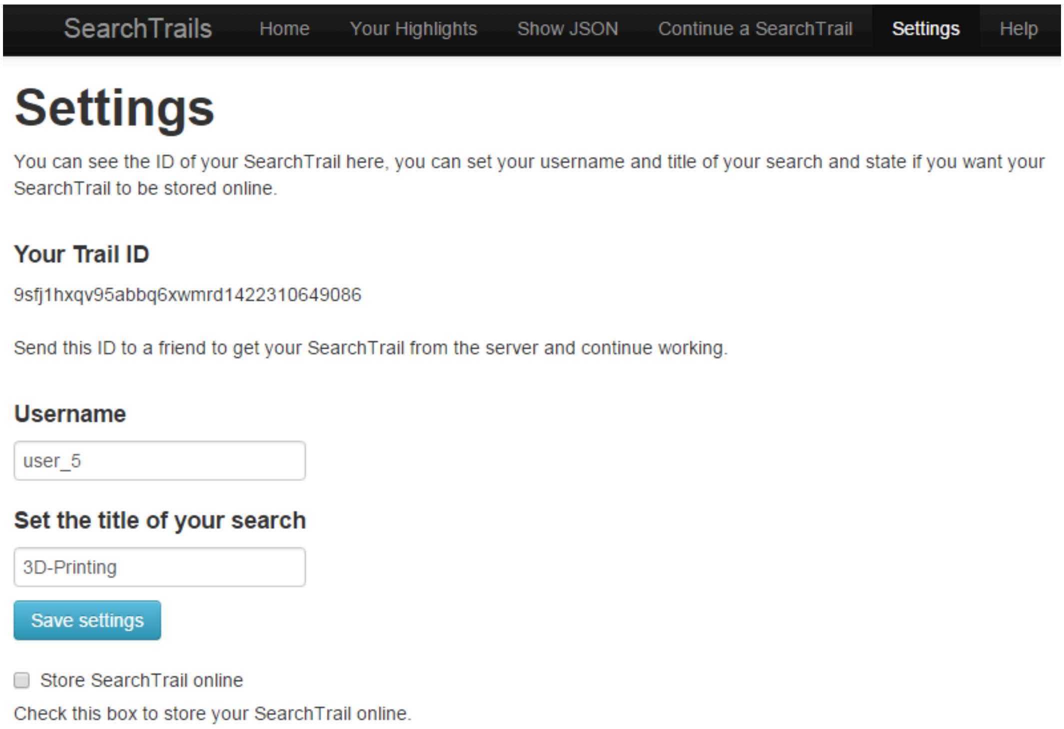
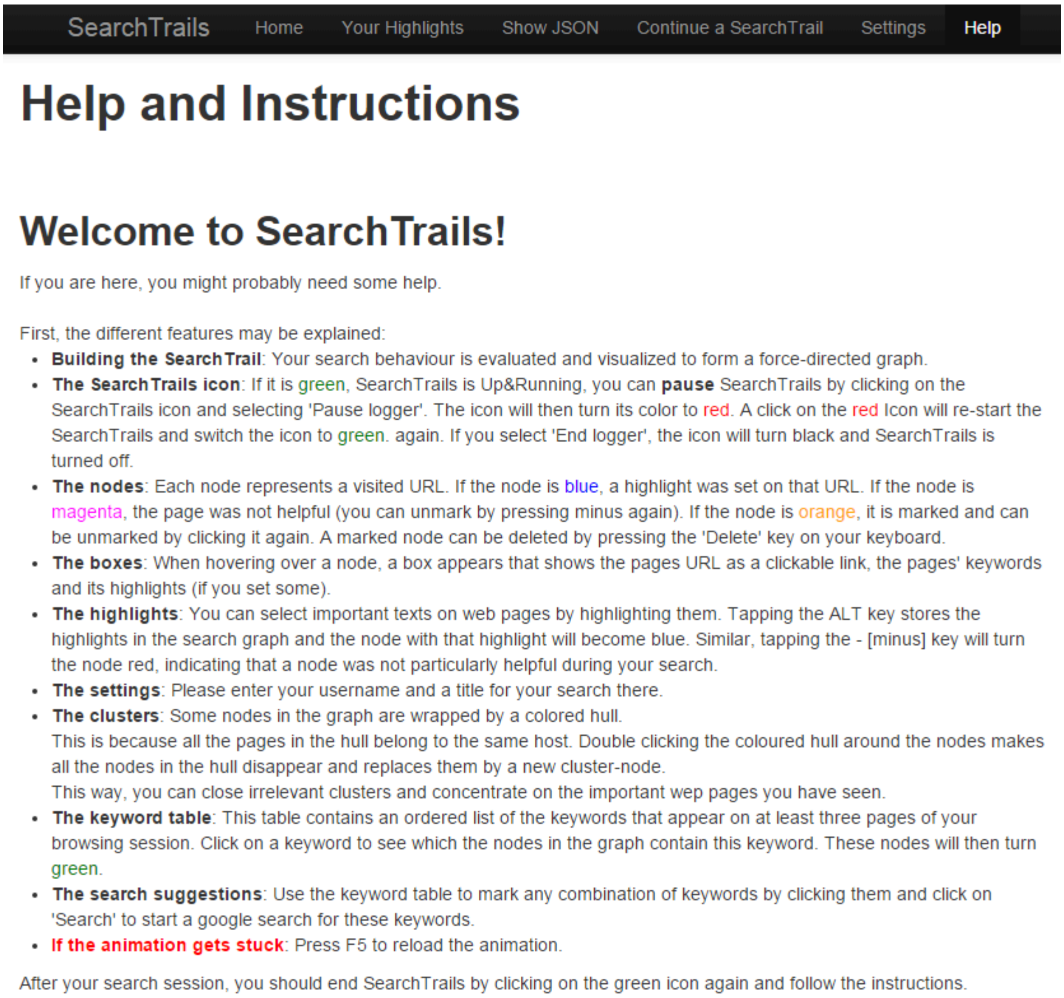


Figure 76: Settings page in the final version of SearchTrails.

Help

Compared to the first version, the 'Help' page was extended and improved. This page now displays information on all functions and explains the color coding of all colored items of SearchTrails. The page also explains how SearchTrails should be used to prevent losing any data (cf. Figure 77).


SearchTrails Home Your Highlights Show JSON Continue a SearchTrail Settings Help

Help and Instructions

Welcome to SearchTrails!

If you are here, you might probably need some help.

First, the different features may be explained:

- **Building the SearchTrail**: Your search behaviour is evaluated and visualized to form a force-directed graph.
- **The SearchTrails icon**: If it is green, SearchTrails is Up&Running, you can **pause** SearchTrails by clicking on the SearchTrails icon and selecting 'Pause logger'. The icon will then turn its color to red. A click on the red Icon will re-start the SearchTrails and switch the icon to green. again. If you select 'End logger', the icon will turn black and SearchTrails is turned off.
- **The nodes**: Each node represents a visited URL. If the node is blue, a highlight was set on that URL. If the node is magenta, the page was not helpful (you can unmark by pressing minus again). If the node is orange, it is marked and can be unmarked by clicking it again. A marked node can be deleted by pressing the 'Delete' key on your keyboard.
- **The boxes**: When hovering over a node, a box appears that shows the pages URL as a clickable link, the pages' keywords and its highlights (if you set some).
- **The highlights**: You can select important texts on web pages by highlighting them. Tapping the ALT key stores the highlights in the search graph and the node with that highlight will become blue. Similar, tapping the - [minus] key will turn the node red, indicating that a node was not particularly helpful during your search.
- **The settings**: Please enter your username and a title for your search there.
- **The clusters**: Some nodes in the graph are wrapped by a colored hull.
  This is because all the pages in the hull belong to the same host. Double clicking the coloured hull around the nodes makes all the nodes in the hull disappear and replaces them by a new cluster-node.
  This way, you can close irrelevant clusters and concentrate on the important wep pages you have seen.
- **The keyword table**: This table contains an ordered list of the keywords that appear on at least three pages of your browsing session. Click on a keyword to see which the nodes in the graph contain this keyword. These nodes will then turn green.
- **The search suggestions**: Use the keyword table to mark any combination of keywords by clicking them and click on 'Search' to start a google search for these keywords.
- **If the animation gets stuck**: Press F5 to reload the animation.

After your search session, you should end SearchTrails by clicking on the green icon again and follow the instructions.

Figure 77: Help page in the final version of SearchTrails.

This subchapter presents the paper prototypes and the three different versions of SearchTrails. I present the changes and improvements of the technical scope and the user interfaces during the development cycles, based on the technical introduction of the previous subchapters. I first describe the prototype SemanticClipboard which is based on different technical foundations than the later implementations of SearchTrails. I then describe the two different versions of the SearchTrails browser extension, and its development from a one-page user interface to the final version.

4.4 Conclusion and outlook

In this chapter, I develop the technical background of SearchTrails and explain the iterative implementation process of SearchTrails. The final architecture of SearchTrails implies several implementation decisions for the structure of the SearchTrails system, e.g. selecting the Google Chrome browser to start from. Incorporating several frameworks is necessary to achieve the desired functionality. A number of external frameworks are integrated for establishing the functionality of SearchTrails, such as frameworks for force-directed visualization, keyword extraction, or URL analysis. The search trail data object is generated during the search process and is stored on a remote server via the search trail storage mechanism. I developed SearchTrails during several design iterations, which converge towards the final functionality. The iterations range from early paper prototypes, a first software prototype called SemanticClipboard, and two versions of SearchTrails which I evaluate in two user studies. Both user studies and their results are described in Chapter 5.

With regards to the research questions, this chapter describes all the details of the SearchTrails system and as such answers research question RQ1.2 ‘Which functions does a system need to offer to the users to support capturing both the context as well as key findings of a web search?’. Chapter 4.3.3 answers this research question for the first version of SearchTrails, while Chapter 4.3.4 answers this research question for the second evaluated version of SearchTrails.

Chapter 5
Evaluation

Chapter 1 and Chapter 2 motivate this thesis and present an overview of related work for supporting discontinuous, asynchronous, collaborative, complex search. In Chapter 3, I evaluate the related work and develop a concept for the implementation of SearchTrails.

The analysis of the comparing tables from Chapter 2.5.1.2 and 2.5.2.2 in Chapter 3.1.2 shows that the concept of SearchTrails is a new and unique combination of approaches, combining both the logging and feedback for individual searches and the support for collaborative search. There exist approaches for logging search sessions, which do not reveal the search trails to the user, and other approaches for supporting collaborative search, that do not build logs during the search process. Taking an enriched search trail as a basis for collaborative work has not been done before.

Chapter 4 describes the technical details of SearchTrails and the implementation process that leads to the two final versions of SearchTrails. This chapter describes the evaluation of SearchTrails. I performed two user studies, first a quantitative study with 7 participants, and a qualitative study with 29 participants. The studies confirm that the tool SearchTrails provides a novel way to support asynchronous individual and collaborative complex search.

The results of the user studies are most important for answering the research questions developed in Chapter 1. The research question group RQ1 ‘Feasibility’ is finally answered by the evaluation of the results of the first, qualitative user study in Chapter 5.1.4. The research question group RQ2 ‘Value’ is finally answered by evaluating the results of the second, more quantitative user study in Chapter 5.2.5.

As the concept of SearchTrails is new, this implies a lack of comparable systems and complicates the evaluation of SearchTrails. The evaluation of complex search systems is itself still a research field. Early workshops on evaluating interactive information retrieval systems (ISSS) have been conducted in 2004 (Belkin et al., 2004). Additionally, it has been shown that evaluating systems like SearchTrails is complex: ‘The evaluation of visualization systems that can support exploratory search is not well understood’ (Koshman, 2006, p.21). Koshman proposes a set of five methodological factors that can be used for ensuring thorough evaluation of exploratory search visualization systems. These factors are:

1. **Field and laboratory testing**. I achieved this by a combination of the first and the second user study, where the first user study was performed in a lab-like environment (i.e. the participant’s offices). The second study was a field test where the participants worked on their tasks on their own computer without supervision.
2. **Longitudinal testing**. This was achieved by requesting each participant of the two studies to perform multiple searches.
3. **Investigator and user-derived tasks**. The tasks were investigator derived, but the users could choose tasks in the first study and concentrate on a specific area of their interest during the second study.

4. **The use of standard metrics**. I used standard metrics during the analysis of both studies, such as statistical measures, measures of graph complexity, and standard questionnaires for evaluating user experience.
5. **Subjective satisfaction measures**. These were collected by the use of self-designed questionnaires, explicitly asking for e.g. the perceived complexity of tasks, the feeling of support, or the impression of lightweightness.

To show the benefit of SearchTrails, I developed and performed two studies, which are described in the following subchapters. The first study was conducted in with 7 participants and confirms the conceptual foundations of SearchTrails with expert users in a qualitative manner. This study could show that the concept of capturing the course of a search by building search trails is an effective way of supporting individual complex search.

After the first study, I improved SearchTrails with respect to the continuation of search tasks and the support for collaborative search. Based on the new version, we[78] designed a second user study. This time, I performed the study with 29 participants, which allowed concluding some significant quantitative results. The second study adds to the first study that it reveals that the strengths of SearchTrails are also in the support for collaborative search processes.

The following sections describe each study's research questions, the conceptual study design, the actual implementation of the design, and the results with respect to the research questions of the study.

5.1 First SearchTrails user study

After the first implementation phase, I tested the developed version of SearchTrails with expert users in a first study with 7 participants. At that time, SearchTrails was able to capture the search trail of an individual complex search, its keywords and user-defined highlights, as well as clustering nodes according to the hosts they come from. Remote storage and the recreation of search trails were not possible with the first evaluated version of SearchTrails. The full scope of functions of the first version of SearchTrails is described in Chapter 4.3.3.

In the next section, I describe the hypotheses to be confirmed during the first study. Confirming these hypotheses helps answering the main research questions of group RQ1 'Feasibility' of my thesis in Chapter 5.1.4.

5.1.1 Hypotheses for the first SearchTrails user study

The first study was intended to show in a qualitative way that the concept of SearchTrails is a promising approach to keep an overview of personal search results, as well as a novel way for capturing important search results. Adding to that, the user-created search trails seemed to be

[78] When I write 'I' in the context of this chapter, the respective actions have been performed exclusively by me. The mentioning of 'we' means that the respective actions have been done by me with help or guidance of Wolfgang Prinz or Ulrich Norbisrath.

able to provide a fast overview of the users' interaction with the Internet to an evaluator. These envisioned results lead to the first set of four hypotheses for the first study.

- H1.1 – The support of complex search tasks by standard search engines is insufficient.
- H1.2 – Creating a search trail is a reasonable way for providing tool support for complex search tasks.
- H1.3 – SearchTrails is an effective tool for capturing important search results.
- H1.4 – The search trails are an effective approach for an evaluator to assess and evaluate the searchers' interactions with the Internet.

H1.1 confirms that the problem SearchTrails is going to solve actually exists and that SearchTrails therefore has a potential academic impact. H1.2 shows that the approach of building a search trail can be a key for providing support during individual complex search tasks. H1.3 confirms that SearchTrails can have an impact on synthesis and therefore helps the users to perform complex search tasks. H1.4 shows that the search trails are also valuable artifacts for evaluators evaluating other user's search results.

5.1.2 Conceptual study design

For testing the aforementioned hypotheses, a study was designed that confronted a limited set of expert users with complex search tasks on topics they were unfamiliar with. The participants should make use of SearchTrails and report about their experiences with it.

For the study, four search tasks were designed, of which three should be performed by each participant. The tasks were intended to be complex, with a varying degree of necessary learning during the complex searches. The tasks are as follows:

1. **'Beethoven'**
   Try to find 9 films, using Beethoven's 7th symphony as a score. Which part of this symphony is normally used as a score and what does it sound like?
2. **'Crypto'**
   Crypto currencies: What are they, what is their current status, how do you obtain them?
3. **'Video'**
   What is the state-of-the-art for video-glasses, and how would it be possible to build some on your own?
4. **'Ad'**
   Check on the manifold of German (or other countries) classified-ad-portals if there is a recent sale of a glassed frame in DIN A0 format, which you could bring home somehow. How much would it cost?

The first of the search tasks ('Beethoven') is a simple search task which is used to get acquainted with SearchTrails. It can be solved with limited effort. The easiest way to solve this task is finding the Wikipedia entry for that musical piece. It contains a list of films in which this piece is used as a score, and a visit to Youtube easily provides the user with listening examples. The 'Crypto' and 'Video' Search tasks were designed to be classical exploratory search tasks, as they require considerable learning efforts to solve the task. The 'Ad' search task is a complex search task. After finding out which product is wanted, it is the searchers' task to go through as many portals as possible and to always collect the same information, namely the availability and the price of the desired good. This task involves less learning than the 'Crypto' or 'Video' tasks, as the actual search is more repetitive in nature.

The tasks 'Crypto', 'Video', and 'Ad' are designed to fulfill seven desirable characteristics for complex search tasks to be used in evaluations with users by (Kules & Capra, 2008, 2009). Table 17 shows an overview of the seven characteristics and how the search tasks fulfill them.

Table 17: Search task characteristics and the search tasks of the first SearchTrails user study.

#	**Characteristic** (Kules & Capra, 2008, 2009)	**Fulfillment of the characteristics by the developed complex search tasks 'Crypto', 'Video', and 'Ad'**
1	'Indicate uncertainty, ambiguity in information need and / or need for discovery.'	The tasks are ambiguous in the information need and require information discovery.
2	'Suggest a knowledge acquisition, comparison, or discovery task.'	The tasks are focused on knowledge acquisition and information discovery.
3	'Be an unfamiliar domain for the searcher.'	The tasks are chosen such that they have a high probability of being new for the participants.
4	'Provide a low level of specificity about the information necessary and how to find the required information.'	The tasks are left very broad, such that a multitude of resources can be chosen for collecting information.
5	'Be a situation which the test persons can relate to and in which they can identify themselves.'	The tasks are intentionally left very broad, such that the participants can follow their own interest. Adding to that, the least interesting search task can be dismissed.
6	'Be a situation that the test persons find topically interesting.'	The tasks are chosen from popular technical domains, to be able to cover the interest of as many participants as possible.
7	'Be a situation that provides enough imaginative context in order for the test persons to be able to relate and apply the situation.'	The tasks are intentionally chosen in a way that they can match the interest of many participants.

I performed the first user study in three steps: In the first step, the users were personally introduced to get acquainted with the system. In the second and third step of the study, the participants performed one complex search task with and another one without the help of SearchTrails.

For performing the study, I made appointments with all seven participants to ensure that they had no time pressure. The study took place in the participants' offices and on their own personal computers to ensure a familiar environment. During the study, I was constantly available for questions or for providing help in case of problems. The study started with a short introduction to SearchTrails and the first search task ('Beethoven') was done under my supervision, as it was intended to be a training task for the participant to figure out the functions of and interaction with SearchTrails. After the first task, I registered that the participants were acquainted with SearchTrails. I left the room and the participants had as much time as necessary to complete the remaining search tasks.

Of the remaining three complex search tasks, the participants were asked to dismiss one to ensure that the least liked task not needs to be worked on. This was done to ensure personal interest in the remaining tasks. The participants performed the remaining two tasks in the second and third phase of the study. In the second phase, they performed the search task without using SearchTrails, while in the third phase they performed the search task with the help of SearchTrails. Not using SearchTrails meant that the participants left the SearchTrails browser tab open in the browser, but did not actively use its features. As I could not control that the SearchTrails tab was not looked at during the second phase of the study, I had to trust that the participants did not use SearchTrails. Unlike the second study, I did not ask the participants to write a report or to produce any other artifact, as the experiences with SearchTrails in comparison to unsupported search were in the focus of this study. Most of the participants took notes in one or the other way, but the participants' notes were not evaluated.

Before each search task, I asked the participants to estimate how difficult they expect the task to be. After each search task, the participants filled out a small questionnaire (cf. Appendix A1), asking them to judge answers to several questions on a 5-point Likert scale. The questionnaire included questions related to the necessary learning efforts during the search task, to self-awareness, and the experiences with SearchTrails. The questionnaire after the last search task was a bit more complex, as I included some general demographic questions and questions on the general experience with SearchTrails during the study in the questionnaire. Additionally, all participants filled out a User Experience Questionnaire (UEQ).

5.1.3 Implementation of the study design

The first user study was conducted with 7 participants, which worked as researchers or senior researchers in the domains of information technology and computer linguistics, and included three students which worked as research assistants. With each participant, an appointment was made to ensure the participants were not under stress for other reasons. At the start of each experiment, SearchTrails was installed at the participants' personal computer, and the functions of SearchTrails were explained to the participant as a hands-on tutorial, based on a written guideline. This guideline included the explanation of the general idea of SearchTrails, the introduction into the search tasks and the study, and the explanation of all functions, such as the search trail graph, keywords, or the interaction with the clusters.

The first search task ('Beethoven') was done by all participants, with me supervising the participants and explaining where and when SearchTrails features could be used. The second and third task were performed without supervision, but with me being available for questions. In the end, no participant needed to make use of this option. Table 18 gives an overview of each participant's selected topics and the support of SearchTrails during the evaluation phases.

Table 18: Overview of the tasks for each participant during the three phases of the study.

Phase	Participant	1	2	3	4	5	6	7
1	**Topic**	Beet-hoven	Beet-hoven	Beet-hoven	Beet-hoven	Beet-hoven	Beet-hoven	Beet-hoven
	SearchTrails support?	Yes	Yes	Yes	Yes	Yes	Yes	Yes
2	**Topic**	Ad	Video	Video	Crypto	Crypto	Crypto	Crypto
	SearchTrails support?	No	No	No	No	No	No	No
3	**Topic**	Crypto	Crypto	Crypto	Video	Ad	Ad	Video
	SearchTrails support?	Yes	Yes	Yes	Yes	Yes	Yes	Yes

The study took approximately 1.5-2 hours per participant, because every participant was guided personally through the first task, and then needed circa 30-40 minutes for each complex search task plus extra time for the questionnaires.

5.1.4 Study results

As this evaluation was the first evaluation of SearchTrails, the study setting was intentionally planned to take place in a smaller setting than the second study. The results are therefore qualitative. Nevertheless, they already provide valuable insights into the used and desired features of SearchTrails and their impact on individual complex search processes.

The demographics of the first study show that three students and four researchers participated, ranging from junior to senior researchers. All participants used search engines (mostly Google) to solve the complex search tasks in the study. We asked the participants to rate their familiarity with the search topics on a scale from 1 'not familiar' to 5 'very familiar'. For the 'Video' task, I computed an average value of 2.5 for the four performed searches, for the 'Ad' task, I computed an average value of 2.00 for the three performed searches, and for the 'Crypto' task, I computed an average of 1.71 for the six performed searches. This shows that the participants were generally not very familiar with the topics. It also indicates that the designed search tasks were suited for my study (cf. Table 17, search task characteristic no. 3). Another hint towards the complexity of the search tasks was given by a participant who stated that *'This search task was quite tedious, it required exploration of dozens of websites, changing search queries several times, and subdividing queries.'*, which implies that the search task characteristics 1, 2, and 4 in Table 17 are fulfilled.

For confirming hypothesis H1.1, I asked the participants' to assess their experience with search engine support during the search tasks ('Did the search engines you use support the way you are searching?'). The questions were rated on a 5-point Likert scale, reaching from 1 'not at all' to 5 'very good', and the average value was 3.86. In general, participants seemed to think positive about the way search engines support the way they were looking for search results. The participants liked the visualization of search results as lists (*'Long lists are perfect'*), the response time of search engines (*'Very fast results'*), search term suggestions (*'The suggestions* *coming up often help'*), or more technical ways of interaction with search engines, such as phrases or special search terms (e.g. 'site:de' to limit the search results to German pages).

When it comes to complex search, the participants' opinions become more critical. One free text question asked about the issues the participants do not like about search engines. In the answers to this question, the participants complain about the missing support during complex search tasks (*'There is no support'*), or ill-fitting search results (*'For complex search top answers not always contain expected results'*). Some participants are distracted by commercial advertisements (*'Don't like: Advertisements'*), others by the lack of context during longer searches (*'Don't understand what I'm searching'*). Especially the loss of context during longer searches is mentioned: *'But when the search requires exploration of multiple links, it becomes difficult to trace the path of your search'*. This statement is especially remarkable, as it shows that the participant perceives searching through the Web as a continuous trail with side branches. These branches are hard to explore, as ordinary web browsers just provide support for going back and forth, but not for the exploration of side tracks. Accordingly, these side tracks get lost during search, even if users want to get back to them.

Taking these statements together, participants seem to be able to solve complex search tasks with the help of standard search engines, using the limited capabilities of search engines in an effective way for fulfilling their complex search tasks. Criticism towards search engines is mostly based on a lack of context and therefore on missing long-time support during complex searches. I conclude that at least for the study participants, there is a general problem with search engines – not with the support for simple searches – but with the support for complex search tasks. These results provide first hints which point towards missing support for complex search tasks among our participants. Therefore, **hypothesis H1.1 can be confirmed**: The support of complex search tasks by standard search engines is insufficient.

I asked the participants for their opinion on the value of the search trail visualization in the form of free-text questions. The overall feedback was very positive. Participants liked the idea (*'Tool provides a good visualization of the search results'* and *'I like the idea of search visualization & 'path through the internet''*) as well as the actual force-directed graph visualization (*'See a 'new way of searching' & search-history visualization'*). Besides its character as a novel way for visualizing a search, the search trail was also appreciated as a means for keeping track of own search locations (*'Keeping track of visited sites and structured browsing'* and *'Visualize the path especially when stepping back'*).

Another important finding is that the participants found that the visualization can provide ways to get back to formerly visited pages (*'The highlighting was helpful to 'save' some information and take notes; I could go back to those'*) and how this may have an effect on the efficiency of future search tasks (*'Keeps track of selected information, which contributes to more effective search in future. This feature allows remember sources where main points were found'*). Other participants appreciated the value for of search trails for complex search tasks in contrast to simple search tasks (*'SearchTrails shows a nice visualization of the history of browsing for a complex topic'*).

The participants' statements show that for the selected complex search tasks the visualization in form of a search trail can offer several benefits during the search process. These are the visualization as a search trail itself, but also an easy way to get back to previous search results, benefits for more structured browsing, or possible help during future searches. Even if the number of participants in the first user study was limited, these statements provide a strong indication that **hypothesis H1.2 can be confirmed**: Creating a search trail is a reasonable way for providing tool support for complex search tasks. These results also help answering research question RQ1.3 'Does a system with the functions from RQ1.2 provide help and support for users during complex search processes?' from Chapter 1.4 positively.

I asked the participants to identify the particularly helpful features of SearchTrails during the complex search tasks with the help of free-text questions. This question aimed at assessing the participants' experiences with SearchTrails during their complex search tasks and the impact of SearchTrails on their search process. The participants' answers can be clustered with respect to several aspects of search support, namely the saving results for later reuse, the highlighting function, the keyword extraction, and the clustering.

With respect to the functions for storing search results, participants could agree that SearchTrails was used for 'copying and storing the search result' and it was used for note taking (*'Using it [SearchTrails] as a note taking tool'* and *'Storing text snippets from the web pages and pdf-documents'*). This already indicates that SearchTrails can replace traditional methods for taking notes during complex search. A long-term effect was detected by one participant, who used SearchTrails for *'saving results for later reuse'*. Other participants could see additional use cases for SearchTrails for *'book or paper excerption'* and the *'creation of reference list for research papers (or any papers)'*.

The feature which the participants like most is the highlight function. Comments on this are very positive (*'The highlighting function is VERY helpful'*). The highlights are used as well for collecting resources (*'Highlight function helps easier navigation and reference collection'*) as for organizing them (*'Visualization of visited pages. Keep track of highlights. Grouping sources'* and *'Highlighting search results, tagging, modifying search queries (Keywords)'*). It turns out that the highlights also seem to have a reminder function, that helps participants moving back to valuable resources and to go into new search directions from there: *'Highlighting feature was very handy. Since the first link only contained 4 out of 9 films, I marked them as preliminary search results and could locate [the] original source quickly'*.

The overall opinion towards the keyword extraction was positive (*'Keyword list was helpful'*), although there were some complaints about irrelevant keywords (*'Keyword list was helpful, but contained irrelevant entries'*). Participants like the added value of getting an overview of relevant keywords (*'See most important keywords of the result pages'*) and use the keywords for generating new search queries (*''Finding' new search queries based on keywords'* and *'Interesting topic. Acquired new knowledge, used search by keywords'*).

Also the clustering of graph nodes based on the websites' hosts is used to get a better overview of search results. Participants state that they used the features *'clustering pages, saving answers'* and use the clustering for concentrating on more important resources: *'In addition to clustering to clarify the graph, I wanted to remove or hide some nodes'*. One participant could also see application of SearchTrails for sharing results between collaborating users: *'Keeping* *list of references. Teaching / learning about some topic using visualization and highlighting. Using and sharing search results for improving search'.*

These comments on the most important functions of SearchTrails and their application during complex search tasks can **confirm hypothesis H1.3**: SearchTrails is an effective tool for capturing important search results. The main findings from hypothesis H1.1, H1.2, and H1.3 were compiled to a full paper that was handed in to the 'Mensch und Computer' conference. The paper was accepted and presented at the 'Mensch und Computer 2014' conference in Munich (Franken & Norbisrath, 2014b). Together with the comparing tables in Chapter 2.5.1.2 and 2.5.2.2, the technical requirements from Chapter 3.2, and the implementation details from Chapter 4, these results positively answer research question RQ1.2 'Which functions does a system need to offer to the users to support capturing both the context as well as key findings of a web search?'.

The search trails that were generated by the study participants during the search tasks were stored on a remote server and evaluated afterwards. The search trails were technically stored as JSON-files, which were recreated by the improved version of SearchTrails after the study to form force-directed graphs again. During the evaluation, we found that the search trails do not only offer benefits for the searcher, but also for experts evaluating the search trails. The visual inspection of the recreated search trails could both reveal the quantity and the quality of the search results. The evaluation of keywords, highlights, and the force-directed graph itself immediately reveal the scope of the participants' search, the results, and the covered facets of the search. In contrast to traditional notes or bookmarks, the search trails also reveal which resources the searcher has visited, but did not find any relevant results.

After the study, all 21 search trails from the three search tasks for the seven participants were evaluated by me and Ulrich Norbisrath during a remote workshop session. As these trails were split upon four topics, no statistically significant results could be gathered. The evaluation of the search trails revealed which participant conducted a successful search with respect to quantity and quality of the search results. As an example, two search trails from the 'Beethoven' task can stand as examples for more or less intensive searches (cf. Figure 78 and Figure 79).

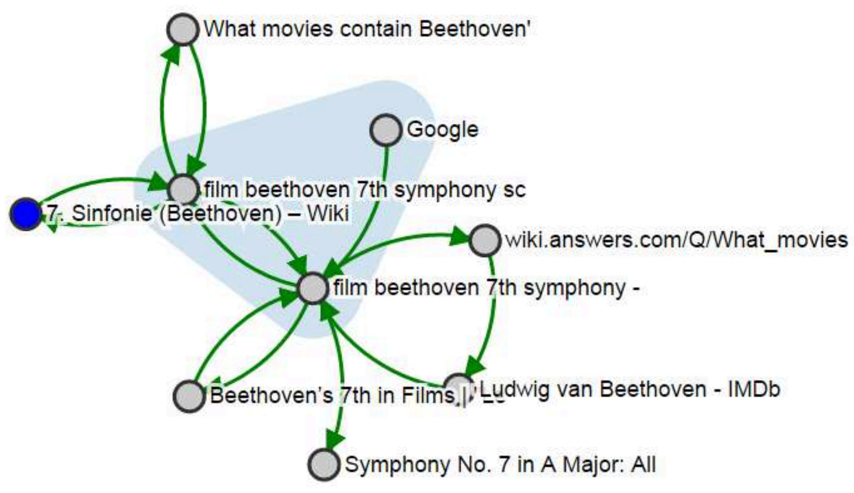


Figure 78: Inefficient example of a search trail for the 'Beethoven' search task.

Figure 78 shows an inefficient example of a search trail for the simple tool-supported 'Beethoven' search task. The search trail shows a star-shaped approach, starting from a search page (here 'Google', within the cluster) and checking the search results for the query 'film beethoven 7th symphony'. The searcher seems unhappy with the results, and refines the query by appending the term 'score'. From the search results, the Wikipedia page on the 7th symphony is accessed and a list of the movies using the symphony as a score is highlighted. This search is successful in fulfilling the minimal requirements set in the task description, but less efficient as the searcher needed to make several approaches to find a satisfying resource and left out Youtube.

In contrast to this search, the more efficient tool-supported 'Beethoven' search trail in Figure 79 shows a shorter but more thorough search approach. A look into the participants' highlights reveals that an extensive list of films was found very early, but the searcher goes further and listens into the specific piece on Youtube. The searcher has visited less pages compared to other search trails, but has found a more thorough set of information on the given search task. This search was successful and fulfilled our expectations, but was done with a minimal effort, indicating high efficiency of the user.

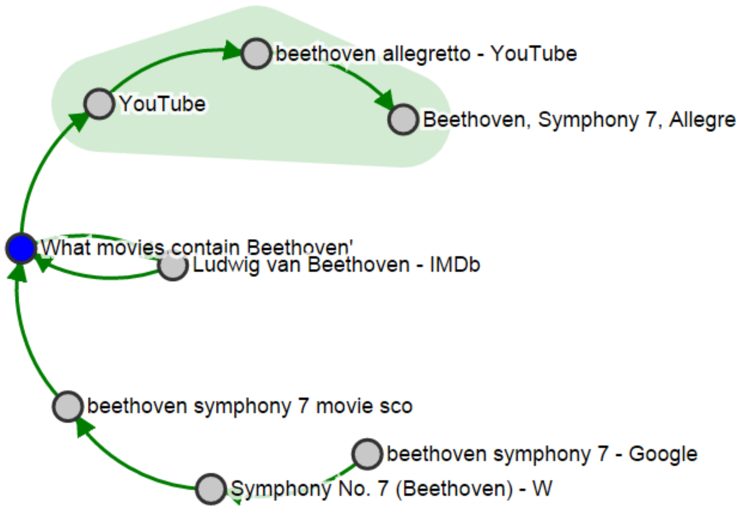


Figure 79: Efficient example of a search trail for the 'Beethoven' search task.

Similar findings can be gathered for the complex search tasks. Figure 80[79] shows an example of a successful search trail for the 'Crypto Currency' task without tool support. The participant starts with a Google search on 'crypto currencies' (on top), discovers the Wikipedia page on it, and reads through different Wikipedia pages (i.e. for 'Fiat money' and 'Bitcoin'). The participant then proceeds to more specific topics, such as 'First steps with bitcoins' and visits several sites on them (as indicated by the cluster). The participant then goes back to Google, and investigates the current status of crypto currencies. This search trail shows how the searcher gets acquainted with a topic, learns about it, and then moves on to new search terms to discover facets of a topic that are individually interesting for the searcher.

[79] The figure shows a node with the title 'Oops! Google Chrome could...' in the upper left part of the search trail. This is no sign of an error of SearchTrails, but of the user trying to access a site which is not found by the browser. Therefore, Chrome returns a warning, and the user return to the initial site, which is depicted in this part of the search trail.

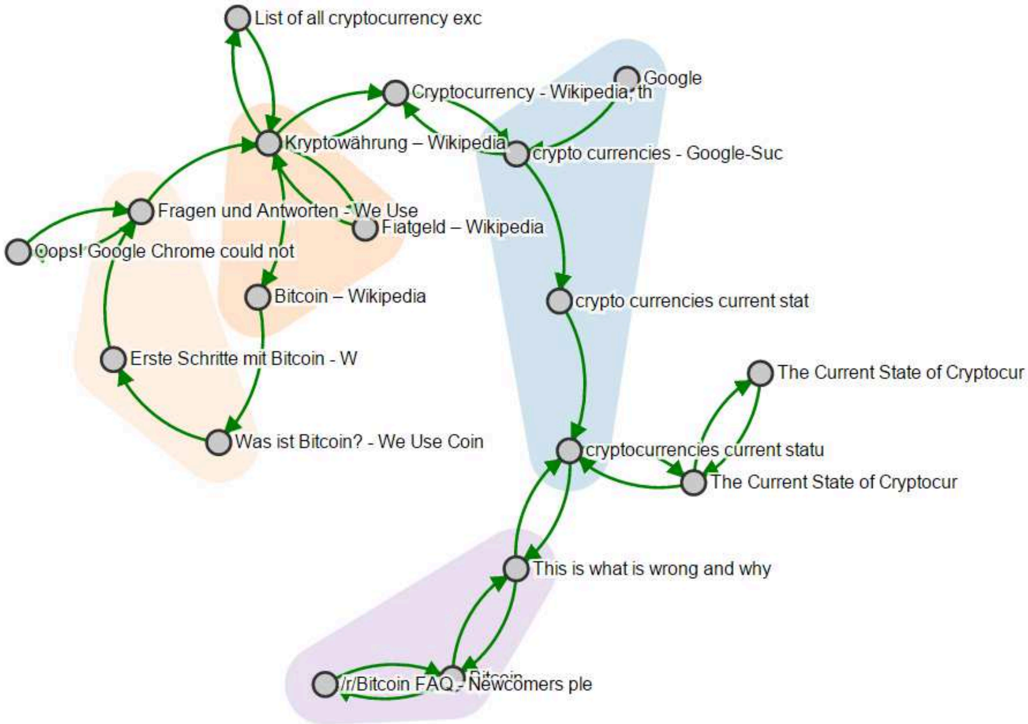


Figure 80: Example search trail for the unsupported 'Crypto' search task.

Figure 81 shows a search trail for the tool-supported 'Crypto' search task. This participant starts with a Google search (to the right), and visits the Wikipedia pages on 'Bitcoin' and 'Cryptocurrency' first. The participant then returns to the search page and alters the query to 'how to obtain crypto currencies'. From the new results page, several links are followed. This behavior is repeated two more times. Several blue nodes show that the participant has set highlights on the respective pages, thus marking valuable information which can be assessed by the searcher and the evaluator on the highlights overview page. This search turned out to be very successful in terms of the number of visited pages and the found results.

Figure 80 and Figure 81 show a search trail for the tool-supported 'Crypto Currency' search task. During the evaluation, we saw for the complex search tasks that the visually perceived complexity of the search trails seems very similar, independent if the search tasks are performed with or without SearchTrails. The search trails for these tasks are much more complex than the search trails for the simple 'Beethoven' search task. A comparison of the statistics of all search trails (in terms of the number of nodes, the number of different hosts, the number of recorded events, and the number of keywords) also returns this result: The complex search tasks result in much higher values than the simple 'Beethoven' task. The average values for the supported and non-supported search tasks are similar in terms of statistical values, but for six of seven participants, the statistical values are slightly higher for the search trails of the supported search task.

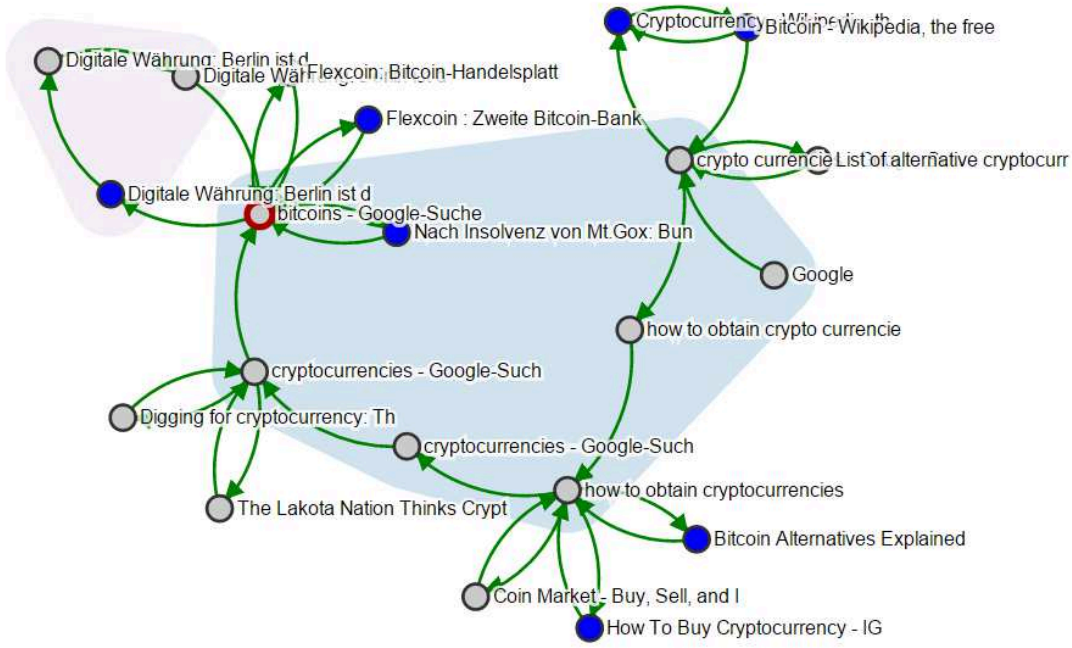


Figure 81: Example search trail for the tool-supported 'Crypto' search task.

Considering the search trails for the 'Ad'-task, it can be seen that they are mostly large in terms of number of nodes, edges and clusters and score high in the mathematical measure of complexity according to (Minoli, 1975). Table 19 shows the average complexity for the four different tasks. These results go along with the overall impression of the search trails.

Table 19: Overview of the average complexities for the four different search tasks of the first user study.

Task	n	Average complexity (rounded)	log(Average complexity)
Beethoven	7	485	2.69
Video	4	1083959	6.04
Crypto	7	17577072	7.24
Ad	3	308761068	8.49

An example search trail from an 'Ad'-task is shown in Figure 82. The high complexity of these trails could be due to the fact that these tasks are solved by repetitive work. Most of the times, the participants start by searching for small ad portals, immediately access the GUI of these portals, and start searching for the requested items. The participants therefore often perform similar searches in all visited portals. This leads to long loops within the clusters of the graph. These search trails seem to grow very fast, as the work requested from the participants is mostly repetitive, and there is no learning necessary, as the goal of the search stays the same. The size of the search trail therefore depends on the number of portals visited, and not on a learning effect. None of the participants did investigate the transport possibilities for the requested item; this part would have involved more creative solutions and (maybe) learning. We assume that the participants forgot about this part, or thought it would be too time intensive and made the decision to skip it. Still, by inspection of the clusters and the highlights, evaluators can easily judge the quality and the quantity of the performed search activity.

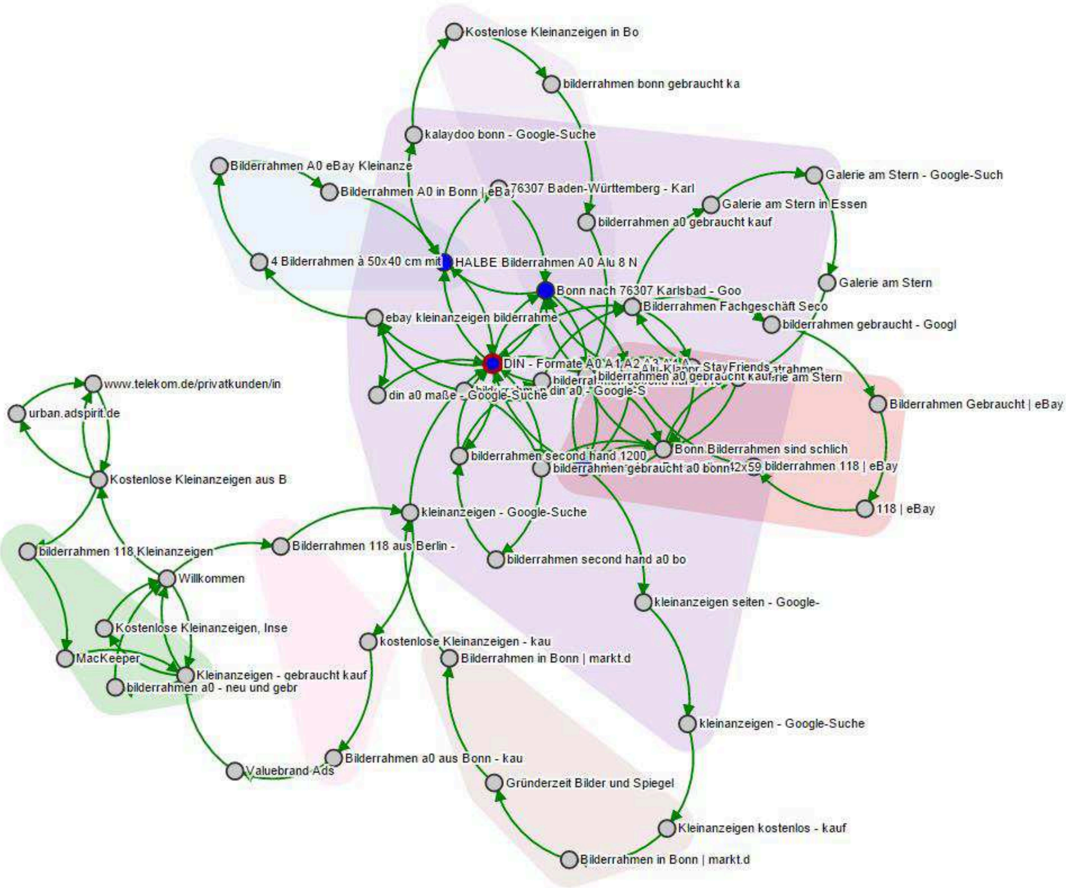


Figure 82: Example complex search trail for the 'Ad' search task.

The study results for evaluating the search trails with respect to quantity and quality of search results are very promising and can **confirm hypothesis H1.4**: The search trails are an effective approach for an evaluator to assess and evaluate the searchers' interactions with the Internet, which at the same time implies that the search trails are highly suitable for resembling the course of a user's web search. These results were compiled to a full paper that was handed in for the 24th Annual International Conference on Computer Science and Engineering (CASCON) in 2014. The paper got accepted and was presented in November, 2014 in Toronto, Canada (Franken & Norbisrath, 2014a). Together with the results on historical origins and recent trends in search trails in Chapter 2.1 and 2.7 and the technical requirements from Chapter 3.2, these results help answering research question RQ1.1 positively: 'Are search trails a means to resemble the course of a user's web search?'.

In addition to the questionnaire aiming at the functionality and the support for complex search tasks, an evaluation with respect to the user experience was conducted. In order to achieve this, we used the User Experience Questionnaire (UEQ) (Laugwitz et al., 2008). The UEQ is a standardized test for evaluating the user experience of interactive products. It was presented to the participants after they performed all search tasks. The UEQ consists of 26 pairs of contrasting attributes, each being separated by a 7-point Likert scale. An example for this is the attribute pair 'inventive / conventional'. Participants are encouraged to express their opinion towards each of the attribute pairs spontaneously.

For the evaluation of the UEQ results, the 26 item pairs are grouped into six properties of the evaluated system, which are ‘Attractiveness’, ‘Perspicuity’, ‘Efficiency’, ‘Dependability’, ‘Stimulation’, and ‘Novelty’. For each of these properties, the average value, its variance, its standard deviation, and its confidence interval are calculated on a 5% error level. The standardized evaluation scheme of the UEQ also provides a benchmark rating, which compares the evaluated system with the average results of 163 studies with 4818 participants.

We presented the UEQ to all seven participants after the last phase of the study, and evaluated the results with the standardized UEQ evaluation scheme. The diagram in Figure 83 shows the average value for each system property and the respective confidence interval. The diagram is scaled from -3 to 3, to match the 7-point Likert scale. Due to the calculation of the average value, values below -2 and above +2 rarely emerge. In general, values below -0.8 are considered negative, while values above 0.8 are considered positive. The values between -0.8 and +0.8 are considered neutral (UEQ Evaluation Sheet, 2015).

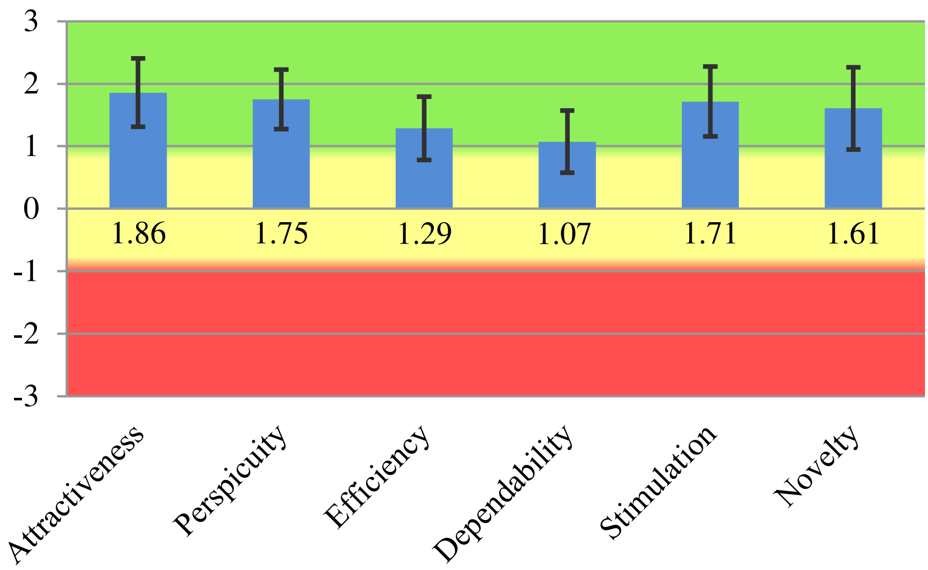


Figure 83: Results of the UEQ evaluation for SearchTrails in the first user study.

Figure 83 shows that SearchTrails scores positive (larger than 0.8) for all of the six UEQ properties. The depicted confidence intervals show the range of values in which the real value is located with a probability of 95%. For most of the properties also the confidence interval is still within the positive range of the diagram. The two properties scoring lower are ‘Efficiency’ and ‘Dependability’, but even the bounds of their confidence intervals still score neutral. An explanation for this could be that the participants used SearchTrails for their own searches, and the use of SearchTrails did not result in a significant increase of efficiency. A gain in efficiency may come into effect when the results of individual searches can be exchanged between users searching on the same topic, or during search tasks that run longer than the search tasks in the study and span multiple sessions, which was not possible with this version of SearchTrails. In these cases, a gain in efficiency is more likely than in single-user, single-session use cases. The second property scoring lower is ‘Dependability’, which may be caused by the force-directed graph, which is the main graphical element in the SearchTrails visualization. The force-directed graph was chosen purposely, as it is a powerful way for visualizing large amounts of data (Eades & Huang, 2000). However, the constant rearranging of the force-directed graph may create the overall impression of SearchTrails being less controllable than other systems.

In order to find out how meaningful the derived values for each property are, Cronbach's alpha value (Cronbach, 1951) is calculated for each of the six properties' subscales. Table 20 shows Cronbach's Alpha value for each of the properties. Within the statistics literature (Kline, 2000; George & Mallery, 2014), a value above 0.7 is generally accepted to provide a sufficient consistency and therefore a reliable value for the according scale. Nevertheless, there is also criticism on this usage of a cut-off criterion (Schmitt, 1996). The values for all properties except 'Efficiency' and 'Dependability' score close to or above 0.7, indicating that these values can be accepted as reliable indicators for the respective properties. Similar to the section above, the values for 'Efficiency' and 'Dependability' could not be tested reliably. The according values in Figure 83 may therefore only be hints towards the real opinion of the participants, but are not as reliable as the other values.

Table 20: Overview of Cronbach's Alpha values for each of the six properties from the UEQ.

Attractiveness	Perspicuity	Efficiency	Dependability	Stimulation	Novelty
0.82	0.65	0.57	0.45	0.79	0.95

The comparison of the UEQ results for SearchTrails with the benchmark results of other systems shows a very positive tendency in favor of SearchTrails. Figure 84 shows the benchmark diagram for SearchTrails in comparison to the average values of 163 studies with together 4818 participants. This diagram shows that SearchTrails could score 'Good' or 'Excellent' in all properties except 'Efficiency' and 'Dependability', but even for these properties, the values are still 'Above average'.

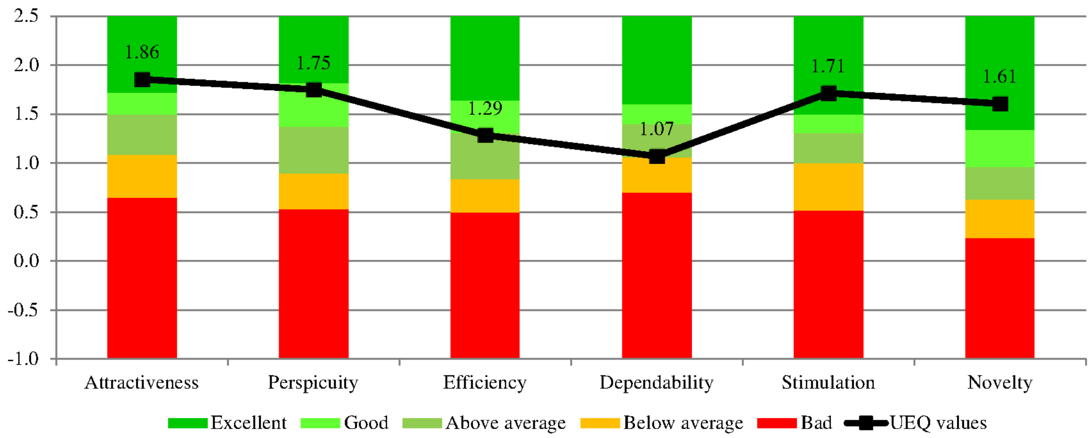


Figure 84: Benchmark comparison for SearchTrails from the first user study.

5.1.5 Limitations

Although the results from the first user study are very positive, some limitations of this study have to be considered. It has to be pointed out that both the qualitative analysis and the UEQ results are not statistically significant due to the limited number of participants. However, the results are first hints towards the general acceptance of SearchTrails and can serve as anecdotal evidence for validation of the concept and approach of SearchTrails. Nevertheless, these results can show a tendency towards more representative results, as they were generated by professional users with intrinsic motivation and without time pressure. Some participant's comments already point into the direction that SearchTrails is more suited for *'Complex searches, long time searches'*, and its features can hardly be appreciated during shorter search tasks: *'SearchTrails seems to be better suited for long running search tasks than for short searches'*.

The treatment of the participants also bears the risk of biasing the study towards more positive results, as every participant was personally introduced into the topic. Nevertheless, even if the amount of participants was small, the participants themselves were a representative sample of the envisioned user group of the system. The participants had experience with complex search tasks during their professional work and could therefore appreciate SearchTrails' functions. Due to their professional work, the participants were used to long-running search tasks where search results from several domains have to be assembled to form a larger picture. For the second user study, it is less likely to be able to work with a similar set of participants with professional expertise in complex search tasks (cf. Chapter 5.2.6 'Limitations').

From a technical point of view, the evaluated version of SearchTrails was not able to recreate search trails and was therefore able to support only one search session for a single user. Therefore, the envisioned time-saving and effort-saving effects of the exchange of search trails could not be shown with this study setting. Similarly, the UEQ can show a tendency towards realistic values, but are hardly statistically significant, as can be seen by the values for 'Efficiency' and 'Dependability'. A larger number of participants would be necessary to fully confirm the positive UEQ results.

5.1.6 Conclusion

The first user study of SearchTrails returns very satisfying results. All four hypotheses can be confirmed. The first three hypotheses deal with the existence of a general problem considering the support of standard search engines for complex searches, the general idea of SearchTrails, and its benefit for taking notes during complex searches. The fourth hypothesis can confirm that the generated SearchTrails are a promising way for evaluating the results of a complex search task. Adding to that, an evaluation of SearchTrails with respect to usability confirms that also from this point of view SearchTrails can score very well and provide an added value during the work on complex search tasks.

With regards to the research questions developed in Chapter 1.4, this subchapter can finally answer the first group of research questions (**RQ1, 'Feasibility'**), which are: RQ1.1 **'Search trails valid?'**, RQ1.2 **'Which functions?'**, and RQ1.3 **'Provide help?'**.

- **RQ1.1: ‘Are search trails a means to resemble the course of a user’s web search?’** can be answered positively by the study results on hypothesis H1.4 in Chapter 5.1.4. The search trails do resemble the course of a user’s web search in a meaningful way.
- **RQ1.2 ‘Which functions does a system need to offer to the users to support capturing both the context as well as key findings of a web search?’** can be answered positively by the study results on hypothesis H1.3 in Chapter 5.1.4, as SearchTrails is an effective tool for capturing important search results within their context.
- **RQ1.3: ‘Does a system with the functions from RQ1.2 provide help and support for users during complex search processes?’** can also be answered positively by the study results on hypothesis H1.2 in Chapter 5.1.4, as users stated that SearchTrails is an effective tool for providing tool support during complex search processes.

For the next implementation phase, the additional comments in the SearchTrails evaluation questionnaires are taken as a basis for the development of further features. These features mostly aim towards the recreation of search trails and therefore the continuation of complex search tasks. These features allow asynchronous collaboration by the exchange of search trails between users.

5.2 Second SearchTrails user study

While the first user study has the character of a qualitative expert study, judging the core principles of SearchTrails, the second user study is intended to corroborate these concepts with 29 participants in a more quantitative manner. Based on the experiences and the feedback gathered during the first user study, I substantially improved SearchTrails for the second version. For the full details of the second evaluated version of SearchTrails, please refer to Chapter 4.3.4. The version of SearchTrails evaluated during the second study allows the deletion of nodes and highlights, the manual creation of highlights, the remote storing of search trails, and the continuation of search trails. These features enable splitting a search task into several sessions and allow the exchange of search trails between users. The extended set of features allows conducting a second user study, aiming for more statistically significant results, comparing the work with and without SearchTrails, and the support for asynchronous, collaborative work. The following sections describe the second user study and its results in detail.

The hypotheses presented in the following section help answering the main research questions of this thesis from group RQ2 'Value', which I develop in Chapter 1.4. The detailed analysis of the study results in Chapter 5.2.5 answers these research questions.

5.2.1 Hypotheses for the second SearchTrails user study

The second version of SearchTrails was evaluated in a larger setting than the first version of SearchTrails. Therefore, a study design was developed that allowed an implementation with a larger number of participants and less personal guidance during the study. This study design was intended to confirm or reject the following hypotheses:

- H2.1 – SearchTrails eases single user complex search tasks by supporting the threefold of aggregation, synthesis, and discovery.
- H2.2 – SearchTrails raises awareness for the own personal search behavior.
- H2.3 – The use of SearchTrails improves the quality of the search process for complex search tasks.
- H2.4 – SearchTrails enables asynchronous collaborative search by exchanging search trails.
- H2.5 – Exchanging search trails in collaborative search improves the quality of search results.

H2.1 focuses on confirming that SearchTrails has a positive impact on all key components of complex search, while H2.2 confirms that SearchTrails can raise the users' awareness during complex search tasks. H2.3 shows that the use of SearchTrails can increase the quality of the search results. The last two hypotheses show the impact of SearchTrails on collaborative search. H2.4 confirms that collaborative search is possible with SearchTrails, while the H2.5 shows that collaborative search in combination with the use of SearchTrails has a positive impact on the outcome of a search process.

5.2.2 Conceptual study design

To confirm or reject the aforementioned hypotheses I planned a second user study to take place with attendants of a university lecture. The study consists of two phases of search, in which the participants should each search for specific topics with the help of SearchTrails. During both phases, a summarizing report should be prepared in a way that someone who did not perform the search would be able to have an overview of the found results and to understand the key results of each search task by reading the report. During the first phase, the study covers individual search to confirm or reject hypothesis H2.1, H2.2, and H2.3. During the second phase, the participants build upon given search results from the first phase to confirm the hypotheses dealing with collaborative search, H2.4 and H2.5. For the study, the participants' private PCs and laptops were used to create a very natural and familiar environment for the search tasks. We decided against a computer lab as a study setting, as this would have created a very artificial environment for search, lacking familiarity and the tools the participants normally use.

Before the start of the first phase, I did a short presentation, consisting of a short motivation for the study, a brief overview of the features of SearchTrails, the two phases of the study, and the requested artifacts. Additionally, a short offline demonstration of a search trail, its features, and interaction possibilities was given. This was done briefly to avoid influencing the participants too much towards the system. The participants were handed out an initial questionnaire that evaluated the personal and educational background, the experience with web search, search strategies, and the expectations towards the first given search task.

For the study, two complex search tasks were developed that both fulfill the seven desirable characteristics for complex search tasks by (Kules & Capra, 2008, 2009). We decided against taking topics from the lecture (e.g. searching about Petri nets) for search topics, as we saw the risk that thorough result pages or essays about these topics could be easily available. The two developed topics were intended to raise the participants' interest and to be broad enough for everyone to find a specific area of interest in them. Providing two topics also helped to reduce the risk of biasing the search results by the selection of just a single topic. If the search results do not differ significantly between the topics, it can be assumed that the search tasks themselves did not have an influence on the results. The two search tasks for the first phase of the study were as following:

1. **Search task '3D_1': Topic '3D printing'**
   '3D printing is talked about a lot these days. Research on the history of 3D-printing, its current projects and future work of it. Find theoretical foundations, example projects from different application areas, and future developments.'
2. **Search task 'HA_1': Topic 'Home automation'**
   'You want to control and automate your student flat from the remote. Research on which affordable sensors and actors you could equip your flat with, what hardware prerequisites would be necessary, and on which standards and protocols you would have to rely on. What are potentially nice use cases to implement (and show off)?'

Table 21 gives an overview of how the search tasks from the first and second phase fulfill the desirable characteristics for complex search tasks by (Kules & Capra, 2008, 2009). In order to avoid a duplication of this table, the characteristics of the search tasks from the second phase are also described in this table, but referenced later in this section.

Table 21: Search task characteristics and the search tasks of the second SearchTrails user study.

#	**Characteristic** (Kules & Capra, 2008, 2009)	**Complex search tasks from the 1st phase: 3D_1 and HA_1**	**Extended complex search tasks from the 2nd phase: 3D_2 and HA_2**
1	'Indicate uncertainty, ambiguity in information need and / or need for discovery.'	The tasks were ambiguous in the information need and needed information discovery.	The extra question was intentionally left broad to create a need for discovery.
2	'Suggest a knowledge acquisition, comparison, or discovery task.'	The tasks were focused on information discovery and comparison.	Despite the need for acquiring knowledge on the topic in general, the extra question required even discovery of additional facts.
3	'Be an unfamiliar domain for the searcher.'	The tasks were chosen such that they had a high probability of being new for the participants.	As the tasks were switched between the groups, they were still new to the participants.
4	'Provide a low level of specificity about the information necessary and how to find the required information.'	The tasks were left very broad, such that a multitude of resources could be chosen for collecting information.	Adding to information from the first phase, even for the second phase lots of resources could be chosen to find the desired information.
5	'Be a situation which the test persons can relate to and in which they can identify themselves.'	The tasks were intentionally left very broad, such that the participants could follow their own interest.	The additional tasks were chosen to be related to the everyday life of the participants or have a high probability of being interesting to them.
6	'Be a situation that the test persons find topically interesting.'	The tasks were chosen from a popular technical domain with a multitude of applications, to be able to cover the interest of as many participants as possible.	Similarly to the first phase, the extra question was chosen to still leave room for personal preferences.
7	'Be a situation that provides enough imaginative context in order for the test persons to be able to relate and apply the situation.'	The tasks were intentionally chosen in a way that they could match to the interest of the participants, e.g. by asking for personal preferences.	As the additional tasks were more focused on an application domain, this criterion was even better fulfilled than for the first phase.

During all phases of the study, the participants are separated into four groups and stay assigned to the same groups. During the first phase, two groups work on the topic '3D printing', and two groups work on the topic 'Home automation'. From each topic, one group uses SearchTrails actively, and the participants are able to make full use of SearchTrails' features. The other group of each topic is given a version of SearchTrails that does not show any information to the users; SearchTrails just serves as a logging engine. Depending on the usage of SearchTrails during the first phase, the topic during the first phase, and the type of artifact in the second phase we assign the following mnemonic codes to the groups:

- **Group SR3** is the group that uses SearchTrails (S) in phase 1 and receives a report as an artifact for phase 2 (R). The group works on the topic '3D printing' (3) in phase 1, and 'Home automation' in phase 2.
- **Group SRH** is the group that uses SearchTrails (S) in phase 1 and receives a report as an artifact for phase 2 (R). The group works on the topic 'Home automation' (H) in phase 1, and '3D printing' in phase 2.
    - We call **Group SR** the group with all participants from group SR3 and SRH.

- **Group NT3** is the group that does not use SearchTrails in Phase 1 (N) and receives a search trail as an artifact for phase 2 (T). The group works on the topic '3D printing' (3) in phase 1 and 'Home automation' in phase 2.
- **Group NTH** is the group that does not use SearchTrails (N) in and receives a search trail as an artifact for phase 2 (T). The group works on the topic 'Home automation' (H) in phase 1 and '3D printing' in phase 2.
    - We call **Group NT** the group with all participants from group NT3 and NTH.

As the topics are designed to not influence the study results (details on this in Chapter 5.2.5.2), the groups SR and NT are referenced mostly throughout the text. The participants of each group are given a complex search task and they should search individually to get an overview of the given topic and to find enough material to prepare a thorough report on the topic.

The first phase of the study was planned to take 6 days, with one day of extra time for compensating the tardiness of some participants to hand in their reports. No time limits for the search or minimum page numbers for the reports were given. The first phase closed with an intermediate questionnaire on the experiences made during the first phase, the search strategies used, and the personal opinion towards the features of SearchTrails. During that phase, search trails were created by the participants during their search process. They were stored on a remote server with the participants not having access to them. Similarly, the reports were collected via e-mail, and the intermediate questionnaires were collected after the evaluation session. The evaluation session took place with all participants being in the same room to allow questions on the evaluation items. As the participants time management during the study was not monitored, the participants were not required to perform the search tasks in several sessions.

After the first phase of the study and just before the second phase started, I evaluated all search trails and all reports from the first phase together with Ulrich Norbisrath. We selected one anonymized report and the according anonymized search trail from one user of each topic to serve as the input for the second phase of the study. We selected the report and the search trail from participant number 01 for the topic 'Home automation' and from user 14 for the topic of '3D printing'. The selected reports and trails come from the group that worked with SearchTrails during the first phase, and the artifacts were chosen because they contained enough information to get a good overview of the topic and did not contain information on the extended search tasks '3D_2' and 'HA_2'.

For the second phase, the participants from group NT received a fully functional version of SearchTrails via e-mail, which replaced the previously installed version of SearchTrails. The topics were exchanged between the groups, such that group SR3 and NT3 (which worked on the topic 3D printing in the first phase) now worked on the topic from group SRH and NTH from the first phase, which was home automation.

Group SR3 and SRH, which worked with the full feature set of SearchTrails in the first phase were given a scanned version of a report from the first phase. The scanned version disabled copying and pasting excerpts from the given report easily into the participants' own reports. Group NT3 and NTH did not work with SearchTrails during the first phase of the user study. We equipped these participants with a complete search trail to start from. We suppose that this was a similar situation for them like being given search results, links, or artifacts from colleagues during collaborative search. The given material was ensured to contain enough information to serve as a good basis, but lacking information on the additional topics for the second phase. The second phase again lasted 6 days, with one additional day for students who did not submit in time. It concluded with a final questionnaire session. The search trails created during this phase were also stored on a remote server and the reports were again collected via e-mail. The final evaluation session was conducted in class and consisted of the final evaluation questionnaires and a user experience questionnaire (Laugwitz et al., 2008). Both paper-based questionnaires were collected after the evaluation session.

During the second phase of the study, the focus changed to the value of a search trail as an artifact for collaboration. We tested whether search trails can be valuable assets for the exchange of search results, as they are supposed to save redundant searches and provide a quicker start into a research than a report could do. For this phase of the study, the search tasks got extended by specifying one aspect of the topic to be researched deeper. While the first search task was intended to cover a broad topic with its many single aspects, the second phases' extended search tasks focused on one specific aspect of the broader topic, needing information from the first phase as a basis to start from. Additionally, the participants were asked to state their own opinion one aspect of the topic, which required them to gather enough material to be able to formulate an opinion. Search task '3D_1' on 3D printing was extended by the task of finding existing and future applications of 3D printing in the car manufacturing domain. Search task 'HA_1' on home automation was extended by finding existing and future applications with regards to home security. Again, like in the first phase, no statements about the minimal efforts or report lengths were made. The extended complex search tasks for the second phase of the study were as follows:

- **Search task '3D_2': Topic '3D printing'**
  'Based on the given material from the first Phase, find applications of 3D-Printing in the car manufacturing domain. Which applications exist, which ones will come? Will 3D-Printing change the way of manufacturing cars in the future?'
- **Search task 'HA_2': Topic 'Home automation'**
  'Based on the given material from the first Phase, find applications of home automation dealing with home security. Which applications exist, which ones will be available? Which applications would you prefer?'

Table 22 summarizes the chronology of the study design and gives an overview of the given topics and evaluation artifacts. The first column shows the days in which the actions from the second column were performed. The remaining columns show the group-specific differences.

This helps keeping an overview when the events happened and how they affected the different groups of participants.

Table 22: Overview of the study tasks and artifacts for each of the four user groups from the second user study.

	Day	Action	Group SR3	Group SRH	Group NT3	Group NTH
First phase	0	1st evaluation session	Same for all participants			
	0 – 6	Topic	**3D_1** ‘3D printing’	**HA_1** ‘Home automation’	**3D_1** ‘3D printing’	**HA_1** ‘Home automation’
	0 – 6	Used SearchTrails	Yes	Yes	Only for logging	Only for logging
	0 – 6	Collection of search trails in the online repository.				
	6	Deadline for submitting the reports. Sent e-mail to tardy students.				
	7	2nd evaluation session	Including SearchTrails specific questions		No SearchTrails specific questions	
Second phase	7 – 13	Topic	**HA_2** Home automation	**3D_2** 3D printing	**HA_2** Home automation	**3D_2** 3D printing
	7 – 13	Type of artifact	Report	Report	Search trail	Search trail
	7 – 13	Used SearchTrails	Yes	Yes	Yes	Yes
	7 – 13	Collection of search trails in the online repository.				
	13	Deadline for submitting the reports. Sent e-mail to tardy students.				
	14	3rd evaluation session	Specific questions on evaluating the report + UEQ		Specific questions on evaluating the search trail + UEQ	

5.2.3 Implementation of the study design

The second user study aimed towards potential users of the system, which are users with experience in web search and information processing, doing several complex search tasks per week. Therefore, students from a lecture on Computer Supported Collaborative Work (CSCW) were selected. As this lecture covers a wide range of web technologies and collaborative systems, the participants can be assumed to be potential future users of the system. Overall, 29 students attended the lecture and participated in the study.

The study was presented to the participants as the final part of one lecture in the second third of the semester. It featured a short presentation with a motivation of SearchTrails and an offline SearchTrails demonstration. All participants were informed about the data captured and about the option of having nodes or even complete trails deleted in case they were not comfortable with the stored data. Every participant signed a consent form to participate in the study, agreeing that web browsing data is captured and analyzed within the study. In the end, no participant made use of the option of withdrawing data.

The study covered a timespan of two weeks, starting on a Wednesday, having the intermediate evaluation on the following Wednesday and being closed with the final evaluation on the third Wednesday. The deadline for handing in the reports was set to Tuesday noon in each week. Each received search trail was answered by a personalized ‘Thank you’ e-mail. On Tuesday afternoon, all participants that did not send a report by then were reminded about the deadline having passed and were encouraged to hand in the report until Wednesday noon, just before the lecture started.

To split up participants into groups, all study materials were numbered (from 1 to 29) and assigned a letter according to the group (Letter A / Group SR3: Participants 2, 6, 10, 14, 18, 22, 26; Letter B / Group SRH: Participants 1, 5, 9, 13, 17, 21, 25, 29; Letter C / Group NT3: Participants 4, 8, 12, 16, 20, 24, 28; Letter D / Group NTH: Participants 3, 7, 11, 15, 19, 23, 27). The study materials were handed out in the lecture hall consecutively from number 1 to 29, and from front to back and to front again. This method ensured an even distribution of students to the groups, avoiding some effects: participants sitting next to each other did not end up in the same group to avoid plagiarism. Also, this method prevented synergies originating from study groups of befriended students usually sitting close to each other. Similarly, students from all rows of the lecture hall were distributed equally among the groups, to avoid clustering students from similar rows into groups and therefore biasing the results.

The instruction sheet was handed out to the participants and should be taken home. It was handed out together with the initial questionnaire, which was filled out in class and was collected by me. The IDs of the participants were matched to the e-mail addresses on a separate sheet of paper, which was destroyed after the study. This was necessary, as different versions of SearchTrails and different materials had to be sent to the participants during the study. However, some participants actually did not really seem to care about these privacy means, as four of 27 participants put both clear name and username onto the reports during the first phase. During the second phase, two of 21 participants put both their name and their user ID on the reports.

In the early afternoon of the starting day of the study, the installation files of SearchTrails were sent to the participants via e-mail, together with detailed instructions for the installation in the Chrome browser on different operating systems. For the next days, the participants were not influenced by any means.

During both weeks, the participants’ reports were collected via e-mail. Figure 85 shows the timely distribution of the incoming reports (large symbols) during the first week of the study. The numbers show important points in time during the evaluation: ‘1’ marks the start of the study, ‘2’ marks the official deadline on Tuesday noon until the reports should be handed in. After this deadline, the reminder e-mail was sent. ‘3’ marks the end of the first phase and the start of the intermediate evaluation session in class. The small dots resemble points in time when log files have been stored on the server. One can see that the search activity started just after the study began and intensified as time got late.

A large share (19 of 27) of the reports came in until Tuesday noon, Jan, 20. On Tuesday afternoon, all participants who did not yet send a report were encouraged via e-mail to perform the search and hand in the report. The additional reminder for the students being late lead to eight more reports coming in until the lecture started on January 21, 2015, at 11:45.

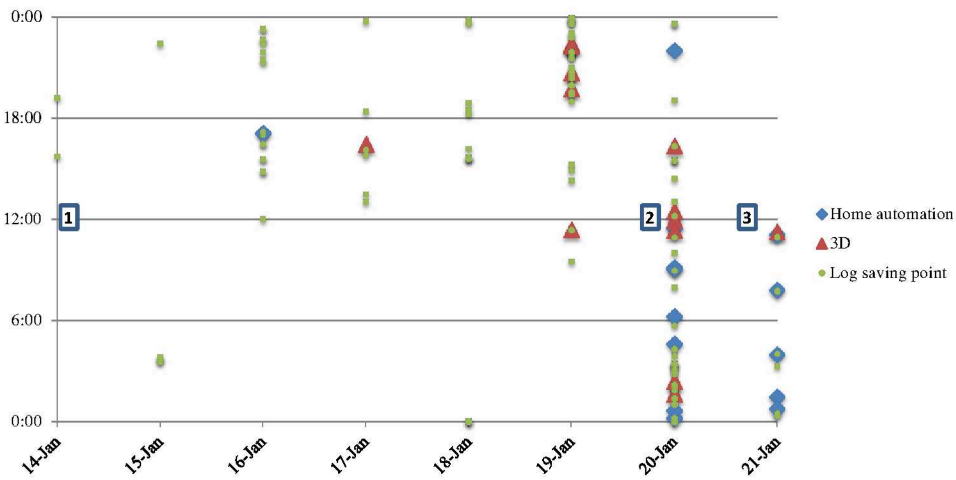


Figure 85: Diagram showing the return times of the study artifacts during the first phase of the second user study.

During the second evaluation session in class, the participants were asked if there were larger technical problems, but except for some minor bugs, SearchTrails seemed to have run very stable. During this session, the intermediate questionnaires were filled out and the instructions for the second phase of the study were handed out. After the classroom session, all generated search trails on the storage server were moved to a specific section, such that the search trails could not be changed anymore by the participants.

On Wednesday afternoon, I equipped the participants with all material necessary for the second phase of the study. For each participant of group NT (who should receive a search trail as input for the second phase), I set up a copy of the selected search trail and sent the respective individual ID to the participant. I sent each participant of group SR (who should receive a report as input for the second phase) the scanned version of the report. I sent all participants a new version of SearchTrails. This new version both fixed minor bugs that became apparent during the first phase, and it enabled the visualization for the participants in group NT.

Figure 86 shows the timely distribution of the incoming reports during the second phase of the study (large symbols). The numbers show significant timestamps for the study: ‘1’ marks the start of the second phase, ‘2’ marks the official deadline on Tuesday noon after which the reminder e-mail was sent out, and ‘3’ marks the official end of the study and the begin of the final evaluation session in class. Before Tuesday noon, 17 of 21 students handed in their reports via e-mail. On the early afternoon of Tuesday (Jan. 27), again an e-mail was sent out to the tardy students, which led to four additional reports being handed in until the start of the lecture on Wednesday. The last lecture session was mainly about filling out the final questionnaires. The small green dots again show the points in time when logs were stored on the server. In opposition to the first phase of the study, less log storage events were produced, and the participants started working much later.

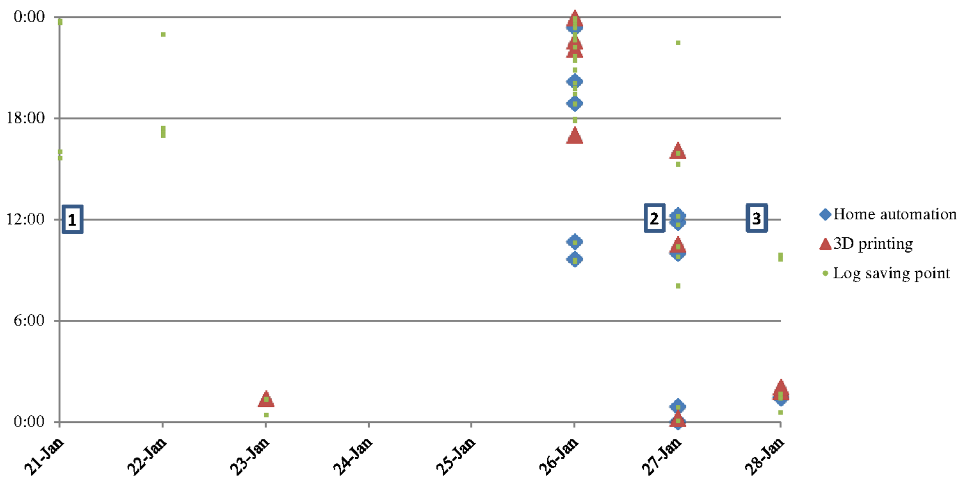


Figure 86: Diagram showing the return times of the study artifacts during the second phase of the second user study.

Table 23 shows all artifacts returned by the participants during the evaluation. 29 participants attended the first questionnaire session. For Participant 10, no search trail was recorded due to a technical difficulty (cf. Chapter 5.2.3 'Experiences during the study'). Participants 15 and 16 did not work during the first search phase, such that there are no trails and reports from them. The intermediate questionnaire session was joined by all participants, except participant 15. During the second phase, the motivation seems to have dropped. Participants 15 and 16 did not start working, and participants 7, 17, 20, 21, and 26 also created neither a search trail nor a report. Participant 23 did a search, but did not provide a report. In the final questionnaire session, more participants were back, but participant 11, 15, and 17 did not attend. The missing trails and reports do not necessarily mean that the participants did not work with SearchTrails. Even if the given search trail was not extended, the participants may have investigated the search trail and can provide valuable feedback in the final questionnaires, including the UEQ.

Table 23: Overview of all returned artifacts for each participant of the second user study.

Participant ID	Group	P1	Support	P2	Artifact	Questionnaire	Trail	Report	Questionnaire	Trail	Report	Questionnaires
1	SRH	Home	Yes	3D	Report	●	●	●	●	●	●	●
2	SR3	3D	Yes	Home	Report	●	●	●	●	●	●	●
3	NTH	Home	No	3D	Trail	●	●	●	●	●	●	●
4	NT3	3D	No	Home	Trail	●	●	●	●	●	●	●
5	SRH	Home	Yes	3D	Report	●	●	●	●	●	●	●
6	SR3	3D	Yes	Home	Report	●	●	●	●	●	●	●
7	NTH	Home	No	3D	Trail	●	●	●	●			●
8	NT3	3D	No	Home	Trail	●	●	●	●	●	●	●
9	SRH	Home	Yes	3D	Report	●	●	●	●	●	●	●
10	SR3	3D	Yes	Home	Report	●		●	●	●	●	●

11	NTH	Home	No	3D	Trail	●	●	●	●	●	●	
12	NT3	3D	No	Home	Trail	●	●	●	●	●	●	●
13	SRH	Home	Yes	3D	Report	●	●	●	●	●	●	●
14	SR3	3D	Yes	Home	Report	●	●	●	●	●	●	●
15	NTH	Home	No	3D	Trail	●						
16	NT3	3D	No	Home	Trail	●			●			●
17	SRH	Home	Yes	3D	Report	●	●	●	●			
18	SR3	3D	Yes	Home	Report	●	●	●	●	●	●	●
19	NTH	Home	No	3D	Trail	●	●	●	●	●	●	●
20	NT3	3D	No	Home	Trail	●	●	●	●			●
21	SRH	Home	Yes	3D	Report	●	●	●	●			●
22	SR3	3D	Yes	Home	Report	●	●	●	●	●	●	●
23	NTH	Home	No	3D	Trail	●	●	●	●	●		●
24	NT3	3D	No	Home	Trail	●	●	●	●	●	●	●
25	SRH	Home	Yes	3D	Report	●	●	●	●	●	●	●
26	SR3	3D	Yes	Home	Report	●	●	●	●			●
27	NTH	Home	No	3D	Trail	●	●	●	●	●	●	●
28	NT3	3D	No	Home	Trail	●	●	●	●	●	●	●
29	SRH	Home	Yes	3D	Report	●	●	●	●	●	●	●
Overall number of participants (n=)						29	26	27	28	26	21	26

After the study was performed, all search trails generated in the second phase were moved to a secure section of the storage server again. The reports were printed out and were graded by me, Wolfgang Prinz, and one of my colleagues independently. No evaluator knew the grades of the other evaluators until the grading was finished. All three evaluators have experience with the grading of student work and used a German standard university grading scheme. This grading scheme starts with 1.0 as the optimal grade and declines by steps of approximately 0.3 reaching 4.0. A grade of 5.0 means 'failed'. This results in a scale of: 1.0, 1.3, 1.7, 2.0, 2.3, 2.7, 3.0, 3.3, 3.7, 4.0, 5.0. After the evaluation of the reports, also the search trails were evaluated by a self-written python script, extracting statistical data from the search trails. Additionally, we used the CSCW lecture exam, which was written one week after the final evaluation, to pose one free-text question to all study participants, which was awarded with bonus points. We asked the participants to write down as many features of SearchTrails as they could remember to find out which features were most important for their work with SearchTrails.

Experiences during the study

I expected the study itself to be a technically critical part of the overall work on this thesis. Although SearchTrails has been tested thoroughly during its implementation, it has never been tested with such a number of users at the same time, requesting the server so frequently. During each of the two phases of the study, lots of search trails were monitored while they were emerging as small files on the server and growing during the regular intervals of saving. A bit surprisingly, only two minor problems occurred during the time of the study, which both could be resolved before the second phase of the study began. The first of the problems was the constant recreation of the search trail, which lead to some sort of flashing behavior in SearchTrails. The second problem was the very rare overwriting of search trails on the storage server in the case of synchronous access of it. This was the reason for participant 10 not having a search trail in the first phase. I fixed both problems for the second version of SearchTrails.

The chosen study setting did include a substantial risk of failure, as the diversity of operating systems, users, networks, and languages could not be tested before. This made me worry whether the study would run smoothly, or if problems would arise that could invalidate the results of the study. As the danger of e.g. a server crash was apparent during the whole study, I was very nervous for the two weeks the study ran, and had to constantly check e-mails to provide help when necessary. Luckily, no server problems happened, and only a few students asked about things that were already explained in the study material handed out.

Figure 85 and Figure 86 show the timely distribution of the reports being handed in via e-mail (cf. the large symbols). During the first phase, 27 reports were handed in, while during the second phase 21 reports arrived. The small green dots show the search trails' saving points. These are the timestamps when the participants' search trails were touched for the last time. During the first phase, 134 search trails were created, while in the second phase 59 search trails were created. Due to the technical implementation of SearchTrails, a new search trail gets created and stored upon each new start of SearchTrails. The recreation of search trail triggered the extension of an existing search trail. The search trails are of various sizes, some being just the initial data, some are much larger. Search trails with just the initial data show that SearchTrails has been turned on, but has been closed before the first storage interval, which was five minutes. For the evaluation of the search trails, the most recent and largest search trail for each participant has been taken.

During the first phase of the study, the search processes and eventually also the work on the reports was distributed more evenly during the week than in the second phase of the study. However, the main work for searching and writing the report concentrated on the two days before the deadline, and was mostly done at night. For the first phase, 92 of 134 search trails (68.66%) were created between 18:00 and 06:00, and 72 of 134 search trails (53.73%) even between 21:00 and 04:00 in the morning. This behavior can also be found during the second phase, where 35 of 59 search trails (59.32%) were created between 18:00 and 06:00, and 28 of 59 (47.46%) search trails were even created between 21:00 and 04:00. This can be argued to be a normal work behavior for university students, whose focus it is to prepare the requested input just in time for the deadline, but may not be representative for office workers like in the first study.

5.2.4 Questionnaire development and statistical prerequisites

While the results of the first user study were qualitative, the second study aimed for more quantitative results. As the total number of 29 participants was split into at least two groups for each analysis, statistically significant results could not be generated for all hypotheses. With a total number of 29 participants, the size of the sample can be considered small (Eid et al., 2010, p. 357), making statistically significant results still hard to achieve.

Questionnaire development

The data collected from the participants consists of three types of artifacts: The search trail data, which was created and stored automatically by SearchTrails, the reports the participants handed in via e-mail, and the questionnaires being filled out during the study. The search trails were stored as JSON objects on the BSCW storage server. The reports were handed in as text documents via e-mail by the participants. The paper-based questionnaires were handed out during the evaluation sessions and were digitized manually afterwards.

The questions were designed as open and closed questions. Most open questions asked for selected tools and the participants' approach for solving complex search problems. Most closed questions were designed as 5-point Likert questions, asking for the level of agreement to the stated hypotheses. The studies' questions were reviewed by me, Ulrich Norbisrath, Wolfgang Prinz, and Dirk Lewandowski during the design of the questionnaire.

The questionnaires required intensive preparation. Every group of questions in the questionnaires is titled with the test criterion this group of questions aims for. In the questionnaires being handed out during the study, these titles were replaced with letters from A to Z and additional numbers, to not reveal this information to the participants. Please find the questionnaires used for the second user study in Appendix A2.

As the design of a questionnaire is sensible, all questions were checked against the criteria given in (Diekmann, 2014, p.440 ff.). These criteria span the setting of a questionnaire session, avoidance of the three main sources of errors, and the design of the questionnaire itself. Considering the setting of the questionnaire session, the willingness for cooperation, honest answers, and a common language should be ensured. Cooperation was achieved by implementing the study as a subsidiary part of a lecture, honest answers were ensured by explicitly telling the participants that there were no right or wrong answers, and a common language was ensured by phrasing the questions as simple as possible.

The three main sources of error for questionnaires are special characteristics of the participants, the questions, and the interview situation. Participant characteristics that could influence the results are social acceptance, the unwillingness to admit problems and a tendency to answer with 'Yes'. To avoid these effects, I ensured that the questions are not phrased to force social acceptance, that the questions do not force the participants to admit problems, and that some questions are phrased negatively.

Question characteristics that could influence the results of the study are hidden information within the questions, question row effects, in which the answers to earlier questions influence the answers on later questions, and estimation problems for temporal questions, which let the users forget about real timespans. To avoid these effects, we reviewed all questions and all questions on timespans used actual days to reference the timespan (e.g. 'since last Wednesday' instead 'during last week'). Biasing effects of the interview situation on the study results are less relevant for questionnaire evaluations, but still it was ensured that all texts on the questionnaires were polite, the participants were thanked for their effort, and privacy was ensured, including the option to have search trails deleted. Despite all efforts, some effects could not be completely avoided. One example for the question row effect can be seen in Table 55 in Chapter 5.2.5.4, where the results for the two consecutive questions R3 and R4 correlate extremely high, while all other responses in the same question group do not correlate that strong.

The second evaluation of SearchTrails returned a number of different artifacts. Table 24 shows the number of returned artifacts from the participants during the course of the second user study. All statistical results are based on the numbers of artifacts in this table.

Table 24: Overview of the numbers of returned artifacts during the second user study.

	Group SR3	Group SRH	Group NT3	Group NTH	Total n
1st evaluation	7	8	7	7	29
Topic	**3D_1** 3D printing	**HA_1** Home automation	**3D_1** 3D printing	**HA_1** Home automation	
Total search trails	134 among all user groups				134
Final search trails	7	8	6	6	27
Report	7	8	6	6	27
2nd evaluation	7	8	7	6	28
Topic	**HA_2** Home automation	**3D_2** 3D printing	**HA_2** Home automation	**3D_2** 3D printing	
Type of given artefact	Report	Report	Search trail	Search trail	
Total search trails	59 among all user groups				59
Final search trail	6	6	7	7	26
Report	6	6	5	4	21
3rd evaluation	7	7	7	5	26

Statistical prerequisites: Comparing groups of samples

In all analyses, the samples from two user groups are compared. This allows a maximum number of participants per group for comparison. In most of the cases, the average value of answers to Likert-scaled questions is analyzed. There is an ongoing discussion whether Likert-scaled data can be interpreted as being interval-scaled (Diekmann, 2014, p.292 ff.), which is a prerequisite for analyzing average values. Most authors argue that this is the case and the resulting values are interpretable (Diekmann, 2014, p.297 ff.), which allows the following analyses.

For comparing two groups of results, significance tests need to be applied to find out whether the visible differences of the samples occurred at random or point towards a larger trend. Depending on the characteristics of the samples and the type of the chosen analysis, different statistical methods need to be applied. The first distinction concerning the choice of an applicable significance test is made with respect to independent and paired samples. For independent samples, there does not exist a dependency between the measured values. For paired samples, there is a dependency between the samples, which may be caused by measuring the results of the same user group before and after an event. The second distinction is made with respect to the scales of the available data, allowing parametric or non-parametric methods. The main difference between parametric and non-parametric methods is that the parametric methods may be applied to interval-scaled data that is following a normal-distribution, while non-parametric methods do not request a specific distribution of data. For analyzing the results of the second study, parametric methods can be chosen, as the data can be considered interval scaled and normal distributed.

In case of independent samples, the 'Independent samples T-test' is a parametric method, while the 'Mann-Whitney U-test' is a non-parametric method. This test is also called 'Wilcoxon rank-sum test', with which it is algebraically identical (Eid et al., 2013, p.322). In case of paired samples, the ‘Paired-samples T-test’ is a parametric method, while the ‘Wilcoxon signed-rank test’ is a non-parametric method (cf. Table 25).

Table 25: Overview of possible statistical methods for evaluating the results of the second user study.

	Independent samples	Paired samples
2 samples	*Parametric method:* Independent samples T-test (Heteroscedastic or homoscedastic)	*Parametric method:* Paired samples T-test
	Non-parametric method: Mann-Whitney U-test (also called: Wilcoxon rank-sum test)	*Non-parametric method:* Wilcoxon signed-rank test

The different tests cannot be applied to any sample directly. Some prerequisites have to be fulfilled to ensure that the results are statistically meaningful. For parametric methods, the following criteria have to be fulfilled:

1. Interval-scaled data
2. Normal-distributed values
3. Homogeneous variances

As the SearchTrails questionnaires were designed using 5-point-Likert scales, we consider the first criterion fulfilled, according to the discussion in (Diekmann, 2014, p.292 ff.). The second criterion has to be tested with a test on normal distribution. This can be done by matching the distribution of results to a normal distribution by sight, or by a Chi-squared-test, which tests samples on discrete scales for normal distribution. As long as there is no statistically significant result of the Chi-squared-test, a normal distribution can be assumed. For testing the similarity of variances of two samples, the F-test has to be used (Eid et al., 2013, p.389 ff.). This test both returns an F-value for the given data and a critical F-value. The critical F-value needs to be exceeded by the F-value to confirm the hypothesis that the variances of the two samples are identical. Depending on its result, the independent samples T-test for homogeneous variances (homoscedastic T-test) or the independent samples T-test for heterogeneous variances (heteroscedastic T-test) has to be used. The heteroscedastic T-test differs from the homoscedastic T-test in that it incorporates the so-called Welch-correction, which eliminates the influence of unequal variances in a T-test (Eid et al., 2013, p.311), which can then be called Welch’s T-test. In the following, T-test references the applicable T-test for the samples under evaluation.

For non-parametric tests, the aforementioned three criteria can be reduced to the existence of interval-scaled data. If the prerequisites for the parametric tests are not fulfilled, the assumption that these tests deliver reliable analyses is not valid. In these cases, non-parametric methods need to be used (Eid et al., 2013, p.310). If all prerequisites for a parametric test are fulfilled, these tests possess higher test strengths than the non-parametric ones. The T-test returns both a T-value and a critical T-value. When the T-value is not within the interval of the negative critical T-value to the positive T-value, the average values of the samples are significantly different.

Investigating linear relations: Correlation

For investigating a linear relation between two groups of samples, a measure of correlation has to be used. Depending on the type of the given data and the prerequisites fulfilled, there are several methods for investigating the correlation between two samples. When both correlated samples are scaled on a metric scale, the Pearson product-moment correlation coefficient can be used. If one of the samples is scaled ordinal, Kendall's tau coefficient can be used (Eid et al., 2013, p.539).

For applying the Pearson product-moment correlation coefficient, several prerequisites have to be fulfilled. First, the given values have to be on a metric scale. This is fulfilled for interval and ratio scales. Furthermore, the data needs to be roughly normal distributed, and a linear relation needs to be ensured. For applying Kendall's tau, the given values only have to be on an ordinal scale. Furthermore, the data needs to be roughly normal distributed, and a linear relation needs to be ensured. For ordinal scales, Spearman's rank correlation coefficient can be used, but is not very recommended, as it is very sensitive to outliers (Eid et al., 2013, p.520). I therefore choose the Pearson correlation coefficient when both samples are metric scaled (e.g. Likert-scaled questionnaire results), and use Kendall's tau to be more exact when both samples are ordinal scaled (e.g. university grades). When a correlation between ordinal and metric scaled samples should be calculated, I again use the Pearson correlation coefficient. I consider the university grades to be well interpretable when for example average values are calculated, such that a Pearson correlation is not problematic in these cases.

Both correlation measures return a value between -1 and +1. For -1 and +1, a perfect negative or positive linear correlation is found. This means, that all values lie on a perfectly straight line in a two-dimensional space. Be aware that a perfect value like -1 or +1 does not explain the steepness of the line, but just the existence of a linear relation. A correlation coefficient of zero means that no linear relation between the given values exists. This can be because the values are completely unrelated, or because the relation is not linear, but e.g. follows a sinus curve. All values between 0 and -1, respectively 0 and +1, describe a weaker or stronger correlation, where a value of 0.1 can be considered a weak correlation, a value of 0.3 can be considered a medium correlation and a value of 0.5 can be considered a strong correlation (Eid et al., 2013, p.508).

As a full description of all statistical tests on preconditions would exceed the scope of this thesis, the described statistical results are limited to the most important findings. For each analysis in the following sections, it was ensured that all prerequisites are fulfilled by performing the different applicable tests. Whenever a confidence interval is referred to, an alpha coefficient of 0.05 is used, meaning an error probability of 5%. Similarly, for correlations a significance level of 5% can be assumed unless stated otherwise.

5.2.5 Study results

As can be seen in Table 24 in Chapter 5.2.4, the second study returned a number of artifacts from all different phases. In order to keep an overview of the relevant results, the following part of this chapter is organized as follows: I first describe the study setting with respect to the distribution of participants among the groups. I then examine the possible influence of the topics of the study results. After that, I confirm the five main hypotheses. The following section presents a set of additional findings made during the study. The last section elaborates on limitations of the study and summarizes the results.

Several of the results generated in this section help answering the main research questions from question group RQ2 'Value' of my thesis, as developed in Chapter 1.4.

5.2.5.1 Study setting and prerequisites

The results described in this section are gathered by the evaluation of the first questionnaire, which was handed out and filled out before the evaluation started. It was filled out by all 29 participants.

Before the study results can be evaluated, it needs to be assured that the selected participants do have a need for tool support during complex search tasks and that there are no statistically significant differences between the user groups. In order to achieve this, the preliminary questionnaire was designed. This questionnaire aimed for demographic information, as well as the experience with web search in general, and the experience with complex search tasks.

The demographic analyses in Table 26 show that the participants are on average 23.7 years of age, and the participant set included 51.7% female and 48.3% male participants (question A2). The participants were asked how often they search on the Internet per day and within the last week, and if they used the Internet yesterday for searching something. Every participant used the Internet the day before (A4). The average value for the number of searches per day is 18.97, with a confidence interval of 4.95 (A5), and for the number of searches per week it is 90.15, with a confidence interval of 28.29 (A6). This means that the true value of the number of searches per day is between 14 and 24 searches per day with a probability of 95%, and the true value for the number of searches per week is between 61 and 119. When being asked for the level of experience with web search on a scale from 1 to 5 (B1), where 1 means 'professional' and 5 means 'newbie', the average value for all participants is 1.79 (with a small confidence interval of 0.2, resulting in a range from [1.59, 1.99]). 93% of the participants rate their level of experience with web search as 'professional' or 'better than average'.

Table 26: Overview of the demographic results of the second user study, split by gender.

	Female	**Male**
(A2) Gender of participants	15 (51.7%)	14 (48.3%)
(A1) Average age of participants	24.1 years	23.4 years
(B1) Average level of experience with web search	2.00	1.57
(A4) Number of participants that used the Internet the day before	15 (100%)	14 (100%)
(A5) Number of searches per day	Mean: 18.97, Confidence interval: 4.95	
(A6) Number of searches per week	Mean: 90.15, Confidence interval: 28.29	

When being asked how often the participants faced a complex/unspecific/tedious search task within the last week (B5), 89.7% of the participants provided a number, implying that they performed such a search task. On average, 6.86 (in a confidence interval of [4.54, 9.18]) complex search tasks were performed per week by the participants. Each complex search task was estimated to consume 22.8 minutes (B10, [16.06, 29.54]). The first impression is that this timespan is rather short. As we see later, most of the participants needed more time to solve the given search tasks. As the confidence interval is calculated on a 5% error probability, it can be assumed that with a 95% probability the real value is within that interval. This implies that a complex search task takes most of the participants normally no longer than 30 minutes.

I also asked which tools were used to cope with the amount information during complex search tasks and received 28 answers (B6). Participants used on average almost 3 (exactly 2.93) different tools to cope with complex search tasks ([2.42, 3.44]). 92.9% (26 of 28) of the participants used additional tools to support their complex searches apart from the standard web browser functions (e.g. the back-button or the history). The most commonly used tools are bookmarks (57.1%, 16 of 28), text editing systems (39.3%, 11 of 28), mobile devices (35.7%, 10 of 28), and taking handwritten notes (25.5%, 7 of 28). Browser extensions or plug-ins are used by 10.7% of the participants (3 of 28), while 7.1% use social bookmarks (2 of 28). The tools used are Evernote[80] and pocket[81], two web-clipping tools that allow storing text and image fragments persistently. This shows that the participants use a large set of tools to cope with the amount of information during complex search. It can be deduced from these results that the support by search engines during complex search tasks is insufficient, as there is a need for using so many additional tools. This confirms the findings in (Singer et al., 2012a) on the insufficient support of search engines during complex search.

These results seem to contradict the participants' subjective feeling of support by search engines during complex search tasks. Participants seem to be satisfied with the support by search engines during complex search tasks (B12), as 82.8% (24 of 29) state that they feel well-supported by search engines in the questionnaire. This is comparable to the results of the first study (cf. hypothesis H1.1), where participants were also satisfied with the support by search engines at first sight, but closer examination revealed that participants did not expect search engines to support complex search tasks. These findings confirm the impression that users are easily satisfied by the results of web search engines (Lewandowski, 2015, p.276). The participants of the second study used additional tools that provide missing functions for complex search support. Insofar, the participants expect search engines mainly to produce lists of results, and do not perceive them as a tool for keeping contexts or capturing result highlights.

Adding to the previous findings, 55.2% (16 of 29) stated that they wanted to go back to older searches within the last week (B13), and 48.3% (14 of 29) wanted to share search results with colleagues (B14). To achieve this, several approaches are used. The most common approach is the sharing of links, may it be via Facebook, Skype, Google Docs, Hangouts, e-mail, or other tools. Two participants even reported about phone calls, in which exact directions are given on which queries to enter where and which links to follow. 34.5% wanted to share a complete set of search results on a specific topic with someone (B17). This was mostly desired during university team work, to prevent team members from redundant searching or for didactic reasons to show other team members where results can be found: *'When I'm in group work and*

---

[80] Retrieved April 10, 2015 from https://chrome.google.com/webstore/detail/evernote-eb/lbfehkoinhhcknnbdgnnmjhiladcgbol?hl=de

[81] Retrieved April 10, 2015 from https://chrome.google.com/webstore/detail/pocket/mjcnijlhddpbdemagnpefmlkjdagkogk?hl=de

it's not easy to find the resource we need. Thus I would like other partners to review my results to prevent visiting those websites again […]' (participant answer to question B18).

During the first phase of the study, where group SR used SearchTrails, but group NT did not, participants were asked how long they expected the search task to take. The average value for all participants was 47.21 minutes (C1, [37.13, 57.29]). The average value for this question is higher than for the question asking for the normal duration of complex search tasks. This may be due to the fact that participants may have felt to be in some sort of test situation, and were asked to write a report, which may have increased the estimations. For this question – as well as for all other statements from the first questionnaire – no statistically significant differences between the participant groups SR and NT can be found.

The results from the first questionnaire show that the participants are spread equally among the different groups, and that some results are already comparable to the results from the first study, such as the satisfaction with search engines. Also the use of search supporting tools confirms the results from the first study. The results on revisiting former search results and the exchange of search results can confirm that there is a need for the feature set of SearchTrails.

5.2.5.2 Possible influence of the search tasks on the study results

During the second user study, we assigned the participants one of two different complex search tasks, namely '3D printing' and 'Home automation' (cf. Chapter 5.2.2). After the first phase of the study, the search tasks were exchanged between the participant groups and extended by an additional question (cf. Chapter 5.2.2). For the following analyses, one could argue that the selection of search tasks or the assignment of participants to the search tasks may have influenced the study results, which would not be desirable. During the planning of the study, we tried to avoid an influence of the search tasks on the study results by checking each search task against the seven characteristics by (Kules & Capra, 2008, 2009), as shown above. The evaluation of the study artifacts can provide proof that this planning was successful, as the search tasks did not influence the search results.

First hints on that the results were not influenced by the selection of the search tasks can be found in Figure 85 and Figure 86, showing the timely distribution of the reports being handed in via e-mail. These diagrams show that there was no apparent inequality of the times the reports from the different groups were handed in. More proof that the search tasks had no influence on the study results can be taken from the quantitative analysis of the search trails, the grades of the reports, and the answers from the questionnaires.

In order to investigate the influence of the search tasks on the search trails, I separate the search trails by the search tasks (cf. Table 27). This analysis reveals that the median values of the search trail characteristics are very similar among the groups. The average values show a little larger difference, which is caused by one participant that can be considered an outlier. This participant produced a very large search trail, which influenced the average values. The only larger difference can be found in the median values of the number of highlights. These values were also influenced by a few users that made frequent use of the highlighting function, while other users may not have used this function during the first phase, maybe because they forgot about it. It is important to note that in the first phase of the study, half of the participants (the ones in group NT3 and NTH) did not actively use SearchTrails, such that they were not able to set highlights. The values for the numbers of highlights shown here are therefore based on just half of the test population. Statistical analyses have revealed that no statistically significant differences of the search trail characteristics can be found. This also holds true for the numbers of events, of nodes, of edges, of hosts, of highlights, of keywords, of sessions, or the duration of the searches.

Table 27: Statistical data of all participants' search trails from the first phase of the second user study, split by search tasks.

	Phase 1: Search task	Number of events	Number of nodes	Number of edges	Number of hosts	Number of highlights	Number of keywords	Number of sessions	Duration in seconds
Median	3D_1	209.0	25.5	52.0	12.5	13.0	118.5	1.0	2743
	HA_1	206.0	26.0	50.0	12.0	0.0	108.0	1.0	3166
Average	3D_1	252.0	27.9	56.0	14.9	13.5	138.5	1.7	3704
	HA_1	323.2	36.7	74.1	15.4	2.4	146.0	1.4	4061

I found similar results for the second phase. Also in this phase, the average and median values for the search trail characteristics are very close to each other (cf. Table 28). Similar to the first phase, I found no statistically significant differences between the values.

Table 28: Statistical data of all participants' search trails from the second phase of the second user study, split by search tasks.

	Phase 2: Search task	Number of events	Number of nodes	Number of edges	Number of hosts	Number of highlights	Number of keywords	Number of sessions	Duration in seconds
Median	3D_2	308.0	39.0	67.0	19.0	3.0	163.0	4.0	7557
	HA_2	258.0	35.0	62.0	19.0	8.0	138.0	2.0	7047
Average	3D_2	328.5	41.7	78.0	19.0	3.0	161.8	3.5	6756
	HA_2	265.2	36.8	72.9	17.2	6.4	129.7	2.2	5931

The grades of the reports confirm the previous findings. During the first phase, the average grades of the reports just show a difference of 0.19 (cf. Table 29). An F-test on the variances of the individual grades of all reports for each topic shows that there is no significant difference between the variances, as the F-value is smaller than the critical F-value. Thus, a T-test without the Welch-correction can be performed, which again shows that there is no significant difference between the average grades of the topics, because the T-value lies within the interval of the negative/positive critical T-value.

Table 29: Statistical results for the difference in report grades during the first phase of the second user study (not significant).

	Report grades, 1st phase	3D printing	Home automation
	Average grade	2.17	2.36
F-test	F-value	1.009	
	Critical F-value	1.694	
T-test	T-value	-1.098	
	Critical T-value	1.990	

The same results can be shown for the grades of the reports from the second phase of the study. The difference between the average values of the grades is 0.13, and both the F-test and the T-test show no statistically significant differences (cf. Table 30).

Table 30: Statistical results for the difference in report grades during the second phase of the second user study (not significant).

	Report grades, 2nd phase	3D printing	Home automation
	Average grade	2.71	2.84
F-test	F-value	1.159	
	Critical F-value	1.823	
T-test	T-value	-0.499	
	Critical T-value	1.999	

The answers from the questionnaires can provide more proof that the choice of the topic did not have an influence on the study results. As it is not feasible to present this analysis for every question, I select the self-reported experiences during the search tasks as an indicator for the influence of the topic on the study results. Questions on the estimated duration of a search and the opinion whether the topic was liked or found interesting can provide a high-level view on the participants' experiences.

Table 31: Statistical results for the difference in the duration of the search processes during the first phase of the second user study (Question D2, not significant).

D2	How long did you need for that search?	3D printing	Home automation
	Average duration	83.21	80.64
F-test	F-value	1.712	
	Critical F-value	2.577	
T-test	T-value	0.160	
	Critical T-value	2.056	

Table 32: Statistical results for the difference in the duration of the search processes during the second phase of the second user study (Question O2, not significant).

O2	How long did you need for that search?	3D printing	Home automation
	Average duration	77.92	64.29
F-test	F-value	2.896	
	Critical F-value	2.634	
T-test	T-value	1.029	
	Critical T-value	2.109	

Table 31 and Table 32 show the average values and the results for the F-test and the T-test for the question 'How long did you need for that search?' for both phases of the study (questions D2 and O2). These results show that there is no significant difference between the average duration of the search tasks in the first or the second phase with respect to the search topics. The same holds true for the statement 'I liked the search topic' for both phases of the study (questions E1 and P1), which the participants answered on a 5-point Likert scale. Table 33 and Table 34 show that the average value for this statement was close to 2.00 and that no statistically significant differences can be found between the topics.

Table 33: Statistical results for the difference in the popularity of the search topic during the first phase of the second user study (Question E1, not significant).

E1	I liked the search topic.	3D printing	Home automation
	Average value	1.93	2.07
F-test	F-value	3.049	
	Critical F-value	2.577	
T-test	T-value	-0.560	
	Critical T-value	2.080	

Table 34: Statistical results for the difference in the popularity of the search topic during the second phase of the second user study (Question P1, not significant).

P1	I liked the search topic.	3D printing	Home automation
	Average value	2.00	2.29
F-test	F-value	1.531	
	Critical F-value	2.761	
T-test	T-value	-0.867	
	Critical T-value	2.064	

The findings in this section show that no artifact that was produced during the second study can provide proof that the selection of the search topics had an influence on the study results or could influence the results for the respective topic statistically significant. No analysis even came close to being statistically significant. This can also be considered strong evidence for the quality of the search topics.

5.2.5.3 Confirming the hypotheses

The previous section presents necessary prerequisites for further analyses by evaluating the first questionnaire. The next sections do not follow this linear approach of evaluating the study artifacts as they were generated anymore. Some of the artifacts need to be seen in context, or differences between the two phases of the study need to be evaluated for confirming the hypotheses. Therefore, the following sections each confirm one of the hypotheses developed for the second user study. Chapter 5.2.5.4 presents additional findings from the second user study.

The tables in the following sections present the results of many selected questions from the questionnaires. Considering the high number of questions in all three questionnaires, I decided to present the results with the most meaningful results, leaving out the questions and results that cannot contribute to the overall results. This may be due to the reason that the interpretation of the results turned out to be not significant or because the questions were not understood correctly by the participants, which occurred once.

Hypothesis 2.1 – Easing single user complex search

The first hypothesis deals with the support of SearchTrails for single user complex search. Hypothesis H2.1 is: 'SearchTrails eases single user complex search tasks by supporting aggregation, synthesis, and discovery'. The first phase of the study can provide data to confirm or reject this hypothesis. During this phase, the test group SR used SearchTrails to support their complex search task, while the control group NT used their normal tools and just used SearchTrails for recording their search, without making use of SearchTrails' features.

However, showing that complex search is supported by a tool cannot be done as a whole, but rather by showing that all single elements of complex search are supported well. According to the definition of complex search, it consists of the subtasks of aggregation, discovery, and synthesis (Singer et al., 2012a). When all three of these tasks can be supported by SearchTrails, we can deduce that complex search is supported as a whole. This methodology is also proposed in (Singer et al., 2012a, p.101). In the questionnaires after both phases, a dedicated group of questions was asked to evaluate aggregation, discovery, and synthesis. As groups NT3 and NTH did not work with SearchTrails during the first phase, we just consider the results of group SR for the following table, except for statement H10, which was posed to all groups. Table 35 shows the statements as phrased during the intermediate evaluation, the number of answers (n), the average value, and the confidence interval on a significance level of 5%. The average value refers to the average value on a 5-point Likert scale, where 1 means the highest level of agreement ('Strongly agree') and 5 means the lowest level of agreement ('Strongly disagree'). Lower values therefore signalize agreement with the given statement.

Table 35: Statistical analysis of the questions on aggregation for the first phase of the second user study (Group SR, Question group H).

H	Aggregation evaluation	n	Average value	Confidence interval (α = 0.05)
3	The graph visualization behaved like expected.	16	2.86	0.65
4	The search graph obviously resembles the course of my search.	14	2.14	0.39
5	Seeing the course of my search was interesting.	15	1.60	0.31
6	It was interesting to see all visited pages on a glance.	15	1.60	0.36
7	It was interesting to see how I came to search results and where I went thereafter.	15	1.80	0.27
8	The search trail helped to see all pages where I found relevant information for my search.	15	2.20	0.46
9	The search graph visualization helps to remember relevant facts of the search topic.	13	1.77	0.38
10	At least once I wanted to get back to pages visited earlier.	29	1.71	0.29
11	SearchTrails helped me to get back to previous pages.	14	2.21	0.66

The level of agreement with the statements in Table 35 was collected after the first use of SearchTrails by group SR after the first phase of the study. The table shows that on average, the graph visualization behaved like expected (H3). The confidence interval is [2.21, 3.51], which means that the result is positive to neutral. We know from the evaluation of the first study that the graph visualization as a force-directed graph can be confusing when being seen for the first time. As such, this result can be considered positive for SearchTrails. This result is a prerequisite for the interaction of the users with the tool, as it implies that the participants are able to understand the visualization. The results for the next statements are much more positive: The visualization was understood to resemble the course of the search (H4, [1.75, 2.53]). Participants found it interesting to see both the course of the search (H5) and having all pages visible at a glance (H6). The average answers for both statements are 1.60, with a narrow confidence interval of 0.31 and 0.36, which shows very strong agreement. Adding to this, participants agree that it is interesting to see how they came to search results and where their search went thereafter (H7, [1.53, 2.07]), which is a type of information that can only be shown by SearchTrails. Agreement whether SearchTrails helped to see all pages where relevant information was found is a little weaker (H8, [1.74, 2.66]). This may be due to the fact that only very little highlights were set during the first phase of the search. This would have resulted in blue nodes in the graph which are easy to identify in the visualization. Despite the limited number of highlights, participants agree that the visualization helps to remember relevant facts (H9, [1.39, 2.15]). This gives a hint that the search trail is also an artifact from which higher level information is visible, for example which clusters were visited often. The statement if users want to go back to pages they visited earlier was posed to all participants and results in high agreement (H10, [1.42, 2.00]). This shows that complex search induces a high need for getting back to previously visited pages. SearchTrails helps participants doing so, although the results are a bit weaker (H11, [1.55, 2.87]) than for other questions in this group.

I posed the same questions again to both groups after the second phase of the study, where the positive results from group SR in the first phase can be confirmed (cf. Table 36). Especially the level of agreement on the statement if the visualization behaved like expected (U3) increased drastically, from 2.86 [2.21, 3.51] in the intermediate evaluation to 1.77 [1.39, 2.15] in the final evaluation for group SR. For group NT, the result is 2.44 [1.99, 2.89]. This shows that group SR got used to the visualization and learned how to interpret the search trail. In the second phase, group NT was confronted with the search trail for the first time, and had to learn how to interpret the visualization, resulting in weaker acceptance of the statement. The difference between the average values for Group SR and NT during the second phase is statistically significant on a 5% level.

Table 36: Statistical analysis of the questions on aggregation for the first phase of the second user study (Question group U); split by the respective user groups.

U	**Aggregation evaluation**	Group SR (Report)			Group NT (Search trail)		
		n	**Average value**	**Confidence interval (α = 0.05)**	**n**	**Average value**	**Confidence interval (α = 0.05)**
3	The graph visualization behaved like expected.	13	1.77	0.38	9	2.44	0.45
4	The search graph obviously resembles the course of my search.	14	2.07	0.42	11	2.55	0.39
5	Seeing the course of my search was interesting.	14	1.71	0.31	10	1.90	0.33
6	It was interesting to see all visited pages on a glance.	14	1.64	0.32	12	1.92	0.49
7	It was interesting to see how I came to search results and where I went thereafter.	14	1.71	0.31	12	2.00	0.33
8	The search trail helped to see all pages where I found relevant information for my search.	14	2.00	0.28	12	2.17	0.56
9	The search graph visualization helps to remember relevant facts of the search topic.	14	2.29	0.42	12	2.25	0.52
10	At least once I wanted to get back to pages visited earlier.	14	1.64	0.32	12	1.58	0.28
11	SearchTrails helped me to get back to previous pages.	14	1.93	0.46	12	1.92	0.54

Table 36 also shows that the acceptance of statements U4, U5, U6, U7, and U8 is higher for group SR than for group NT. This strongly suggests a learning effect during the second use of SearchTrails. Acceptance for U9, U10, and U11 is similar for both groups, and it is still positive. These results confirm that there is a need for going back to previous pages, and that SearchTrails can offer this kind of support. In general, the results for these statements on aggregation show that there is both a need for aggregation features and that SearchTrails is able to support these needs.

The second subtask of complex search is discovery. Table 37 compiles the central results from question group J from the intermediate evaluation. SearchTrails' main feature for supporting discovery is the list of keywords. The overall results are not as positive as for the aggregation, but still contain hints on the high value of the keywords for discovery. It is important to note that discovery cannot be measured directly, as it aims for knowledge gaps. As such, I could only try to measure discovery by the questionnaire. This could turn out to be problematic, as it relies much on the honesty of participants. Discovering knowledge gaps forces the participant to implicitly admit to not know important facts, which may influence the study results. Similarly, participants may forget very soon that a certain feature of SearchTrails induced the discovery of a piece of information.

Table 37: Statistical analysis of the questions on discovery for the first phase of the second user study (Question group J); split by the respective user groups.

J	Discovery evaluation	Group SR (Used SearchTrails)			Group NT (No use of SearchTrails)		
		n	**Average value**	**Confidence interval ($\alpha = 0.05$)**	**n**	**Average value**	**Confidence interval ($\alpha = 0.05$)**
1	I think I have found all relevant facts that can be found on my search task.	15	2.60	0.48	12	2.33	0.35
4	**Group SR:** SearchTrails enabled me to discover additional facts for my search task. **Group NT:** Taking notes during my search enabled me to discover additional facts for my search task.	13	3.31	0.49	12	2.42	0.43
6	The keywords column helped to see the context in which I was browsing.	15	2.33	0.75	Not applicable for group NT.		
7	The keywords helped to find relevant search terms that I would not have come up with.	14	2.93	0.67			

In general, participants in both groups are positive to neutral on the quality of their results (J1). Statement J4 aims for the perceived value of SearchTrails for fact finding. Participants from group SR are neutral to negative about the value of SearchTrails for finding facts on a topic (J4, [2.82, 3.80]). Participants from Group NT are statistically significant more positive on the value of taking notes for supporting fact finding (J4, [1.99, 2.85]). The findings for group SR seem not to speak for SearchTrails as a tool for supporting discovery at first sight, when being compared to the results of group NT. However, the positive results of group NT do not lower the value of SearchTrails for discovery. The results for Group NT speak more for the general value of taking any form of notes for reflection of the search process and organizing the results, which also worked with SearchTrails. Group SR may have to invest more efforts in discovering facts. Participants may possibly miss to integrate the keywords into their search processes, e.g. because they forgot to check the keywords. When we asked for the value of the keyword list itself, the results turned out to be more positive. Participants like to see the context of the search by the keywords (J6, [1.58, 3.08]). The actual support of the keyword list for finding new search terms is considered rather neutral (J7, [2.26, 3.60]). Remarks from the questionnaires show that the participants are aware of the keyword list, as some participants state that there were irrelevant keywords. Participants are probably so engaged in their search tasks that they do not feel a need to be pointed towards new search terms. However, several of the participants' comments in Chapter 5.2.5.4 (Section 'Qualitative results from the questionnaires') show a very positive opinion towards the keyword list.

After the second phase of the study I posed the same question to both groups again. Table 38 compares the results for groups SR and NT. We gave a report to group SR to start their searches from, and the participants had to do a new initial search on a new topic. As expected, the results for group SR confirm the results from the first phase, which shows a limited impact of the keywords for discovery (W4, group SR, [2.65, 3.49]). In contrast, we gave a search trail to group NT, which resulted in much better results than the ones from group SR in the first phase (W4, group NT, [2.32, 3.02], cf. Table 38), and in much better results than from group SR in the second phase. Unfortunately, the differences between the average values of both groups and phases are not statistically significant. This indicates that SearchTrails can provide valuable support for discovery when it comes to cases in which search trails are exchanged and the keywords can be evaluated to get an overview of the topic.

Table 38: Statistical analysis of the questions on discovery for the first phase of the second user study (Question group W); split by the respective user groups.

W	Discovery evaluation	Group SR (Report)			Group NT (Search trail)		
		n	**Average value**	**Confidence interval ($\alpha = 0.05$)**	**n**	**Average value**	**Confidence interval ($\alpha = 0.05$)**
1	I think I have found all relevant facts that can be found on my search task.	14	2.79	0.53	12	2.25	0.24
4	SearchTrails enabled me to discover additional facts for my search task.	14	3.07	0.42	12	2.67	0.35
6	The keywords column helped to see the context in which I was browsing.	14	2.36	0.47	12	1.92	0.49
7	The keywords helped to find relevant search terms that I would not have come up with.	13	2.85	0.47	12	2.25	0.52

Summing up, these findings indicate that the overall value of SearchTrails is more in aggregation and synthesis than in discovering new facts of a topic, but some features of SearchTrails can improve discovery during single user search tasks. Further usage of SearchTrails may lead to better results, as participants interact more or learn better to interpret the contents of the keyword list and start appreciating its value over time. SearchTrails' value for discovery becomes more apparent when it comes to collaborative search. The findings show that the keywords column can help to discover the context of browsing (W6, group NT, [1.43, 2.41]) or help discovering new search terms (W7, group NT, [1.73, 2.77]). So in general, SearchTrails can support discovery during complex search.

The last subtask of complex search to be confirmed is synthesis. Table 39 shows the most important questions related to synthesis from the intermediate evaluation. All results come from group SR. Synthesis is the step of complex search in which the found pieces of information are evaluated and connected. This process is related to the process of sensemaking of information (cf. (Evans & Chi, 2008a)), as described in Chapter 2.2.4. The main feature for synthesis in SearchTrails is the highlights overview, which condenses all found valuable pieces of information and their sources into an overview table.

Table 39: Statistical analysis of the questions on synthesis for the first phase of the second user study (Group SR, Question group I).

I	Synthesis evaluation	n	Average value	Confidence interval ($\alpha = 0.05$)
1	SearchTrails was helpful to capture relevant search results.	15	2.47	0.48
2	It felt safe to capture search results.	15	2.60	0.40
3	I looked into the search highlights after the actual search.	15	2.20	0.38
4	The highlights helped to gather all relevant resources for the report.	15	2.60	0.40
5	Being able to interact with the graph and its highlights helped to condense the relevant information.	14	2.14	0.39
7	I had SearchTrails and the tabs used for browsing both always visible (e.g. on two screens).	14	42.9%, 6 of 14	
8	Switching between the SearchTrails-Tab and the tabs used for browsing was cumbersome.	12	2.42	0.78

The first question aims for the participants' general impression of SearchTrails' support for synthesis. The participants' general impression is rather positive (I1, [1.99, 2.95]). Similar results are achieved for the feeling of safety of capturing search results (I2, [2.20, 3.00]). Participants agree that the highlights helped to gather all relevant resources for the report (I4, [2.20, 3.00]). Even if SearchTrails captures only texts and not images, these seem to be valuable for the participants. Good results can also be achieved for the actual practical impact of the highlights. Participants agree that they looked into the highlights after the actual search (I3, [1.82, 2.58]). Similarly, participants agree that the highlights help to condense the relevant information (I5, [1.75, 2.53]). This clearly indicates that the synthesis of information gets leveraged by using SearchTrails. We asked two questions aiming for the usability of SearchTrails during the process of synthesis. 6 of 14 participants from group SR had SearchTrails and the browsing window visible on two screens (I7). Therefore, 8 participants had it all on the same screen and needed to switch between the different views. Participants seem to agree that the switching can be cumbersome, but this was not felt by all of them, as the high confidence interval shows (I8, [1.64, 3.20]).

The described results improved during the second phase of the study, after which the same questions were asked to all participants again (question group V, cf. Table 40). The results for group SR have not changed much since the first phase (question group I). The values for statement V1 increased from 2.47 to 2.14 for group SR. For the other questions, the deviances from the values of the intermediate evaluation are also very small. It is also interesting to note that the switching between different views for browsing and synthesis is rated more unobtrusive in the second phase than in the first phase, and especially unobtrusive for group NT (V8, [2.68, 3.82]).

Table 40: Statistical analysis of the questions on synthesis for the second phase of the second user study (Question group V); split by the respective user groups.

V	Synthesis evaluation	Group SR (Report)			Group NT (Search trail)		
		n	Average value	Confidence interval (α = 0.05)	n	Average value	Confidence interval (α = 0.05)
1	SearchTrails was helpful to capture relevant search results.	14	2.14	0.27	12	2.00	0.23
2	It felt safe to capture search results.	14	2.64	0.42	12	2.33	0.48
3	I looked into the search highlights after the actual search.	14	2.29	0.50	11	2.00	0.25
4	The highlights helped to gather all relevant resources for the report.	13	2.62	0.40	12	2.58	0.36
5	Being able to interact with the graph and its highlights helped to condense the relevant information.	14	2.14	0.27	12	2.42	0.36
7	I had SearchTrails and the tabs used for browsing both always visible (e.g. on two screens).	14	57.1%, 8 of 14		12	50.0%, 6 of 12	
8	Switching between the SearchTrails-Tab and the tabs used for browsing was cumbersome.	14	2.64	0.64	12	3.25	0.57

It is remarkable that most values for group NT are more positive than for group SR (V1 – V4). Taking into account that the values for group SR in the second phase did not decrease, we can see that the overall opinion on synthesis is good for group SR and very positive for group NT in the second phase. This shows both an impact of the use case on the search support, but also the power SearchTrails possesses when being used for supporting collaboration.

This section presents the analysis of the questionnaire results for all three subtasks of complex search, aggregation, discovery, and synthesis. From these results we can deduce that all three subtasks are supported by SearchTrails. The results also show that some tasks are supported better than others. While the support for aggregation and synthesis is convincing, the support for discovery seems better when relying on the search results of other users. Overall, we can

confirm hypothesis H2.1: ‘SearchTrails eases single user complex search tasks by supporting aggregation, synthesis, and discovery’. Based on the results on hypothesis H2.1, research question RQ2.1: ‘Does SearchTrails help users coping with complex search tasks?’ can be answered positively, as SearchTrails helps the users with the aggregation, discovery, and synthesis tasks during complex search processes.

Hypothesis 2.2 – Raising awareness for the own search behavior

During a fruitful discussion within a doctoral colloquium, the question was raised whether SearchTrails could also serve as a tool for self-awareness. This was suspected because SearchTrails may raise the users’ self-awareness for search processes and the visited pages by its graph visualization. Based on this discussion, hypothesis H2.2 was formulated: ‘SearchTrails raises awareness for the own personal search behavior’.

In order to confirm this hypothesis I included question group G ‘Self-awareness evaluation’ in the questionnaires for the second user study. Table 41 presents an overview of the results for group SR from the intermediate evaluation.

Table 41: Statistical analysis of the questions on self-awareness for the first phase of the second user study (Group SR, Question group G).

G	Self-awareness evaluation	n	Average value	Confidence interval ($\alpha = 0.05$)
1	Sometimes, I visited pages that were irrelevant for my search (e.g. news pages).	15	2.27	0.63
3	Seeing the search graph helped me to be aware of my search results.	15	2.27	0.39
4	I became aware of pages I visit frequently.	14	2.29	0.54
5	List three insights that you gained when monitoring your own search behavior.	9	Free text question	

The statements on self-awareness are answered positively by the participants. Statements on frequently visited pages (G4, [1.75, 2.38]) and irrelevant pages being visited (G1, [1.64, 2.90]) show that SearchTrails raises the awareness for the pages visited during a search. When we asked the participants directly whether SearchTrails helps them to be aware of the actual search results, the answers are also positive (G3, [1.88, 2.66]). The comparably small confidence interval confirms this result. When we asked the participants about the insights they gained about their own search behavior, the answers provide hints that the participants reflect their search processes with the help of SearchTrails. Some of the participants statements from G5 are: *‘I can trace my search behavior, which helps me to remember my initial intentions’*, *‘I am distracted by visual materials, ideas, tools, design, accessories, etc.’*, *‘The complexity of searches increased’*, and *‘I realized how many pages I visited for a specific topic. I also realized that many of the sites I visited were relevant to the topic’*. One participant describes the search process as being guided by the clusters: *‘Mostly, I conduct searching in a cluster first. If there is no more info I'm interested in, I change to another cluster of search. The highlights come from only a few specific pages. The keywords are confined in a small range when I start searching, but grow larger as I connect to other sites which are provided by the search engine results’*. The results for the first phase are therefore positive.

Table 42: Statistical analysis of the questions on self-awareness for the second phase of the second user study (Question group T); split by the respective user groups.

T	Self-awareness evaluation	Group SR (Report)			Group NT (Search trail)		
		n	**Average value**	**Confidence interval ($\alpha = 0.05$)**	**n**	**Average value**	**Confidence interval ($\alpha = 0.05$)**
1	Sometimes, I visited pages that were irrelevant for my search (e.g. news pages).	14	2.50	0.59	12	1.58	0.28
3	Seeing the search graph helped me to be aware of my search results.	14	1.86	0.27	11	2.55	0.46
4	I became aware of pages I visit frequently.	14	1.79	0.29	12	1.83	0.56
5	List three insights that you gained when monitoring your own search behavior.	12	Free text question.		5	Free text question.	

Table 42 compares the average values for the same statements after the second phase, split by groups SR and NT. It is remarkable that the results now show larger differences than in the first phase. The differences in the values between groups SR and NT for statement T1 and T3 are statistically significant on a 5% error level. Statement T1 shows that group NT agrees significantly stronger to have visited irrelevant pages, while group SR shows similar values than in the first phase. This gives a hint that SearchTrails raises awareness for the visited pages, and that a given search trail may lead the participants to visit more irrelevant pages. It remains open whether these irrelevant pages are from the given trail or if the participants actively went there while extending the search trail. Both groups agree very similarly that they became aware of pages they visit frequently (T4). The values for statement T3 reveal that for group SR, the search trail visualization helps to be aware of the search results (T3, group SR, [1.59, 2.13]). This value has improved, compared to the one from the first phase (G3, group SR, [1.88, 2.66]). It is remarkable that the value for group NT is significantly lower (T3, group NT, [2.09, 3.01]) than for group SR. This could give a hint that the additional experience with SearchTrails by group SR has improved the participants' abilities to interpret the search trail. These skills may not be that developed for the participants of group NT, which are confronted for the first time with a given search trail and have to learn how to interpret it.

The answers to statement T5 provide further evidence that SearchTrails improves self-awareness. Similarly to statement G5, the participants state that Wikipedia is a good point to start a search from (*'First focus on search within the cluster Wikipedia. I open several tabs at the same time and make the highlights one by one.'*), that several pages add to the overall result (*'There were more than one page for same content. The information in search became more focused.'*), and that the search trail reveals that the participants tend to return to good resources (*'I used a strategy from general to more specific task. I used graph to return to pages that interested me.'*).

Summing up these results, I deduce that self-awareness can be raised by the use of SearchTrails. The results from the first phase seem to improve for the second phase, and a lower number of irrelevant pages seem to be visited by the participants. The participants from group NT return remarkably good results; while a lower score for statement T3 gives another hint that interpreting the search trail needs some training. Overall, **hypothesis H2.2 can be confirmed**: 'SearchTrails raises awareness for the own personal search behavior'. As a high awareness for the personal search behavior leads to a constant reflection of the performed action, we can deduce that SearchTrails also leads to more controlled search processes.

Hypothesis 2.3 – Improving the quality of the search process for complex search tasks

The previous hypotheses show that SearchTrails eases single user complex search and raises self-awareness during complex search processes. Most interesting is the influence of SearchTrails on the quality of the results of the complex search tasks. This section confirms hypothesis H2.3 'The use of SearchTrails improves the quality of the search process for complex search tasks'.

I define the search process in Chapter 2.2.2 as the sequence of actions related to search for information to reach an information goal. As the quality of a sequence of actions cannot be measured directly, I need to consider the quality of the search results as an indicator for the quality of the search process. Based on this definition, the search tasks induce a search process for our participants, who create search results. The results are the search trail, which basically is the JSON-log file that is stored on the server, and the report. The search trails' JSON file gets interpreted as the search graph, but also as highlights, clusters, etc. These artifacts are all results of the search process. Similarly, the report is a search result, as it is created by the participants during or after the search process, based on the material gathered during the search process. This means that the search trail can be considered a direct search result, as it resembles the search process. The report can be considered an indirect search result, as it consists of information processed by the searcher.

The first idea to measure the quality of the search process is to compare the quality of the search results, namely the reports. I started by comparing the grades of the reports between the groups SR and NT. The results of this analysis for the first phase of the study are shown in Table 43.

Table 43: Statistical results for the difference in report grades during the first phase of the second user study; split by the support by SearchTrails (significant).

1st phase Average grade of the reports		**Group SR** Used SearchTrails	**Group NT** No use of SearchTrails
Average grade		2.44	1.96
F-test	F-value	1.449	
	Critical F-value	1.731	
T-test	T-value	2.957	
	Critical T-value	1.992	

The results in this table show at first sight that the grades for the group that used SearchTrails during the first phase are significantly lower than for the group that did not use Search-

Trails. These results were not expected this way. The results suggest that SearchTrails corrupts the quality of reports by roughly two tendencies of a grade. In order to analyze all generated search results, we also analyzed the search trails as the direct artifacts of the search. During a workshop, we clustered the search trails into five categories. Those five categories resemble the five German standard university grades. For this clustering, we took into account the following four criteria:

- Amount of nodes. A large number of nodes implies that the participant most probably spent a large amount of time searching, which should influence the quality of the search trail positively. To not overestimate the number of nodes from search engines, we closed all clusters with search engine nodes before the clustering session.
- Depth of the search trail. We counted the length of the deepest loop before it reached a search engine page again. We suppose that this is a measure for the depth of the information that was collected during the search, which should also positively influence the quality of the search trail.
- Diversity of the search trail. We counted the number of clusters and evaluated the diversity of the different nodes. We suppose that this has a positive influence on the quality of the search trail.
- The overall visual complexity of the search trail. We judged the overall impression of complexity of the search trail, taking into count the aforementioned factors and degree of connectedness of the nodes.

The findings during this workshop confirmed the findings which are described in (Franken & Norbisrath, 2014a): The search trails enable a quick visual evaluation of the quality of the search trails. The resulting grading can be seen in Figure 87. The left half of the clusters is from the participants who used SearchTrails (group SR); the right half is from the participants which had no support during the search process (group NT). The five rows resemble the five clusters. The top row shows the trails which were clusters which were considered best, the lower row shows the worst trails. Table 44 shows the participants IDs connected to the search trail grades.

Figure 87: Result of the search trail clustering process for the first phase of the second user study.

Table 44: Mapping of the search trails of the first phase of the second user study to university grades; split by the support by SearchTrails.

Trail grades **1st phase of the study**	**Group SR** Used SearchTrails	**Group NT** No use of SearchTrails
1.0	2, 25	3, 23
2.0	6, 22, 29	19, 20
3.0	1, 13, 14, 18	8, 9, 11
4.0	5, 21, 26	4, 7, 12, 27, 28
5.0	17	24

The distribution shows a light tendency towards better results for the search trails from group SR. This is reflected by the statistical results in Table 45, where the average grade of the trails with SearchTrails support is better than without support. It is interesting to note that SearchTrails' influence on the average search trail grade is a positive 0.41 (cf. Table 45, search trail grades), where the influence on the report grade is a negative -0.48 (cf. Table 43, report grades). This means that the search trails improve with the support of SearchTrails, but the reports do not. However, these results are not statistically significant.

Table 45: Statistical results for the difference in search trail grades during the first phase of the second user study; split by the support by SearchTrails (not significant).

1st phase **Average grade** **of the search trails**		**Group SR** Used SearchTrails	**Group NT** No use of SearchTrails
Average grade		2.86	3.27
F-test	F-value	1.041	
	Critical F-value	2.671	
T-test	T-value	-0.876	
	Critical T-value	2.069	

Taking into account that SearchTrails' influence on the grades of the reports and on the grades of the search trails seem to contradict each other, I calculated the correlation between the report and the search trail grade for all participants as one group. As the university grades are ordinal scaled, I used Kendall's tau. It turns out that Kendall's tau-correlation is 0.007, which means that no correlation between the two samples exists. The quality of the report therefore is no indicator for the quality of the search trail.

These results trigger taking a closer look on the value of the report as an artifact for evaluating the quality of the search process. While the search trail reflects the complete search process, the report is a highly processed and condensed artifact, which mainly relies on the highlights. For these reasons, we take the quality of the search trail as a stronger indicator for the quality of the search process than the quality of the report. The report mainly depends on the searchers motivation and the ability to condense information; an ability which is trained during university education. As the participant sample consisted of university students in the first semesters of their master degree education, this ability may not be fully developed yet.

So far, the described findings indicate that the search trail may be the more important artifact for evaluating the quality of the search process. The findings suggest that the quality of the search trail seems to increase when using SearchTrails, however not significantly. The quality of the report has to be considered a less meaningful indicator for the quality of the search process. Its average grade seems to decrease when using SearchTrails. These results seem to indicate a relationship between good trails and bad reports, but a correlation between the search trail grades and the report grades could not be found when considering all participants as one sample group, as described above.

These results can already provide indication towards better search processes when using SearchTrails. A problem with the previous correlation analysis is that it mixes the results from the participants that used SearchTrails with results from participants that did not use SearchTrails. It is possible that there exist two different correlations between the data from the different groups. In this case, it is likely that no correlation can be determined, as correlation can only determine one linear relation within a sample, but not more. Therefore it is necessary to evaluate the data further, and split it by the support of SearchTrails.

First, I consider the data for the group that used SearchTrails. This group consists of 14 participants, from which eight have been using the highlight feature. I consider those participants to have used SearchTrails intensively. For these participants, it would be interesting to know whether the intensive use of SearchTrails makes a difference with respect to the average grade of the reports. Table 46 compares the average grades of the reports for the participants from group SR which used the highlights feature more or less intensively.

Table 46: Statistical results for the difference in report grades for all participants from group SR during the first phase of the second user study; split by the intensity of use of SearchTrails (significant).

1st phase. Group SR. Average grade of the reports		**Used highlights / Intensive user**	**Did not use highlights / Extensive user**
Average grade		2.11	2.88
F-test	F-value	2.907	
	Critical F-value	3.972	
T-test	T-value	2.379	
	Critical T-value	2.189	

Table 46 shows a difference with respect to the report grade between intensive and extensive users. This difference of the grades is large (0.77) and statistically significant. This means that when a user is intensively working with SearchTrails, the report is likely to become good. Accordingly, there exists a weak negative Pearson correlation between the number of highlights and the report grade (-0.315). Unfortunately, this correlation is not significant, so the value itself can just be a hint on the true relation. I correlate the number of highlights with the trail grade, which results in no correlation (0.009). This means that the number of highlights does not influence the grading of the trails. As there are not many nodes with highlights, I did not consider this a criterion for grading.

Even more interesting are the correlations between the search trail grade and the report grade for both groups SR and NT separately. For group SR, there exists a weak negative correlation between the search trail grade and the report grade: Kendall's tau correlation coefficient is -0.259, which is significant on a 10% level. This means that that there is a tendency that a good search trail results in a bad report. Accordingly, I did the analysis for the grades from group NT, where the participants had no support. For this group, there exists a weak positive correlation, where Kendall's tau correlation coefficient is 0.309 (also significant on a 10% level). For these participants, a good search trail results in a good report. These findings are resembled in Figure 88, where both grades are shown together with their regression lines for the respective user groups. This confirms the assumption from the earlier paragraphs of this section and explains the missing correlation. When considering all participants as one group, the two correlations eliminate each other.

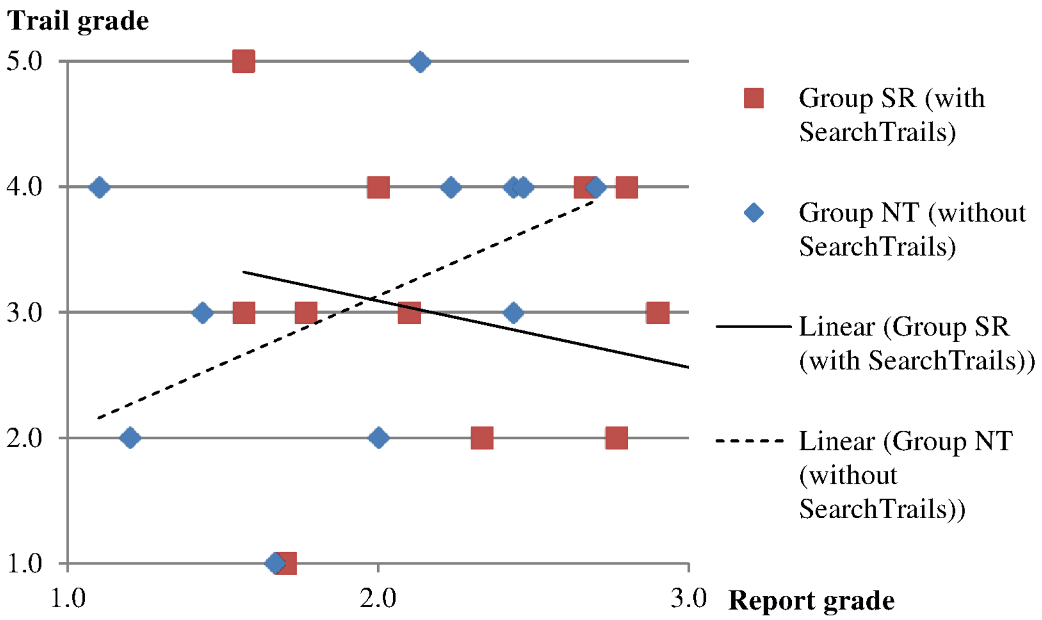


Figure 88: Correlation of search trail grade and report grade for each of the two participant groups SR and NT for the first phase of the second user study.

These results open room for an explanation. The participants without support have to collect their findings during the search with the help of note taking tools. These tools are most likely text-based (cf. the results on the used tools in Chapter 5.2.5.1) and can be transformed into a report easily. This results in increasing report grades when the grades of the trails increase (cf. the correlation between the search trail grade and the report grade for group NT in Figure 88). The participants who used SearchTrails have put efforts into creating the search trail with its highlights. The main findings of the search are stored as a part of the trail, which increases the value of the search trail for the searcher. Asking the participants to create a report forces them to linearize the non-linear trail, which would result in double work: First generating the non-linear trail, which is associative in its nature, then linearizing it into a report. With increasing complexity of the trail, the participants' motivation to take this effort may decrease. This can explain why increasing grades for the search trails lead to decreasing grades for the reports (cf. the correlation for group SR in Figure 88). These findings find confirmation from the evaluation for group ST: report grades decrease when using SearchTrails.

Summing up these findings, I conclude that reports are a less meaningful indicator for measuring the quality of the search process; search trails are better suited for these purposes. Concerning the search trail grades, I found that these are slightly better when using SearchTrails than without support. This implies that the use of SearchTrails improves the quality of the search process. Closer examination reveals the influence of SearchTrails on the report grades: When being supported, the participants invested their resources more into creating a trail, and less into writing a report, while the non-supported group showed a positive correlation. Taking into account the statistical significances for the correlations and the differences between the samples, I can **confirm hypothesis H2.3**: 'The use of SearchTrails improves the quality of the search process for complex search tasks'. The just given rationale therefore suggests that it is reasonable to assume that hypothesis H2.3 is true.

Hypothesis 2.4 – Enabling asynchronous collaborative search

The previous hypotheses could confirm that SearchTrails eases single user complex search, raises awareness for the search behavior and improves the quality of search results. The following two hypotheses deal with the value of SearchTrails as a supporting tool for collaborative search. SearchTrails intention is enabling asynchronous, collaborative search by the exchange of search trails. This section confirms hypothesis H2.4: 'SearchTrails enables asynchronous collaborative search by exchanging search trails'. For these analyses, the results from the second phase of the study are most important.

Hypothesis H2.4 deals less with a quantitative evaluation of search results, but more with an analysis of the feasibility of support for collaborative search with SearchTrails. The collaboration artifact of SearchTrails is the search trail itself. In order to confirm hypothesis H2.4, I show that the exchange of search trails is an enabler for collaborative search. In the best case, a search trail turns out to be superior when being compared with other collaboration artifacts. This can be achieved by showing that it is possible to recreate and extend a given search trail, and that the participants appreciate this novel way of interaction by making use of it.

In the first phase of the study, we requested the participants to write a report such that a person who did not perform the search would be able to understand the main findings of the search. From all reports that were handed in, we selected two medium good reports to serve as a collaboration artifact for group SR for the second phase of the study. We chose the corresponding search trails as collaboration artifacts for group NT. To confirm hypothesis H2.4, I analyze the results of the statements on evaluating and extending the given collaboration artifacts.

From the 15 participants of group SR, which received a report as an artifact, twelve participated actively and all were able to hand in a report after the second phase of the study. From the 14 participants of group NT, ten participated actively and extended the given search trail. From these participants, nine were able to hand in a report after the second phase of the study. The participant who did not hand in a report performed a very short search and extended the search trail only very little. This general decrease of active participants may be due to the upcoming exams and final assignment submissions, as one participant reported after the study. So in general, participants were able to work with a given search trail like with a given report, and the decreasing numbers are most probably not due to technical difficulties.

Table 47 and Table 48 show the participants' answers from the final questionnaire regarding the evaluation and the extension of the given artifact (Question groups Q and R). Comparing the potential savings of search efforts for the report and the search trail for the continuation of a search task (Q4), participants from group NT are more positive about the search trail (Q4, group NT, [2.15, 3.01]) than the participants from group SR (Q4, group SR, [2.41, 3.59]). These results show that the report is not considered to be able to save redundant searching, but the search trail is. These results get confirmed when asking for the time saving potential of a search trail (Q10). The results from group SR are less positive (Q10, group SR, [2.55, 3.73]) than the results from group NT (Q10, group NT, [1.80, 3.36]). Considering the extension of the given artifact, all participants are confident to have found additional relevant information (R1). It is interesting to note that group SR cannot agree as strong as group NT to be able to reuse many facts from the given artifact (R2). The results for group NT are slightly more positive, but with a smaller confidence interval (R2, group SR: [2.16, 3.42], group NT: [2.13, 2.87]) which indicates a higher value of the search trail. Statements R3 and R5 confirm that both groups could find additional information and that a given artifact on a topic eases writing a report.

Table 47: Statistical analysis of the questions on artifact evaluation for the second phase of the second user study (Question group Q); split by the respective user groups.

Q	Evaluating your colleagues artifact	Group SR (Report)			Group NT (Search trail)		
		n	**Average value**	**Confidence interval (α = 0.05)**	**n**	**Average value**	**Confidence interval (α = 0.05)**
4	**Group SR (Report):** The report helped to avoid redundant searching. **Group NT (Search trail):** The search trail helped to avoid redundant searching.	14	3.00	0.59	12	2.58	0.43
10	**Group SR (Report):** Evaluating the given report helped to save a lot of time. **Group NT (Search trail):** Evaluating the given search trail helped to save a lot of time.	14	3.14	0.59	12	2.58	0.78

Table 48: Statistical analysis of the questions on artifact extension for the second phase of the second user study (Question group R); split by the respective user groups.

R	Extending the artifact	Group SR (Report)			Group NT (Search trail)		
		n	**Average value**	**Confidence interval (α = 0.05)**	**n**	**Average value**	**Confidence interval (α = 0.05)**
1	Did you find additional relevant information?	14	2.14	0.33	12	2.00	0.23
2	**Group SR (Report):** I could use many facts from the given report for my report on the topic.	14	2.79	0.63	12	2.50	0.37

	Group NT (Search trail): I could use many highlights from the given search trail for my report on the topic.						
3	I could find facts of the search topic that my colleague did not already cover.	14	2.14	0.44	12	2.17	0.51
5	Compared to my first search, it was easier to write this report.	14	2.36	0.55	12	2.33	0.74
6	My colleague and I have produced a sufficient result on this topic.	14	2.50	0.47	12	2.17	0.45
7	SearchTrails enhances collaboration between colleagues / teammates.	14	2.64	0.51	12	2.25	0.66

The answers to questions R6 and R7 can provide some more hints on the collaborative search experience with reports and search trails. Participants from group SR agree (R6, [2.03, 2.97]) to have **produced** a sufficient result on he requested topic together with their (undisclosed) colleague. However, the level of agreement on this statement is more positive for the participants from group NT (R6, [1.72, 2.62]). For the statement whether SearchTrails enables collaboration between colleagues, the results are also positive. Again, compared to the results for group SR (R7, [2.13, 3.15]), the results for group NT are more positive (R7, [1.59, 2.91]).

In general, group NT was able to transform the given search trail into a report and perceived this as a collaborative result. Although both groups were positive about having produced a sufficient result, participants from group NT were even more positive about this than the participants from group SR. Also, the general impression of SearchTrails as a tool for enhancing collaboration is better for group NT, which may be explained by the unfiltered view on the collaborator's search process via the search trail. Even if these results are not statistically significant, these results **confirm hypothesis H2.4**: 'SearchTrails enables asynchronous collaborative search by exchanging search trails'. Based on these results, research question RQ2.3: 'Does visualizing the course of a web search help with understanding what other users have done during a search?' can be answered positively. These results are reinforced by the results on the 'Usability of SearchTrails', which can be found in Chapter 5.2.5.4. Research question RQ2.2: 'Does SearchTrails help users with continuing web searches?' can be answered for one of two cases. Considering the exchange of search trails between different users, the results on hypothesis H2.4 indicate that RQ2.2 'Ease continuation?' may be answered positive, as the exchanged artifacts allow users to continue the search sessions of other users. Considering the continuation of own web searches, no statistically significant results could be gathered, as the participants were not forced to work in multiple sessions. However, it can be supposed that when users are able to recover information from a given search trail well, recovering information from an own and known search trail may also be possible and eases the continuation of own search processes.

Hypothesis 2.5 – Exchanging search trails

Hypothesis H2.4 can confirm that exchanging SearchTrails enables asynchronous collaborative search. Although some hints are already given in the previous section, I still need to confirm that the search trail as a collaboration artifact has a significant advantage over a report as a collaboration artifact. Therefore, this section confirms hypothesis H2.5 'Exchanging search trails in collaborative search improves the quality of search results'. In contrast to hypothesis H2.3, which confirms that the use of SearchTrails improves the quality of the search process for single user complex search, hypothesis H2.5 examines the quality of the search results for collaborative complex search. I assume that when a collaboration artifact is given, the quality of the search result is more important than the quality of the search process. This is because the common goal of the collaborators is more on producing a collaborative result than in experiencing a highly qualitative search process. Therefore, hypothesis H2.5 deals with the quality of search results.

As already argued in H2.3 based on the definitions from Chapter 2.2.2, the search results are the search trail and the report. From these search results, the search trail can be considered a direct result of the search process, while the report is an indirect result. First, I compared the average grades of the reports, depending on the given collaboration artifacts. Table 49 shows an overview of the results.

Table 49: Statistical results for the difference in report grades for all participants during the second phase of the second user study; split by the type of the given artifact (significant).

2nd phase Average grade of the reports.		**Group SR** Given artifact: Report	**Group NT** Given artifact: Search trail
Average grade		3.20	2.21
F-test	F-value	2.471	
	Critical F-value	1.874	
T-test	T-value	4.732	
	Critical T-value	2.000	

These results show that the average grade of the reports seems to depend heavily on the type of the given artifact. The reports of group NT, whose participants I equipped with a search trail as a collaboration artifact, are graded approximately one full grade better than the reports of group SR, whose participants I equipped with a report as collaboration artifact. The results of an F-test show that the variance of the samples is different with a probability of more than 95%, such that a T-test for samples with different variances was performed. This T-test therefore made use of the Welch-correction. The T-test can confirm that the difference in the average grades is significant on a 5% level.

Similar to the clustering of the search trails from the first phase, we clustered the search trails from the second phase a workshop. Figure 89 shows the final clustering of the trails. We excluded all trails from this clustering which were not extended by the participants. This clustering also made use of the full spectrum of academic grades, even if the overall quality of the search trails seemed to be better for the second phase than in the first phase. A search trail which was graded 5.0 for the second phase of the study would have been graded with a 4.0 in the context of the first phase of the study.

Figure 89: Result of the search trail clustering process for the second phase of the second user study.

The clustering of the search trails from the second phase shows that the search trails from group SR who received a report are spread across the full range of academic grades. The trails of group NT are all in the upper half of the range of grades. Table 50 summarizes the results of the clustering.

Table 50: Mapping of the search trails of the second phase of the second user study to university grades; split by the type of the given artifact

Grade	Group SR Given artifact: Report	Group NT Given artifact: Search trail
1.0	2	3, 4, 19, 28
2.0	1, 6, 22, 29	11, 12, 24
3.0	5	8, 23, 27
4.0	9, 14, 18, 25	-
5.0	10, 13	-

These results are reflected in the average grades of the search trails. For group SR, the average grade for the search trail is 3.17, while for group NT the average grade is 1.78. These results are statistically significant (cf. Table 51).

Table 51: Statistical results for the difference in search trail grades for all participants during the second phase of the second user study; split by the type of the given artifact (significant).

2nd phase Average grade of the search trails.		**Group SR** Given artifact: Report	**Group NT** Given artifact: Search trail
Average grade		3.17	1.78
F-test	F-value	2.575	
	Critical F-value	3.313	
T-test	T-value	2.734	
	Critical T-value	2.093	

The statistically significant differences in the search trail grades may not be too surprising, as group NT was given a proper search trail to build upon. This influenced the grading, as there was only little chance for these participants to deteriorate the given trails, e.g. by deleting nodes and highlights. Therefore, I have to further investigate the value which the participants of group NT have added to the given trails. Table 52 shows the average values of the key parameters for the search trails from group SR and from group NT. The values for group NT show exclusively the added value during the search process for the second phase of the study; the values of the given search trails were already subtracted. The last two rows show whether the difference between the average values of group SR and group NT (added value) is significant on a 5% or a 10% error level.

Table 52: Statistical data of search trails and their significance from the second phase of the second user study; split by the type of the given artifact.

	# Nodes	**# Edges**	**# Steps**	**# Clusters**	**# Highlights**	**Duration**	**Time not on search engines**	**Average loop length**
Group SR Given artifact: Report	34.4	64.4	108.3	2.8	2.9	2752	1532	2.8
Group NT (added value) Given artifact: Search trail	23.0	28.4	42.8	1.0	1.0	1497	1006	5.0
T-test 5%	no	sig	sig	sig	no	no	no	sig
T-test 10%	no	sig	sig	sig	no	sig	no	sig

The key properties of a search trail are the number of nodes, edges, steps, clusters, and highlights. Other key characteristics are the duration and the number of seconds not spent on search engine pages. The last characteristic is the average loop length. The numbers of nodes, edges, and highlights of a search graph are self-explaining. The number of steps is the number of user induced actions during the search process for navigation from one node to another node.

When a user walks the same path within a graph several times, the graph is not altered anymore, but the number of steps increases. The number of clusters is the number of hosts from which more than three different web pages were visited. The duration is the number of seconds in which the participants have searched actively, meaning that no interruptions of more than 15 minutes occurred. When two user-induced events are more than 15 minutes apart, the times are counted as idle times. I furthermore calculate where the participants have spent their time. The second to last column shows the number of seconds that were spent on non-search engine pages. The last column shows the average loop length. I count the length of all paths that start at a search engine until the path reaches a search engine page again and divide this by the total number of paths. This number serves as an indicator for the participants' average depth with which the participants dived into the topic.

Table 52 shows that all key characteristics of the search trails are on average larger for group SR than they are for group NT. These values show that the value that was added to the given material is smaller for group NT than for group SR. This lower added value for group NT did not occur randomly, but is significant in many cases: The lower numbers of edges, steps, and clusters are significant on a 5% error level, while the shorter duration is significant on a 10% error level. The participants from group NT also produced significantly longer loops than the participants from group SR. This means that these participants did longer tours through the Internet before going back to search engine pages and therefore dived deeper into the domain than the participants from group SR. A search trail therefore seems able to avoid redundant searching and brings the participants into a position where they are able to produce better results with lower efforts, which means that they are significantly more efficient.

The presented results show that search trails as collaboration artifacts have an advantage over written reports as collaboration artifacts. The results show that the participants who were given a search trail invested less resources into extending the given material than the participants who were given a report. These differences are significant in most cases. Altogether, the collaborative effort led to significantly better search results. Both the report and the search trail improved significantly. Adding to that, the results also show that the participants of group NT were significantly more efficient: These participants invested only 54% of the time that group SR needed to invest, but ended up with a report that was on average one full grade better than from group SR. These results show that I can strongly **confirm hypothesis H2.5**: Exchanging search trails in collaborative search improves the quality of search results. The results on hypothesis H2.5 allow answering research question RQ2.4: 'Does SearchTrails ease collaboration between users by exchanging web searches?' positively. Especially the impact of search trails as collaboration artifacts on the grades of the reports in the second phase of the study confirms that SearchTrails eases collaboration and increases the efficiency of the searchers.

5.2.5.4 Additional findings from the second user study

The previous sections show that all hypotheses can be confirmed by the results of the study. As the hypotheses were developed before the study was performed, some additional findings can be derived from the gathered data. These findings include differences on the experienced and actual timespans during the study, discrepancies between times used, remarks on the self-awareness of the study participants, and the lightweightness of SearchTrails itself, but even more results are presented.

Some remarks on times

The subjective feeling of time passing is not necessarily connected to the actual time passing. In the questionnaires, I asked the participants several times for the time they expect to need for a task and the time they actually needed. I then compared these subjective measures of time with the objective measure of the time span covered by the search trails and subtracted idle times from this. The results show that the participants overestimated the time they invested into the actual search. Table 53 shows more detailed results.

Table 53: Estimations on the amount of time needed for the tasks of the second user study and real numbers from the statistical analyses; split by study phases and groups.

		Phase 1		**Phase 2**	
		Group SR Used SearchTrails	**Group NT** No use of SearchTrails	**Group SR** Given artifact: Report	**Group NT** Given artifact: Search trail
1	Estimation before searching (s)	2927 (C1)	2830 (C1)	5125 (N1)	4200 (N1)
2	Estimation after search (s)	5846 (D2)	3901 (D2)	4075 (O2)	4000 (O2)
3	Real duration (s)	4034	3373	2752	1497
4	Deviation (row 2/ row 3) (s)	1812	528	1323	2503

Before the start of the first phase, participants from both groups SR and NT estimated the time necessary for performing the search tasks very similar (question C1, phase 1, row 1). When being asked at the beginning of the intermediate questionnaire, the participants from group SR provided a considerably higher estimation than the participants from group NT (question D2, phase 1, row 2). Looking at the real duration, which is derived from the actual search trails and cleaned by the idle times, less significant deviations emerge (phase 1, row 3). Group SR has invested more time into the actual search, but not as much as estimated. The deviation between the estimation in the intermediate questionnaire and the actual duration is way higher for the participants from group SR than for the participants from group NT (phase 1, row 4). It seems that participants seem to overestimate the time needed for a task when being confronted with something new that requires some learning effort.

For the second phase of the study, group SR estimated to require more time than group NT, maybe due to the experiences from the first phase (question N1, phase 2, row 1). However, the estimation in the final questionnaire does not differ significantly between the groups (question O2, phase 2, row 2). Each group estimated to have needed roughly the same amount of time. When comparing this estimation with the time needed, group NT seems to have overestimated the required time even more than group SR (phase 2, row 4), as group NT has spent significantly less time searching than group SR (phase 2, row 3). Similar to the first phase, it could be that the necessity to spend some effort for learning how to work with SearchTrails has raised the estimations. These results confirm the results from the first phase, and indicate that the use of a tool that requires learning makes the participants overestimate the necessary efforts for a task.

Self-awareness of the participants

During the second user study, we asked for the participants' agreement on several statements about their feeling of the fulfillment of the tasks and the coverage of the topic. In all three questionnaires (Q1, Q2, and Q3), statements towards self-awareness can be found. The questions are shown in Table 54.

Table 54: Overview of questions on self-awareness from all three questionnaires (Q1, Q2, and Q3) of the second user study.

Question ID	Question
Q1 - C3	I feel confident to cover the topic thoroughly.
Q2 - D4	I feel confident to have covered the topic thoroughly.
Q2 - J1	I think I have found all relevant facts that can be found on my search task.
Q2 - N3	I feel confident to cover the topic thoroughly.
Q3 - O4	I feel confident to have delivered a thorough answer to the question.
Q3 - R3	I could find facts of the search topic that my colleague did not already cover.
Q3 - R4	I could improve or extend the given report.
Q3 - W1	I think I have found all relevant facts that can be found on my search task.
Q3 - X2	I searched longer with SearchTrails than I would have searched without the tool.

I then split the level of agreement on these statements by the groups SR and NT and correlate all questions with each other and with the report and the search trails' grade. For these analyses, I use the Pearson correlation coefficient, as the data is interval scaled. It turns out that for both groups, large significant correlations can be found between questions R3 and R4 (from questionnaire Q3). As these statements were presented next to each other, this does not surprise a lot, as it can be explained by the question row effect.

Table 55: Overview of correlations between questions on self-awareness within the three different questionnaires from the second user study for group SR.

Group SR		Q1	Questionnaire Q2			Questionnaire Q3					Grades	
		C3	D4	J1	N3	O4	R3	R4	W1	X2	Report Grade	Trail Grade
Q1	C3	1.000										
Q2	D4	0.135	1.000									
	J1	-0.113	0.317	1.000								
	N3	0.153	0.392	0.236	1.000							
Q3	O4	0.277	0.377	0.056	**0.459***	1.000						
	R3	-0.226	0.395	0.328	-0.085	0.143	1.000					
	R4	-0.071	0.242	0.063	-0.085	-0.011	**0.897***	1.000				
	W1	0.394	0.334	0.213	0.204	**0.551***	0.290	0.290	1.000			
	X2	-0.010	0.273	0.092	0.080	0.386	0.240	-0.025	0.213	1.000		
Grades	Report Grade	-0.055	-0.062	-0.159	-0.283	-0.164	-0.212	-0.255	-0.204	0.104	1.000	
	Trail Grade	0.136	0.163	-0.056	-0.114	-0.394	0.310	**0.497***	-0.175	**-0.405**	-0.315	1.000

For group SR, it is interesting that there are hardly any correlations between the statements at all: The correlations for only three question pairs exceeds +/- 0.4 (cf. Table 55, * marks the correlations significant on a 5% level). There are no correlations between the opinion on one of the statements and the grade of the report. For group SR there is a significant positive correlation (0.497) between R4 and the search trail grade, which means that the agreement to this statement correlates with a good grade for the search trail. Inversely, there is a negative correlation (-0.405) between X2 and the search trail grade, which means that agreement to this question correlates with bad grades, however this is not significant. There is no correlation between the questions from questionnaire Q2 (middle grey box), and hardly a correlation between the questions from questionnaire Q3 (lower right grey box). Interpreting the correlations in detail is hard, as too many influence factors could have an impact on the results. The general result is more interesting: The answers to the self-awareness questions do not correlate with each other for the participants of group SR, and the self-awareness of the participants does hardly correlate with the actual quality of the search results.

The results for group NT are slightly different (cf. Table 56, * marks the correlations significant on a 5% level). The correlation table shows more correlations between the questions: the correlation for 17 question pairs exceeds +/- 0.4. Overall, there is still only one correlation between a question and the report grade (W1). Group NT shows correlations between all statements from questionnaire Q2 (middle grey box), and for six out of ten statements from questionnaire Q3 (lower right grey box). Group NT also shows three positive correlations between the agreements to statements and the search trail grade, from which two come from the final questionnaire (W1 (0.425) and X2 (0.408)). Again, these correlations are not significant. These results show that the answers from group NT are overall more consistent than the answers from group SR, but there are no significant correlations between answers to the statements and the search trail or report grades.

Table 56: Overview of correlations between questions on self-awareness within the three different questionnaires from the second user study for group NT.

Group NT		Q1	Questionnaire Q2			Questionnaire Q3					Grades	
		C3	D4	J1	N3	O4	R3	R4	W1	X2	Report Grade	Trail Grade
Q1	C3	1.000										
Q2	D4	-0.083	1.000									
	J1	0.322	**0.806***	1.000								
	N3	**0.432**	**0.432**	**0.557***	1.000							
Q3	O4	-0.184	-0.184	-0.130	0.389	1.000						
	R3	-0.392	0.280	0.191	0.184	**0.665***	1.000					
	R4	**-0.563***	-0.165	-0.267	0.112	**0.815***	**0.510***	1.000				
	W1	-0.174	**0.522***	0.289	**0.528***	**0.454**	0.322	**0.444**	1.000			
	X2	**0.579***	0.313	**0.543**	**0.597***	-0.225	**-0.408**	-0.323	0.093	1.000		
Grades	Report Grade	0.302	-0.309	-0.043	0.262	-0.021	-0.397	-0.074	**-0.427**	0.399	1.000	
	Trail Grade	0.220	-0.020	0.113	**0.481**	0.309	-0.236	0.314	**0.425**	**0.408**	**0.439**	1.000

The presented results and correlations show that there is a certain discrepancy between the participants' agreement to different statements and the relation to the actual search results. This implies that the questionnaire results are often inconsistent and can only give limited hints towards the participant's performance in the search tasks.

Correlation of search artifact and search trail characteristics

In the section on confirming hypothesis H2.3 (cf. Chapter 5.2.5.3), the correlation between the search trail and the report grades was shown for the first phase. For the second phase, a different picture emerges. Figure 90 shows the distribution of report and search trail grades and the correlations between the values, depending on the given collaboration artifact.

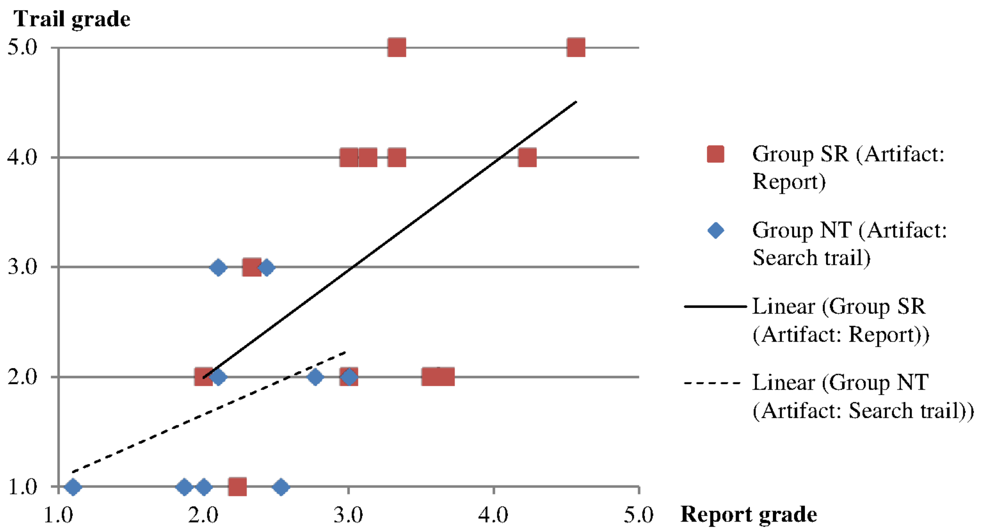


Figure 90: Correlation of search trail grade and report grade for each of the participant group SR and group NT for the second phase of the second user study.

For this phase, both correlations are positive. For group SR, which was given a report, the report grade and the search trail grade correlate with a Kendall's tau correlation coefficient of 0.429 (significant on a 5% level). For group NT, which was given a search trail, the report and the search trail achieve a Kendall's tau correlation coefficient of 0.365 (significant on a 10% level). For group SR, better search trails have therefore not led to inferior reports like in the first phase of the study. This can be explained by two reasons. First, the motivation of the participants may have played an important role, as some participants dropped out for the second phase of the study and therefore, the more motivated participants remained, which created a good report when creating a good search trail. This could explain the high correlation for group SR, which received the report. The second reason could be that we graded the complete search trails from the participants of group NT including the given search trail. On average, their trails therefore have to be better than the trails of group SR. This can explain the low correlation for group NT, as the variance in the search trail grades is lower than in the first phase of the study. When creating an overall trend line for all samples from the second phase; it turns out that this would be rather steep, as I calculated a 5% significant Kendall's tau of 0.539. For the first phase, this computation returns no correlation, as the correlations for the two single groups point into opposite directions. In general, it seems like the provision of a collaboration artifact induces a higher correlation between the search trail grades and the report grades than in the first phase.

I also correlate the key characteristics of the search trails from the second phase with each other and with the grades of both the search trail and the report. Table 57 shows the Pearson correlation coefficients of the search trail characteristics from the second phase with each other and with the report and the search trails' grades. For this large analysis, I choose to analyze the characteristics of the search trails from the second phase, as all participants had the chance to interact with the search trail during this phase. The table shows the Pearson correlation coefficients of the search trail characteristics with each other and with the report and the search trails' grades. The table highlights the positive correlations between 0.4 and 0.6 in green and the high positive correlations above 0.6 in blue. The negative correlations between -0.4 and -0.6 are highlighted in orange, and the high negative correlation below -0.6 are highlighted in red. All colored correlations are significant on a 5% level. It is not meaningful to elaborate on all correlations, but some are remarkable. Several correlations exist between the search trail characteristics, as most characteristics influence each other.

One remarkable observation is that it seems not to be relevant how much visits a participant made to search engine pages (row/column 2 in Table 57), as the only correlations are with related search trail statistics, but not with the report grade. There are hardly any correlations between the number of search engine steps (row/column 5) or the time spent on search engine pages (row/column 16) with other search trail characteristics, including the grades of the search trails and the grades of the reports. Surprisingly, the number of highlights does hardly correlate with any search trail characteristics (row/column 12), but shows a weak negative correlation with the report (cell R22/C12: -0.430) and the search trail grades (cell R23/C12: -0.508). This means that a high number of highlights is not correlated with the search trail characteristics, but has a positive impact on the grades, which confirms the findings on the intense use of SearchTrails from hypothesis H2.3 in Chapter 5.2.5.3.

What seems to have a positive impact on both grades are the following characteristics: The number of non-search engine nodes (row/column 3) and steps (row/column 6), the number of clusters (row/column 9) and keywords (row/column 11), the average loop length (row/column 7), the real duration of the search (row/column 15) and the time spent on non-search engine pages (row/column 17). The fact that these numbers have a positive influence on the search trail grade is not very surprising, as we evaluated the search trails with respect to these criteria. What is surprising is that the search trails' grades from our workshop correlate well with many single characteristics of the search trails, and also the grades of the reports. This confirms that SearchTrails as a tool eases the assessment and evaluation of search results. Additionally, these properties also correlate with the report grades. This implies that a good search trail leads to a good report during the second phase of the study, which is also reflected in the correlation between the search trail grade and the report grade (cell R23/C22: 0.671).

The correlations between search trail grade and average loop length (cell R23/C7: -0.607), number of keywords (cell R23/C11: -0.779), real duration of the search (cell R23/C15: -0.624), and the time spent on non-search-engine pages (cell R23/C17: -0.704) are astonishing. These characteristics could not be seen from the graph visualization, and still have a high correlation with the search trail grades. Characteristics like the average loop length speak for the searchers' motivation to dive into a topic. The number of keywords speaks for the diversity of the search trail and the search itself. Of course, a longer search should result in a better artifact, and also the time on real result pages should do so. For these characteristics, all correlations with report grade are below -0.642, so they are even stronger than the correlations with the search trail grade. All these factors have a positive impact on both the search trail and the report grade. Interestingly, the percentage of steps spent on search engines and of time spent on search engines have a large negative impact on both report and search trail grades.

Table 57: Overview of all correlations between statistical measures of the search trails from the second phase of the second user study and the search trail grades and report grades for all participants.

Correlations of search trail properties from phase 2 of the second user study	**Row index**	**# Nodes**	**# Search engine nodes**	**# Real nodes**	**# Steps**	**# Search engine steps**	**# Real steps**	**Average loop length**	**# Unique hosts**	**# Clusters**	**# Edges**	**# Keywords**	**# Highlights**	**# Events**	**Idle seconds**	**Real duration**	**Time on search engines**	**Time elsewhere**	**# Sessions**	**% of nodes on search engines**	**% of steps on search engines**	**% of time on search engines**	**Report grade**	**Search trail grade**
Column index		**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**	**C13**	**C14**	**C15**	**C16**	**C17**	**C18**	**C19**	**C20**	**C21**	**C22**	**C23**
# Nodes	**R1**	1.000																						
# Search engine nodes	**R2**	0.878	1.000																					
# Real nodes	**R3**	0.966	0.724	1.000																				
# Steps	**R4**	0.529	0.495	0.494	1.000																			
# Search engine steps	**R5**	0.026	0.188	-0.064	0.651	1.000																		
# Real steps	**R6**	0.674	0.506	0.697	0.815	0.091	1.000																	
Average loop length	**R7**	0.243	-0.001	0.351	0.017	-0.520	0.420	1.000																
# Unique hosts	**R8**	0.794	0.658	0.787	0.501	0.098	0.582	-0.018	1.000															
# Clusters	**R9**	0.675	0.388	0.763	0.461	-0.132	0.706	0.523	0.506	1.000														
# Edges	**R10**	0.755	0.692	0.714	0.804	0.168	0.927	0.189	0.730	0.601	1.000													
# Keywords	**R11**	0.806	0.538	0.870	0.493	-0.037	0.675	0.243	0.784	0.739	0.669	1.000												
# Highlights	**R12**	0.242	0.007	0.345	0.052	-0.224	0.239	0.255	0.337	0.538	0.193	0.385	1.000											
# Events	**R13**	0.839	0.764	0.796	0.876	0.492	0.774	0.083	0.700	0.592	0.810	0.726	0.125	1.000										
Idle seconds	**R14**	0.343	0.142	0.418	0.319	-0.103	0.497	0.362	0.328	0.660	0.395	0.481	0.428	0.340	1.000									
Real duration	**R15**	0.445	0.226	0.520	0.485	-0.142	0.745	0.597	0.323	0.778	0.617	0.617	0.350	0.471	0.696	1.000								
Time on search engines	**R16**	0.044	0.100	0.010	0.123	0.147	0.049	-0.329	0.171	0.051	0.168	0.183	-0.070	0.102	0.213	0.344	1.000							
Time elsewhere	**R17**	0.460	0.213	0.548	0.481	-0.190	0.777	0.720	0.297	0.811	0.609	0.606	0.390	0.472	0.681	0.968	0.099	1.000						
# Sessions	**R18**	0.602	0.432	0.633	0.386	-0.123	0.600	0.482	0.522	0.770	0.551	0.573	0.216	0.492	0.639	0.720	0.173	0.717	1.000					
% of nodes on search engines	**R19**	0.198	0.601	-0.040	0.325	0.537	0.016	-0.453	0.087	-0.192	0.227	-0.099	-0.337	0.344	-0.138	-0.147	0.263	-0.226	-0.034	1.000				
% of steps on search engines	**R20**	-0.342	-0.081	-0.450	0.025	0.690	-0.495	-0.869	-0.170	-0.572	-0.328	-0.384	-0.394	-0.064	-0.396	-0.554	0.337	-0.676	-0.489	0.509	1.000			
% of time on search engines	**R21**	-0.425	-0.183	-0.514	-0.337	0.271	-0.649	-0.769	-0.232	-0.701	-0.480	-0.491	-0.504	-0.365	-0.434	-0.604	0.454	-0.760	-0.462	0.305	0.819	1.000		
Report grade	**R22**	-0.550	-0.237	-0.659	-0.301	0.398	-0.685	-0.642	-0.400	-0.657	-0.544	-0.687	-0.430	-0.415	-0.520	-0.693	-0.013	-0.741	-0.460	0.445	0.682	0.616	1.000	
Search trail grade	**R23**	-0.868	-0.637	-0.902	-0.450	0.235	-0.757	-0.607	-0.651	-0.813	-0.759	-0.779	-0.508	-0.668	-0.482	-0.624	0.124	-0.704	-0.583	0.130	0.712	0.775	0.671	1.000

These findings show which characteristics have an impact on the quality of the search results and which have not. Some characteristics have a large impact on the grades; other influences are smaller than expected. In case of the highlights, the influence may have been higher when the highlights would have been used more frequently.

Time based learning effects

After each time the participants worked with SearchTrails, they were asked to state whether they used each feature of SearchTrails, and how important it was during the search process (question groups F and S). This resulted in group SR evaluating SearchTrails' features twice, and group NT evaluating its features once.

In both question groups F and S, one question turned out to be problematic. We asked for each feature of SearchTrails, whether it 'was a good idea, but a bad implementation'. This question could be misleading, as positive and negative opinions got mixed up in that question. If participants did not read the question completely, it could happen that a participants' good intention lead to a bad result for SearchTrails. For group SR, there are 22 negative mentions from 15 participants (1.47 per participant) for the features after the first phase, and 16 negative mentions from 14 participants (1.14 per participant) after the second phase. Group NT produced 10 negative mentions from 12 participants (0.83 per participant). Even if the results for these statements have to be considered with caution, they indicate a rising understanding of the features, and a very good start for group NT.

Table 58: Overview of the evaluation of each of the single features of SearchTrails; split by the phases of the second user study and the user groups.

F / S	SR, 1st phase			SR, 2nd phase			NT, 2nd phase		
Feature	Used in % of 15	Average importance	Confidence interval (α = 0.05)	Used in % of 14	Average importance	Confidence interval (α = 0.05)	Used in % of 12	Average importance	Confidence interval (α = 0.05)
Graph visualization	87	2.64	0.55	93	1.92	0.36	92	1.80	0.46
Keyword table	47	3.40	0.69	57	2.56	0.76	75	2.00	0.62
Select & search keywords	27	3.82	0.66	29	2.71	0.76	50	1.86	1.00
Blue nodes	74	2.43	0.55	50	2.33	0.69	58	1.00	0.00
Red nodes	13	2.78	0.74	36	2.71	0.65	33	1.25	0.42
Clusters	60	2.46	0.46	43	2.57	0.54	50	1.50	0.40
Collapsing clusters	20	2.86	0.92	57	2.33	0.69	67	2.17	0.72
Deleting nodes	20	3.00	0.97	29	2.25	0.67	17	1.75	0.81
Continuing a search	60	1.75	0.73	43	1.71	1.03	67	1.63	0.59
Adding highlights by 'ALT'	27	2.78	0.86	21	2.43	0.78	18	2.50	0.85
Adding highlights manually	7	3.25	1.02	0	-	-	9	2.00	0.92
Deleting highlights	27	2.90	0.81	14	2.50	0.89	17	2.50	0.85
Marking pages irrelevant	7	3.17	1.26	7	2.40	0.70	17	1.33	0.53
Clicking URLs	53	2.09	0.89	64	1.80	0.72	42	1.33	0.60

Table 58 shows the result for all three single-feature evaluations. It depicts the percentage of users who stated to have used the feature and the average value of the importance of the feature during the search. The importance was rated on a 5-point Likert scale, on which 1 means that the feature was 'very important', and 5 means that the features was 'unimportant / useless'.

As it is not meaningful to explicitly interpret every single result, I interpret the larger trends. The first and most astonishing trend is that the average importance for every feature decreases from left to right. The only exception is the importance of the clusters, which increases very slightly from 2.46 to 2.57, before it decreases to 1.50. The overall results show that SearchTrails' features become more important for group SR as the experience with SearchTrails increases. Group NT evaluates all features better than group SR, even if they worked with SearchTrails just once. This shows that the selection of a proper use case influences the results, and furthermore that SearchTrails can perform very well when it is used the intended way.

Figure 91 shows which features were used most. To generate the visualization, the percent values for the feature usage of group SR during the first phase were sorted in descending order (monotonic decreasing blue line). Obviously, the graph visualization was most important for the participants from all groups. The next important features are the blue nodes (which mark the nodes on which highlights were set), the continuing of searches, clusters, and clicking URLs to get back to specific pages. All other features were used by less than 50% of the participants from group SR in the first phase.

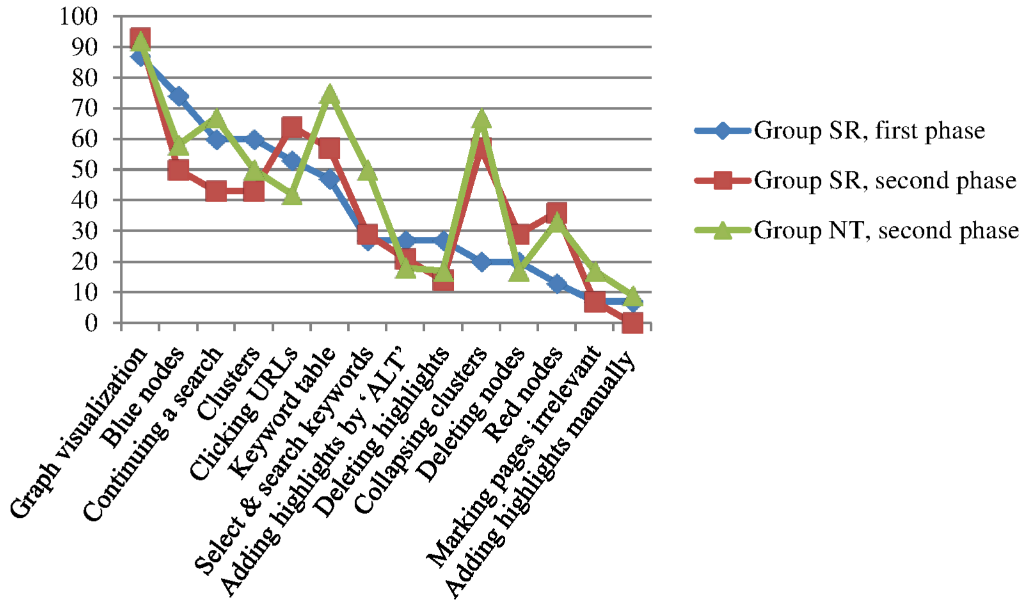


Figure 91: Visualization of the importance of SearchTrails' features for the individual groups.

Comparing the usage of features during the first phase to the usage of features during the second phase, it is interesting that there are some outliers from the decreasing course of the values from group SR in the first phase. These outliers behave very consistently for both group SR and group NT during the second phase. This gives a hint that some features become more important when participants have more experience with SearchTrails or need to evaluate a given search trail. For these participants features like the keyword table or collapsing clusters are also important. Some features decrease in importance, such as blue nodes, clusters, or deleting highlights. In case of deleting highlights, this may be due to the necessity to extract as much information as possible from the given material. For blue nodes or clusters, the lower results were surprising, as they also ease the extraction of information from a given trail. Other features increase in importance, such as keyword table, collapsing clusters, or red nodes. This can be explained by the evaluation of the given material, where it is important to extract information, but not to delete information. Although these differences are not statistically significant, they can give a hint that features for the extraction of information gain importance with increasing experience with SearchTrails, or that SearchTrails requires the users to learn its features to be able to fully appreciate them and use them efficiently.

Lightweightness

Although there exist a lot of collaborative search support systems, their practical impact remains rather limited (cf. Chapter 2.2.5). This is mainly because the tools developed for this purpose tend to be complex and are therefore heavyweight (Ringel Morris, 2013; Kelly & Payne, 2014). This is why avoiding the burden of being heavyweight becomes important. Naturally, every interaction with a system has to be learned and understood by the users, and the findings from the previous sections confirm that there has been a process of learning and understanding SearchTrails. Although a fuzzy concept like lightweightness is hard to measure explicitly, we want to find some hints on whether the participants consider SearchTrails lightweight or not.

During the final evaluation, three statements were made regarding lightweightness of SearchTrails. Table 59 compares the participants' level of agreement for these statements.

Table 59: Statistical analysis of the questions on lightweightness for the second phase of the second user study (Question group 1); split by the respective user groups.

1	**Lightweightness**	Group SR (Report)			Group NT (Search trail)		
		n	**Average value**	**Confidence interval (α = 0.05)**	**n**	**Average value**	**Confidence interval (α = 0.05)**
1	The concept of SearchTrails is immediately clear.	14	2.21	0.49	11	2.09	0.47
2	SearchTrails is a lightweight tool.	13	1.77	0.38	11	2.09	0.39
3	The amount of features is sufficient.	14	2.50	0.43	11	2.18	0.42

The results show for statement 1.1 that the participants from both group SR ([1.72, 2.70]) and group NT ([1.62, 2.56]) can agree that the concept of SearchTrails becomes clear to them without larger problems. There is no apparent difference between the levels of agreement for both groups. Statement 1.2 shows a larger agreement on the impression of lightweightness for group SR ([1.39, 2.16]) than for group NT ([1.70, 2.48]). This can be explained by the users of group NT having used SearchTrails only once before being asked to agree on the statement. It could be that the participants of group NT could cope well with the search task, but felt more demanded by learning how to interact with SearchTrails, which lowers the impression of working with a lightweight tool. Statement 1.3 can provide another hint that confirms this assumption, as group NT agrees more ([1.76, 2.60]) that the amount of features is sufficient than group

SR ([2.07, 2.93]). As the participants of group NT were occupied with understanding SearchTrails as it is, they rather think that the amount of features sufficient, while more experienced users may already think about improvements or extensions.

These results show that there were no larger problems with understanding SearchTrails and give a hint that SearchTrails can be seen as a lightweight approach for improving collaborative search processes. Literature review (cf. e.g. the results on lightweight systems in Chapter 2.2.5 from (Ringel Morris, 2013)) shows that this is a prerequisite for creating a potentially powerful system. The results allow answering research question RQ2.5: 'Is SearchTrails lightweight enough to be attractive for users?' positively. Even if not statistically significant, SearchTrails avoids overcomplicated features and returns very positive results for the participant group SR and even for group NT, which worked for the first time with SearchTrails.

Search trail complexity

I measured the complexity of the search trails objectively by applying Minoli's graph complexity algorithm (Minoli, 1975), as already described in Chapter 2.6.2. This algorithm defines the 'combinatorial graph complexity' χ of a given graph G = (V, E) in a more abbreviated version than shown in Chapter 2.6.2 as:

$$\chi(G) = \frac{ne}{n+e} \sum_{\substack{(v_i, v_j) \\ i>j}} \sigma_{i,j}$$

where

- $\chi(G)$ is the complexity of a graph G = (V, E)
- V = $\{v_i, \text{i} \in \mathbb{N}\}$ is the set of vertices
- E = $\{(v_i, v_j), v_i \in V\}$ is the set of edges
- n = |V|
- e = |E|
- $\sigma_{i,j}$ is the number of proper paths of all lengths between a pair of vertices (v_i, v_j)

As this calculation is based on the number of proper paths in a graph, one can easily imagine how the complexity measure grows exponentially as the graph grows in complexity. Especially highly connected parts in a graph can significantly increase the number of possible paths between two nodes, even if they do not consist of many nodes. I ran the calculations for all search trails of the second user study (cf. Appendix A3), and as the search trails grew larger, the calculations took longer as well. In both phases, I registered three calculations which were stopped after 30 hours of calculation without returning a result. These calculations were triggered by the most complex search trails. For all other search trails, the calculation returned complexity values between approximately 105 and more than 7 billion. In order to correlate these values with the grades of both the reports and the search trails, I logarithmized these values to the basis ten. I manually assigned maximum values to the search trails where the calculations have been interrupted.

I analyzed the Pearson correlation between the search trail complexity and both report grades and the search trail grades. For the first phase of the second user study, the report grade does not correlate with the search trail complexity (-0.091). For the second phase of the study, the report grade correlates with the search trail complexity (-0.566, significant on a 5% level). The correlation of the search trails' complexity with the report grades confirms the results from the large correlation table (cf. Table 57 in Chapter 5.2.5.4), where most search trail characteristics show a negative correlation with the report grade.

During the first phase of the study, the search trail grade correlates with the search trail complexity (-0.869, significant on a 5% level). For the second phase, this correlation can also be found, but a bit weaker (-0.716, significant on a 5% level). The high correlation between the search trail grade and search trail complexity is interesting. These results show that a visual evaluation of the search trail comes very close to the results of a mathematical measure of complexity. This underlines the value of search trails for the evaluation of search results and shows that visual inspection by experienced evaluators can come close to hour-long computations.

Usability of SearchTrails

To examine the user experience of an interactive product, I used the User Experience Questionnaire (UEQ), similar to the first study (cf. Chapter 5.1.4). The UEQ consists of 26 pairs of adjectives, such as 'annoying/enjoyable', 'creative/dull', or 'organized/cluttered'. When filling out the UEQ, the participants are requested to spontaneously express their opinion where the evaluated product can be located on a 7-point Likert scale between the given adjective pairs. The values for the 26 adjective pairs are later grouped into the six categories of 'Attractiveness', 'Perspicuity', 'Efficiency', 'Dependability', 'Stimulation', and 'Novelty'. The UEQ offers a standardized Excel-sheet that incorporates the viable statistical analyses for the entered questionnaire results. These analyses include the computation of the average values, standard deviances, confidence intervals, and Cronbach's alpha coefficient for all six scales, and a benchmark comparison of the evaluated product with the values from 163 studies with 4818 participants.

The last evaluation session of the second user study included filling out the UEQ. Just like our self-developed questionnaire from the final evaluation, we received 26 complete UEQs. I evaluated these questionnaires in two ways: once for all users as one group and once split into the groups SR and NT, to find differences between the two groups.

Figure 92 shows the resulting UEQ values when all participants' results are treated as one group. The black lines show the confidence intervals on a 5% error level. It is obvious that no result is negative, and all results except the score for 'Dependability' are higher than 1.00. The consistency of a scale is considered to be satisfying when Cronbach's alpha value is larger than 0.7 (cf. the discussion of this value in Chapter 5.1.4). Table 60 shows Cronbach's alpha values for all scales. Except for 'Novelty', all values are satisfying.

Table 60: Overview of Cronbach's Alpha values for each of the six properties from the UEQ for all participants of the second user study.

Attractiveness	Perspicuity	Efficiency	Dependability	Stimulation	Novelty
0.91	0.85	0.71	0.76	0.75	0.59

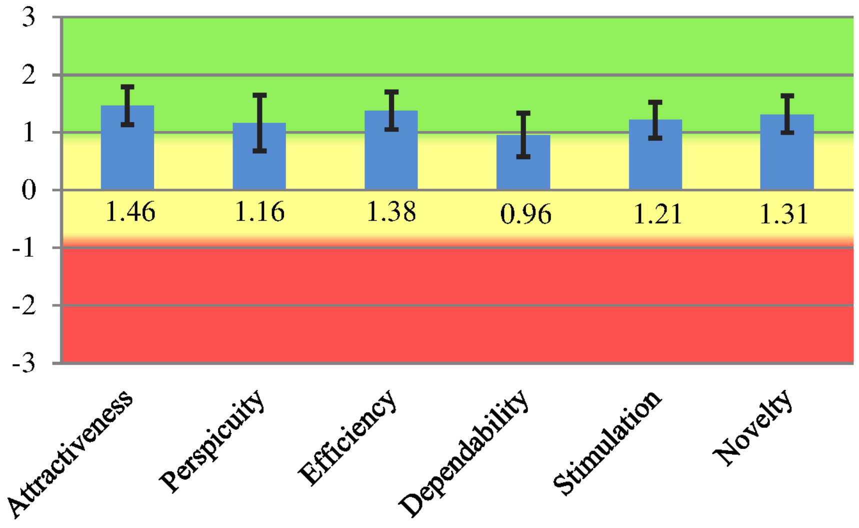


Figure 92: Results of the UEQ evaluation for SearchTrails in the second user study for all participants.

The results for all users as one group already seem to be promising with regards to the overall user experience of SearchTrails. Only the value for 'Dependability' is slightly weaker than the other values, which can be explained by the choice of a force-directed graph for the layout of the search trail, which leaves less room for manual design of the search trail. These values are resembled in the benchmark evaluation (Figure 93). In all categories, SearchTrails scores at least 'Above average'. The only exception is 'Dependability', where it scores 'Below average'. The benchmark visualization shows by its colored scales that a system needs far higher values to score 'Good' in 'Dependability' than in 'Novelty', which means that SearchTrails does not score bad, only the comparison with other systems is less successful.

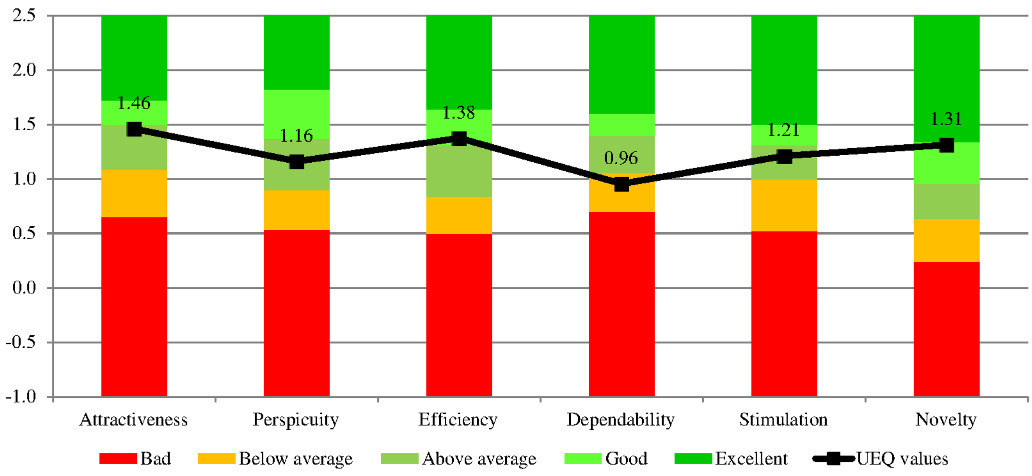


Figure 93: Benchmark comparison for SearchTrails from the second user study for all participants.

When splitting up the results based on the type of artifact during the second phase of the evaluation, the results show some differences. Figure 94 shows both the values for group SR, whose members received a report and for group NT, whose members received a search trail. These results show that all values are better for group NT than they are for group SR. Most significant are the differences for ‘Perspicuity’, where the value is 0.80 for group SR and 1.58 for group NT and for ‘Dependability’, where the value is 0.72 for group SR and 1.23 for group NT.

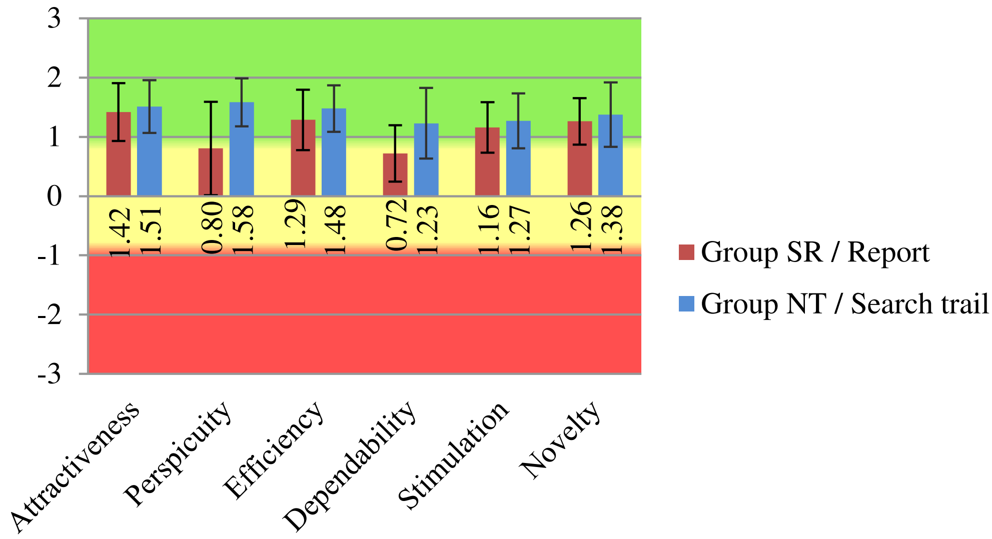


Figure 94: Results of the UEQ evaluation for SearchTrails in the second user study; split by the respective user groups.

Table 61 shows the values for Cronbach’s alpha for both groups and all scales. For group SR, the values of ‘Dependability’ and ‘Novelty’ are below 0.7, where the value for ‘Dependability’ is 0.68, which can still be considered satisfying. The value for ‘Novelty’ is not satisfying (0.47). For group NT, only the value for ‘Efficiency’ is not satisfying (0.60). This means that the according values for both groups are not reliable.

Table 61: Overview of Cronbach’s Alpha values for each of the six properties from the UEQ of the second user study; split by the respective user groups.

	Attractiveness	Perspicuity	Efficiency	Dependability	Stimulation	Novelty
Group SR	0.93	0.87	0.77	0.68	0.79	0.47
Group NT	0.90	0.71	0.60	0.82	0.74	0.71

The benchmark comparison in Figure 95 visualizes the differences between both groups and sets them into relation. Group NT generally scores very positive, with at least ‘Above average’ values, while group SR generally scores lower. For the scales ‘Attractiveness’, ‘Efficiency’, ‘Stimulation’, and ‘Novelty’, the values for both groups are close to each other, and they are very positive. For group NT, these four scales score at least ‘Good’. This means that 10% of all systems in the benchmark comparison score better, but 75% not as good. The value for ‘Novelty’ is ‘Excellent’, meaning that SearchTrails is among the best 10% of all evaluated products. For group SR, the value for these four scales is at least ‘Above average’, which means that 25% of all systems score better and 50% not as good, while the value for ‘Novelty’ is ‘Good’.

Large differences occur for the scales 'Perspicuity' and 'Dependability'. While group NT scores 'Good' for 'Perspicuity', group SR scores 'Below average'. This difference can be explained by the participants of group SR seeing the search trail as a somehow obscure way of visualizing their own search behavior, while it is very clear for the participants from group NT how the search trail resembles their colleagues' work. Similarly, group NT scores 'Above average' for 'Dependability', while group SR scores 'Below average'. For participants from group SR, the behavior of the search trail may seem not very controllable, while for the participants from group NT the force-directed layout is appreciated as a type of flexible visualization that does not need to be laid out by hand, but leaves flexibility to the user and opens room for information discovery.

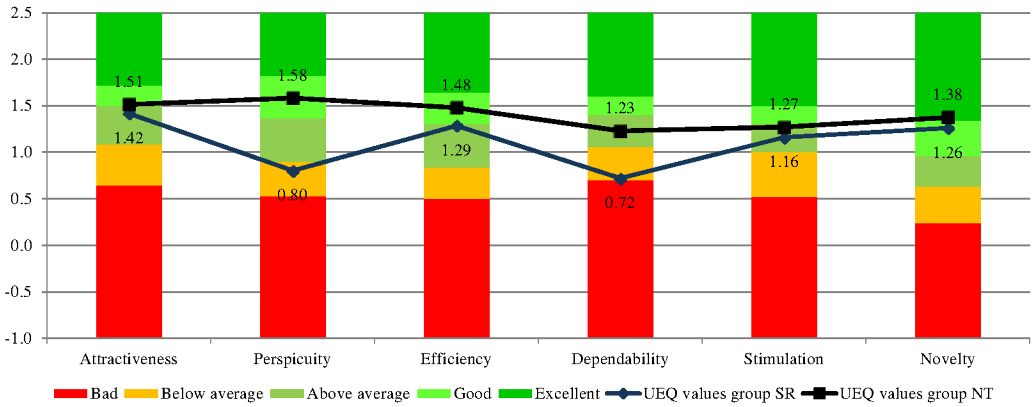


Figure 95: Benchmark comparison for SearchTrails from the second user study; split by the respective user groups.

From the UEQ results shown above, I deduce that the selected use case has impact on the user experience with SearchTrails. This is because group NT had less time to work with SearchTrails, but scored better in the UEQ evaluation than group SR. A given search trail seems to be a reasonable way to explore a collection of data, while the appreciation of a search trail as a visualization of a user's own search may need more training. These results provide more proof on the possible impact of search trails as collaboration artifacts and for answering research question RQ2.3: 'Does visualizing the course of a web search help with understanding what other users have done during a search?'.

Exam results

The second user study of SearchTrails was a subsidiary part of the lecture on Computer Supported Collaborative Work (CSCW). One week after the final evaluation, the exam on the CSCW lecture was written. This exam was a mandatory exam for the study participants; such that all study participants participated in the exam. We asked the participants for all features of SearchTrails they remember in the form of a free-text question to find out which features were remembered best. The number of mentions of each feature is shown in Table 62.

Table 62: Features of SearchTrails as being named in the exam, sorted by the number of mentions.

Feature	Number of mentions
Search trail visualization	22
Keywords	22
Highlights	19
Closing and opening clusters	16
Sharing of trails between collaborators	9
Deleting nodes	9
Sessions / Interrupting own searches	9
Important / unimportant nodes	5
Interaction with web links	2

The facts that the search trail visualization is one of the most mentioned features is not very surprising. What is more remarkable is that most participants could remember the keyword list and its interaction possibilities. Two times, participants perceived the keyword list as a tagging-mechanism for pages. Relating this to the feature evaluation, it seems that the keywords were more important for the users than I deduce from the results on discovery from the intermediate evaluation (cf. the results on hypothesis H2.1in Chapter 5.2.5.3). The high values for the high-lights and for the interaction with clusters relate well to the importance of the features and their visibility within SearchTrails. It is remarkable that altogether 18 users remembered the use of the remote search trail storage, may it be for sharing results or for storing and continuing an own search trail. This underlines the importance of this feature for the actual work with Search-Trails. The low value for the interaction with web links can be explained by the function being so obvious and not SearchTrails specific such that participants did not perceive it as a special feature of SearchTrails.

Qualitative results from the questionnaires

The questionnaires of the second user study contained several questions that encouraged the participants to express their thoughts in the form of free texts. By far not all participants made use of this chance, and as such, the results of these questions can only have anecdotal character. Nevertheless, I want to take the chance to present some interesting insights from these questions.

After the first phase of the study, the participants were asked for general comments or re-marks on the first phase of the evaluation (question M). Most results to this question reveal a generally positive opinion about SearchTrails and show some impact on the self-awareness dur-ing the search tasks. Some participants mention problems which occurred during the study. One of the problems is that the keyword list shows irrelevant keywords, but still the participants like the feature.

- (M) *'The topic was really interesting and using SearchTrails made me more motivated. The only bad part was that the SearchTrails graph was blinking sometimes and its size was increasing and decreasing and I could not control it.'*
- (M) *'The graph is very attractive. And the keywords ranking is also impressive. But what ranks No.1 is 'account'. I don't know why, I did not search it. I think google shows my account automatically.'*

- (M) *'I moved too many times back and forth from a web page, that's why my graph became like a flower and it wasn't easy to distinguish nodes. Maybe a list of search trails would be easier than Ids. I was not aware of shortcuts '-' and 'alt' key.'*

The final questionnaire contained more free-text questions, which deal for example with evaluating the given search trail (question Q12), self-awareness (question T5), and discovery (question W5). Considering the evaluation of a given search trail, participants explained how they made sense of the given artifacts. This was done for example by evaluating the titles of the nodes in the search trail, evaluating the keywords, or evaluating the highlights. It seems that no predominant strategy emerged during this phase, which may be due to the limited number of participants that received a search trail and actually worked with it, but also because the participants worked with SearchTrails for the first time.

- (Q12) *'Check the former trails and chose the useful ones according to title. Use keywords from right bar.'*
- (Q12) *'I was looking at the graph for links that seemed to be relevant. I was also choosing some keywords on the right side to do some searching.'*
- (Q12) *'I went to the highlighted search trail node and looked into that link and then, I searched further with the topic as that highlight was helpful.'*

Considering self-awareness, the participants were asked to list some insights they gained during monitoring their own search behavior (T5). Similar to the third answer from question M, one participant reported about a rather centralized approach of searching (*'A centralized website was shown from where the search originated. Visualization helped to improve search'*). Another participant reported about a steadily focusing process (*'The information in search became more focused.'*). When being asked about the support for discovery, one participant noted that SearchTrails helps recovering previously ignored information (W5, *'Ignored some info initially but was able to find it later.'*).

Question Z asked the participants for general comments or remarks on the second phase of the evaluation. Some of the participants' answers exactly expressed what SearchTrails was intended for. One participant from group SR, which was given a report, liked working with SearchTrails and asked for a search trail to work from (question Z). Some participants were convinced that search trails as collaboration artifacts can help saving time, especially when dealing with complex topics (question Z). In question 3, one participant pointed out that SearchTrails may be less important for simple daily searches. What I did not expect was that search trails can be perceived by the collaborator as some sort of intrusion into another user's privacy, as one participant expressed that it felt weird to dig so deep into another user's search results (question Z). However, this still shows the power of exchanging unfiltered search trails instead of high-level results like bookmarks. Another participant also confirmed our suspicion that SearchTrails required some experience and training to handle its features efficiently.

- (Z) *'It was nice to continue on a report, but maybe it would also be interesting to have the search trail.'*
- (Z) *'The trails and nodes really help and save time. But I will only use it while dealing with really complex big topic.'*
- (Z) *'This helped me to reduce time if working collaboratively.'*

- (3) *'Innovative and interesting. Helps when dealing with big topic. Not easy to use in daily searching.'*
- (Z) *'The whole concept was really interesting. The feeling that I was using somebody else's search was weird. But I believe what when a user gets familiar with it, it can be really helpful.'*

Although these findings can only serve as anecdotal evidence, they provide valuable insights into the users' approaches and single users' motivations or problems. Some participants seem to have understood the underlying concepts of SearchTrails very well and were able to appreciate the strengths of the concept. Adding to that, I received e-mails from participants of the first user study, asking for further developments of SearchTrails. I was content to be able to supply them with the now evaluated version, and SearchTrails is used for supporting these users' private complex search tasks since then.

More results

The data gathered during the second user study is very rich and contains lots of details. The results presented in this section contain the most interesting and significant results that could be derived from the available data. Of course it would have been possible to research more correlations between single question results and other artifacts of the study, such as UEQ results or statistical results from the search trails. I investigated more correlations or connections between question results than the ones presented here, but the results turned out to be anecdotal. Frequently the number of participants was too small or the deviations in the data were too large to end up with statistically significant results.

5.2.6 Limitations

Even a well-planned study has to face limitations when being turned into practice. All decisions were made for well-thought reasons, but some of them still carry some drawbacks with them.

Participants were not monitored during their work on the search tasks. So the participants could not be forced to split their work into several sessions, which would have helped to find out if SearchTrails helps a single user to get back into the cognitive context of a search process. The results concerning the splitting of search tasks into several sessions are therefore statistically not significant. This limitation complicates answering research question RQ2.2: 'Does SearchTrails help users with continuing web searches?' for individual users. This lack of data also complicates answering the question if SearchTrails can prevent collaborators from running into the same dead ends as their co-workers. Further studies are necessary to determine the value of own search trails after certain periods of time.

Even though the user study aimed at potential users of the system, it might not have been fully representative, as the participation in the study and the complex search tasks themselves may have been perceived as a lecture assignment. Therefore, the participation in the study concurred with other lecture assignments. As it is natural for students to try to receive optimal grades with minimal effort, it can be assumed that the participants did not invest maximal effort in the tasks, resulting in less exploration of the functions of SearchTrails, and more in a possibly fast completion of the tasks. This may result in lower usage of features like highlights and keyword search than in the first user study, where the participants were not in some sort of an assignment-situation. The previous findings confirm some differences between the user groups from the two SearchTrails user studies. The participants of the first user study came from a more professional domain, had experience with complex search tasks, and were introduced into SearchTrails thoroughly, which may have led to more intense search processes than in the second user study.

The differences between the personal background of the participants in the first and the second user study may also have influenced the study results. While the participants from the second user study were not trained to long-running search tasks (as they stated that one of their complex search tasks normally does not run longer than 30 minutes, cf. Chapter 5.2.5.1), the participants from the first study were used to such situations. The citation from the first phase (cf. Chapter 5.1.4) illustrates for the differences between the participant groups in the first and second study: *'But when the search requires exploration of multiple links, it becomes difficult to trace the path of your search'*. This may explain why features like the highlighting function were used more intensely by the participants of the first study.

Designing the questionnaires took several months of preparation, and all questions were evaluated by me, my two supervisors, and another external professor with thorough experience in user studies. Unfortunately, at least one statement turned out to be misleading. Regarding the questionnaire results, I expected more statistical differences between the two user groups SR and NT. It turned out that the difference between the average values of only 11 of 135 Likert-scaled questions are statistically significant between the two groups on up to a 10% significance level. I suppose that this is mainly due to using 5-point Likert-scales in the questionnaires. Taking into account a bias of the participants not to tend to extreme values, this turns out to make statistical analyses very hard, as the participants effectively use less than the five given options of the Likert scale. This narrow corridor of answers requires a larger sample to result in statistical significances.

The larger a study design grows, the more potential influence factors exist. Although I could show that the search topic had no influence on the study results, the results may have been influenced by other details of the study design. The very positive results for group NT in the second phase therefore may have been influenced by the well-fitting use case. It remains open how the results would have developed if this group had the chance to collect some more experience with SearchTrails. Considering the influence of training on the study results of group SR, the results for group NT could even have improved with more training.

Interviews with participants after the study revealed more potential influence factors. Although the study was planned to take place before the actual exam phase of the semester started, it somehow conflicted with some deadlines of assignments for other courses which were due some weeks before the final assignments. This could be a reason for the decreasing number of participants for in second phase. One participant reported being afraid not to be able to do the whole search task in one session and therefore decided not to do it at all. This is also a sign that some participants did not read the given instructions properly, in which splitting up search tasks into sessions was explained. One other participant reported about the own search behavior as being frequently distracted, with hardly being able to search for one topic for more than five to ten minutes before being distracted by other search tasks. This participant decided not to use SearchTrails at all, as splitting up the search at every distraction seemed unnatural for the participant.

5.3 Conclusion and outlook

The previous two subchapters provide an overview of the two user studies I performed with SearchTrails. Each of the two studies answers a set of hypotheses. The description of the study design, its implementation, and its evaluation clarifies the evaluation process.

The results of the first study with 7 participants confirm the validity of the concept of SearchTrails on a qualitative level. For this user study, I develop four hypotheses, which are confirmed. First, there is a problem with the lack of support for complex search tasks. Second, building search trails is a reasonable way of providing tool support for complex search. This helps confirming the third hypothesis that SearchTrails is an effective tool for capturing important search results. The fourth hypothesis claims that search trails are an effective approach for the evaluation of the searchers' interactions with the Internet and can also be confirmed.

Based on the results from the first study, I improved SearchTrails and extended it by features enabling collaborative search. The features include the recreation of search trails, the implementation of a remote logging mechanism, and some features for more direct interaction with the search trail.

My second user study with 29 participants provides more quantitative results. Being performed as a subsidiary part of a university lecture, it enables the confirmation of five study hypotheses. First, I show that SearchTrails can support complex search by supporting aggregation, synthesis, and discovery. Second, SearchTrails raises the searchers' awareness for their own behavior during search process, which leads to more controlled and reflected search processes. Third, I show that using SearchTrails improves the quality of the search process. Fourth, technical reasons allow confirming that SearchTrails enables asynchronous collaborative search by exchanging and interaction with search trails. Last, I can confirm that the exchange of search trails can significantly improve the quality of search results.

Additional findings provide interesting insights into the time needed for the search tasks and show that search trails are a means for avoiding redundant search efforts. These results also show discrepancies between the self-awareness of participants and the actual results as well as learning effects during the study. Further questions reveal that SearchTrails can be considered a lightweight tool, and the evaluation of search trails shows that visual inspection of the search trails is almost as powerful as hour-long computations of graph complexity.

While some hypotheses can be confirmed with more statistical significance than others, the overall result is very positive. The approach of SearchTrails is confirmed to be a sound concept for supporting asynchronous collaborative search, and has a positive influence on both single and collaborative search. The summarizing table (cf. Table 63) shows an overview of the hypotheses from both studies and their verification.

Table 63: Overview of the study hypotheses for both SearchTrails user studies and their verification.

	Hypothesis	Title	Verification
1st user study	H1.1	The support of complex search tasks by standard search engines is insufficient.	Confirmed by the questionnaire results. Participants use the limited capabilities of search engines for their search tasks, but criticize missing long-time support and a lack of context.
	H1.2	Creating a search trail is a reasonable way for providing tool support for complex search tasks.	Confirmed by the questionnaire results. The search trail visualization provides a way to get back to lost results and structures the browsing.
	H1.3	Creating a search trail is a reasonable way for providing tool support for complex search tasks.	Confirmed by the questionnaire results. Many features of SearchTrails support capturing search results, such as highlights or clustering.
	H1.4	The search trails are an effective approach for an evaluator to assess and evaluate the searchers' interactions with the Internet.	Confirmed during a workshop with me and Ulrich Norbisrath. Visual evaluation of SearchTrails is very easy.
2nd user study	H2.1	SearchTrails eases single user complex search tasks by supporting the threefold of aggregation, synthesis, and discovery.	Confirmed by questionnaire results. Good support for aggregation and synthesis, less support for discovery.
	H2.2	SearchTrails raises awareness for the own personal search behavior.	Confirmed by the questionnaire results. The awareness strongly increases when using SearchTrails.
	H2.3	The use of SearchTrails improves the quality of the search process for complex search tasks.	Confirmation by questionnaire results, but the given hints suggest that the hypothesis is true.
	H2.4	SearchTrails enables asynchronous collaborative search by exchanging search trails.	Confirmed by questionnaire results for group NT, even if not statistically significant.
	H2.5	Exchanging search trails in collaborative search improves the quality of search results	Confirmed by analysis of the search trails and the reports. Group NT scored significantly better.

This overview of confirmed hypotheses enables answering the second group of research questions (**RQ2, 'Value'**), as developed in Chapter 1.4.

- **RQ2.1: 'Does SearchTrails help users coping with complex search tasks?'** can be answered positively by the results on hypothesis H2.1 in Chapter 5.2.5.3, as SearchTrails helps users with the aggregation, discovery, and synthesis tasks of complex search.
- **RQ2.2: 'Does SearchTrails help users with continuing web searches?'** could be answered for the continuation of other user's search trails. The results on hypothesis H2.4 indicate that the exchanged artifacts allow users to continue the search sessions of other users. The value of own search trails after a certain period of time was not measured (cf. Chapter 5.2.6).

- **RQ2.3: 'Does visualizing the course of a web search help with understanding what other users have done during a search?'** can be answered positively by the results on hypothesis H2.4 and the results on the 'Usability of SearchTrails' in Chapter 5.2.5.4, as published in (Franken et al., 2015).
- **RQ2.4: 'Does SearchTrails ease collaboration between users by exchanging web searches?'** can be answered positively by the results on hypothesis H2.5. Especially the impact of search trails as collaboration artifacts on the grades of the reports in the second phase of the study confirms that SearchTrails eases collaboration. These results are published in (Franken et al., 2016).
- **RQ2.5: 'Is SearchTrails lightweight enough to be attractive for users?'** can be answered positively by the results on 'Lightweightness' in Chapter 5.2.5.4, where users agree that SearchTrails is a lightweight system.

The next chapter provides a conclusion of the achieved work and summarizes the main research results. It presents possible areas of further work with SearchTrails that remained untouched so far and therefore leave room for future work and improvements.

Chapter 6
Main results and outlook

This is the concluding chapter of my thesis. In this chapter, I wrap up the main results of my thesis and provide an outlook on possible future work. A summary of the previous five chapters gives an overview of the research process of my thesis. In Chapter 1, I develop three scenarios where necessary support is missing. This holds for supporting individual and collaborative search processes as well as for offering support for resuming interrupted searches. These scenarios could be supported by capturing and exchanging the user's search trails of search process. I derive three challenges to be overcome with this thesis, which are capturing search results as a whole, enabling the exchange of search results between collaborators, and resuming interrupted searches. My main research question **'Can search trails provide support for complex web search and how should tool support look like?'** states that I want to develop tool-support for complex web search processes with search trails and investigate their value. I select the research methodology of design science for my thesis, for which I describe in Chapter 6.1 how I followed it. I develop, implement, and evaluate an approach which allows capturing the course of the user's web search and the exchange of search trails as collaboration artifacts

Chapter 2 presents an overview of the related work relevant to all major aspects of the thesis. The historical origins show that no approach has been made to determine the individual value of search trails for single user and collaborative search scenarios. I present the theoretical concepts which are most important for my work, which are the definition of complex search and a cyclic model of search. I point out that my approach is able to capture negative search results, a explain differences to several approaches that could be argued to help in my application cases. I analyze set of related systems for logging search processes and for supporting collaborative search, which I condense in two comparing tables. Finally, I provide an outlook on recent research approaches towards search trails and point out how my approach is different from existing research approaches.

In Chapter 3, I evaluate the findings from the historical origins as well as the related work. I evaluate the two comparing tables, from which I derive a set of abstract requirements for the envisioned system named 'SearchTrails'. Based on these requirements, I develop a set of technical features to be incorporated, such as implementing SearchTrails as a web browser extension or using force-directed graph visualization. From the abstract requirements, I develop a set of technical features, which ensure support for complex search support and the interaction with search trails. A set of user interface views helps visualizing the main screens of SearchTrails. I present a rough architecture of SearchTrails, which incorporates both the user interface and the logical background of SearchTrails.

Chapter 4 explains the technical background of SearchTrails. The final architecture of SearchTrails relies on the Google Chrome browser and makes incorporating several frameworks necessary to achieve the desired functionality. I select and integrate a number of external frameworks for establishing the functionality of SearchTrails, such as a visualization framework or a framework for keyword extraction. I developed SearchTrails during several design iterations, which converge towards the final functionality. The iterations range from early paper prototypes, a first software prototype called SemanticClipboard, and two versions of SearchTrails which I evaluate in two user studies.

In Chapter 5, I present the two user studies of SearchTrails, based on the two main technical versions of SearchTrails. I explain the study hypotheses concerning the effectivity of SearchTrails from the first user study, and the hypotheses concerning the efficiency in collaborative search scenarios from the second user study. With the help of the data generated in each SearchTrails user study, I analyze the study results and confirm the hypotheses developed for each version of SearchTrails. These results show that SearchTrails is both effective and efficient in single user and collaborative search cases and helps evaluators assessing the search results of other searchers. The results answer the main research questions of my thesis, which are wrapped up in the following Chapter 6.2.

This chapter sums up the main results achieved and finally evaluates whether the research questions can be answered positively. The rest of this chapter is divided into three subchapters. In the first subchapter, I point out how my thesis follows the research methodology of 'design science' as proposed in Chapter 1.6 of this thesis. The second subchapter discusses the results achieved in the thesis and answers the research questions. In the third subchapter, I point out possible areas of future work in the field of my thesis.

6.1 Following the methodology

This thesis follows the research methodology of design science. Furthermore, I selected to do iterative design cycles during the technical development of SearchTrails. In Chapter 1.6, I introduce both methodologies. As the methodology of design science does not define a strict process, but is more a set of seven guidelines to be followed, this subchapter evaluates for each of the guidelines whether and how it was followed (based on (Hevner et al., 2004, p.82 ff.)).

1. **Design as an artifact**
   'A research project following the design-science guidelines must produce a viable artifact, may it be an application, a model, or an instantiation.'
   With the different evaluated versions of the software prototype 'SearchTrails', the thesis produced a number of viable artifacts. These artifacts are mostly applications, running in the environment of a common web browser.

2. **Problem relevance**
   'Design science aims for developing technology based solutions for important and relevant business problems.'
   My analysis of the related work shows that the idea of creating search trails from the user's interaction with the Internet has been proposed a long time ago, but has not been realized since. Similarly, my related work shows that the support for collaborative and complex search is highly relevant (e.g. (Ringel Morris, 2013; Kelly & Payne, 2014; Awadallah et al., 2014)), but several realized approaches could not fulfill the expectations (cf. Chapter 2.2.5 and Chapter 2.5.2.1). Therefore, the solutions I develop address important and really existing problems.

3. **Design evaluation**
   'Whether a product developed in a design science process has a potential practical impact needs to be shown in a well-conducted evaluation.'
   The software prototype SearchTrails has been developed in several design iterations involving potential users of the system. The experiences during the design iterations lead to improvements in the following versions of SearchTrails. Two versions of SearchTrails have been evaluated with users. While the first study was a qualitative user study, the second study was a quantitative field study.

4. **Research contributions**
   'The design science process must provide clear and verifiable contributions of at least one of the following types: design artifact, design foundations, or design methodologies.'
   With the software prototype SearchTrails, the thesis contributes a design artifact to the set of tools for supporting discontinuous, asynchronous, collaborative, complex search. I have shown the validity and the impact of this software tool in the two user studies.

5. **Research rigor**
   'The application of rigorous methods during the construction and evaluation of the design artifact is mandatory.'
   I developed SearchTrails in an iterative design process and evaluated it in a series of user studies, using both self-developed and also well-accepted standard evaluation methods, such as User Experience Questionnaires.

6. **Design as a search process**
   'The search for an effective artifact must make use of proper methods to reach the desired needs, but must obey the rules of the problem environment.'
   SearchTrails is designed upon a thorough analysis of related work, which enables the development of requirements and technical features. Features for SearchTrails are derived from solutions and features of existing related systems. SearchTrails therefore builds upon existing systems and obeys the rules of the problem environment, but also tries to overcome weaknesses of existing solutions by the analysis of related approaches.

7. **Communication of research**
   'The results of a design science research process must be presented to the matching audiences.'
   The results of the development process have been published in four full research papers to national and international conferences: While (Franken & Norbisrath, 2014b) has been published at 'Mensch und Computer 2014' in Munich, (Franken & Norbisrath, 2014a) has been published at CASCON 2014 in Toronto, (Franken et al., 2015), has been published at 'Mensch und Computer 2015' in Stuttgart, and (Franken et al., 2016) has been published at 'Mensch und Computer 2016' in Aachen.

This subchapter recapitulates the seven guidelines of the design science research method. I confirm that I consistently followed the select research methodology throughout this thesis. The next subchapter discusses the main results of my thesis and the contribution to the existing work on supporting collaborative complex search.

6.2 Discussion of the main results

In this subchapter, I wrap up the main results of my thesis and answer the research questions raised in Chapter 1.4.

In my thesis, I develop the system 'SearchTrails', which captures the individual search process of a user during the individual complex search process and visualizes the search process as a search trail. SearchTrails supports complex search in the sense of the definition in (Singer et al., 2012a) by supporting the aggregation, discovery, and synthesis of information. SearchTrails allows capturing textual highlights on web pages, which are stored both as parts of the search trail and within a collection of highlights, and it extracts keywords from web pages to point users towards new search directions. SearchTrails allows the remote storage of its search trail data objects and makes them exchangeable between users by exchanging the respective search trail IDs. This allows the access of search trails from different browser instances at any point in time, as they can be recovered from a central server. Thus SearchTrails provides a benefit for both individual and collaborative search scenarios and bridges the gap between logging search results and providing support for collaborative search. SearchTrails achieves this by a custom made browser extension for a commonly known web browser, thus ensuring a familiar browsing experience and low effort for supporting SearchTrails in coming browser generations.

It is important to note that my thesis is the first approach to actually reveal the search trails to the users and it is also the first work to let the users decide which trails to exchange with colleagues and which not. The related approaches presented in earlier Chapters (e.g. Chapter 2.5.1.1 or Chapter 2.5.2.1) only capture the user's search trails and use them for their own calculations, i.e. to be able to present generalized recommendations to other unknown users. Even if some approaches already use search trails for analysis, these approaches use the search trails only as an additional type of data to extract knowledge from. None of these approaches has pointed out that there is individual value in search trails. Furthermore, none of the approaches presents the actual search trails to the users who created them, or to other users to make sense of them and therefore fail exploiting the power of individual search results. Logically, so far no efforts have been taken to show the value of individual trails for collaboration.

SearchTrails values the individual search results of users and enables the exchange of individual, unfiltered search artifacts between collaborating users. SearchTrails therefore provides an approach to overcome the filter bubble (as defined in (Pariser, 2011)) by valuing individual search aspects and efforts. It leaves the user the freedom to discover and exchange individual search results. This way, mechanic recommender systems are overcome and individual search paths and recommendations receive the collaborator's full attention. My approach keeps all individual characteristics of a search process, enables the searcher to interact with it, and enables the full individual search trail to be exchanged between users. As search trails become larger and larger, the evaluation of the trails in all their details may not even be that important anymore, but the overview of a search trail conveys much information about the searcher's interaction with the Internet.

At the beginning of this chapter, I give an overview of the most important contents of all previous chapters. I now recall the different challenges and research questions I raise throughout my thesis and answer them based on the achievements of this thesis. I explain how my approach helps overcoming the three challenges from the motivating scenarios and address the partial research questions from Chapter 1.4. Answering these partial research questions helps answering the overall research question ‘Can search trails provide support for complex web search and how should tool support look like?’.

In Chapter 1.2, I describe three technical challenges emerging from the motivating scenarios. The **first scenario ‘Capturing a search process’** raises the first challenge of capturing information within its context, which goes beyond plain bookmarks. This challenge is addressed and solved by SearchTrails, as SearchTrails allows capturing interesting highlights within the context of their discovery, i.e. by storing the addresses of web pages visited before and after the visit of an interesting web page, and by providing a condensed highlight overview.

The **second scenario ‘Exchanging search results’** raises the challenge of enabling the exchange of search results between known or unknown collaborators to achieve asynchronous collaboration. This challenge is addressed and solved by SearchTrails by the remote storage of the search trail data objects with the help of a unique identifier. This identifier may be shared via e-mail between known collaborators, but could also be shared in forums with unknown collaborators. This way, SearchTrails solves the second challenge and makes unfiltered search results exchangeable between users.

The **third scenario ‘Continuing a search’** raises the challenge of resuming interrupted searches and enabling the updating of information on a certain topic. This challenge is addressed and solved by SearchTrails by the remote storage of the search trail data objects. SearchTrails users have the chance to store their search trails remotely and to retrieve them by their identifier at any point in time and from any location. This way, catching up with previous search processes and updating information can be achieved by re-evaluating one’s own search trail.

This short overview shows that the final version of SearchTrails is able to address and overcome the challenges raised in the motivating scenarios. SearchTrails is therefore able to provide practical help in everyday search situations and goes beyond existing solutions.

From the goals of my thesis in Chapter 1.3, I develop the overall research question in Chapter 1.4: **‘Can search trails provide support for complex web search and how should tool support look like?’** This research question is broken down into two groups of research questions. The first group (RQ1 ‘Feasibility’) deals with the possibility of offering support for complex web search tasks by constructing search trails. The second group (RQ2 ‘Value’) deals with value of search trails in individual and collaborative search scenarios and their impact on the quality of the search process. All research questions have been addressed in the respective chapters of this thesis. The following summary gives an overview of the results.

The first group of research questions (**RQ1, ‘Feasibility’**) consists of the following questions, which are mainly answered in the chapters on the first qualitative user study described in Chapter 5.1.4.

The research question **RQ1.1: ‘Are search trails a means to resemble the course of a user’s web search?’** can be answered positively by the historical origins of search trails in Chapter 2.1, the recent trends towards search trails in Chapter 2.7, and the technical requirements for the SearchTrails system in Chapter 3.2. RQ1.1 could finally be answered positively in Chapter 5.1.4 by the results for hypothesis H1.4, as the search trails are an effective approach for an evaluator to assess and evaluate the searcher's interaction with the Internet. Therefore, search trails are a means to resemble the course of a user's web search.

The research question **RQ1.2: 'Which functions does a system need to offer to the users to support capturing both the context as well as key findings of a web search?'** can be answered positively by findings in the comparing tables in Chapter 2.5.1.2 and Chapter 2.5.2.2, which I evaluate in Chapter 3.1, and derive detailed technical requirements for SearchTrails in Chapter 3.2. In Chapter 4.3.3 and Chapter 4.3.4, I define the set of functions for both evaluated versions of SearchTrails based on the requirements. RQ1.2 can finally be answered by the results on hypothesis H1.3 in Chapter 5.1.4, which shows that SearchTrails is an effective tool for capturing important search results within their context.

The research question **RQ1.3: 'Does a system with the functions from RQ1.2 provide help and support for users during complex search processes?'** can be answered positively by the results for hypothesis H1.2 in Chapter 5.1.4. The results show that SearchTrails is an effective tool for providing tool support during complex search processes.

The first group of research questions RQ1 'Feasibility' qualitatively confirms that the approach of SearchTrails is an effective way for resembling the course of a complex web search process (RQ1.1 'Search trails valid?'), and that the set of functions implemented in SearchTrails is sufficient for supporting complex web search processes (RQ1.2 'Which functions?'). Additionally, RQ1.3 'Provide help?' confirms that SearchTrails is furthermore an effective way of supporting users performing complex web search.

The second group of research questions (**RQ2, 'Value'**) is mainly answered by the results of the second user study, as described in Chapter 5.2.5. Research question **RQ2.1: 'Does SearchTrails help users coping with complex search tasks?'** is answered positively by the results on hypothesis H2.1 in Chapter 5.2.5.3, as SearchTrails helps users with the aggregation, discovery, and synthesis tasks of complex search and raises the participants' efficiency and motivation.

The research question **RQ2.2: 'Does SearchTrails help users with continuing web searches?'** is not answered clearly, as it can be split into two cases: The results on hypothesis H2.4 show that RQ2.2 can be answered positively for the continuation of unknown web searches from collaborators (cf. Chapter 5.2.5.3 on hypothesis H2.4). The exchanged artifacts allow the users to continue the search sessions of other users. However, for the continuation of own web searches, no statistically significant results could be achieved because the participants were not forced to work in multiple sessions. Hints pointing towards answering RQ2.2 positively can be deduced from the results on hypothesis H2.5 (cf. Chapter 5.2.5.3). If it is easily possible for users to evaluate a given, but unknown search trail, it should be even easier to evaluate an own search trail even after some time has passed.

Research question **RQ2.3: 'Does visualizing the course of a web search help with understanding what other users have done during a search?'** is answered positively by the results on hypothesis H2.4 'SearchTrails enables asynchronous collaborative search by exchanging search trails' (cf. Chapter 5.2.5.3) and the results on the 'Usability of SearchTrails' in Chapter 5.2.5.4. These results have been published in (Franken et al., 2015).

The research question **RQ2.4: 'Does SearchTrails ease collaboration between users by exchanging web searches?'** is answered positively by the results on hypothesis H2.5 'Exchanging search trails in collaborative search improves the quality of search results' (cf. Chapter 5.2.5.3). Especially the impact of search trails as collaboration artifacts on the grades of the reports in the second phase of the study confirms that SearchTrails eases collaboration between users and has a very positive impact on the quality of the search results. These results have been published in (Franken et al., 2016).

Research question **RQ2.5: 'Is SearchTrails lightweight enough to be attractive for users?'** is answered positively by the results from the questionnaire on 'Lightweightness' as described in Chapter 5.2.5.4, where users agree that SearchTrails is a lightweight system. The results of (Ringel Morris, 2013) presented in Chapter 2.2.5 suggest that being lightweight is a vital acceptance criterion for a collaborative search support system.

The second group of research questions RQ2 'Value' focuses more on a quantitative evaluation and aspects of collaborative complex web search. RQ2.1 'Ease complex search?' shows that SearchTrails eases complex web search for individual users and that it may help with continuing web searches, at least in collaborative search scenarios (RQ2.2 'Ease continuation?'). The visualization of the search trails plays an important role for understanding what other users did during their search processes (RQ2.3 'Ease sensemaking?'), and SearchTrails eases collaboration between users (RQ2.4 'Ease collaboration?'). Finally, the overall approach of SearchTrails is lightweight enough to be attractive for users (RQ2.5 'Lightweight').

Taking these findings together, the overall research question **'Can search trails provide support for complex web search and how should tool support look like?' can be answered positively with the system SearchTrails**. While the first group of research questions confirms that SearchTrails is a means for supporting individual complex search processes, the second group of research questions confirms that SearchTrails also supports and improves collaborative search processes. The considered collaborative search processes are asynchronous and may be discontinuous, and focus on complex search processes.

It is possible to provide support for complex web search by building search trails, and the developed system SearchTrails does exactly that.

Looking back at the three scenarios described in Chapter 1.1, it is obvious that SearchTrails enables user support in all three scenarios. In the first scenario 'Capturing a search process', SearchTrails would enable the searcher to capture and save the complex search process and to store all relevant findings during the search process. In the second scenario 'Exchanging search results', SearchTrails would enable the two colleagues to simply exchange a search trail ID of a search trail that one colleague already created. The colleague would be able to dive into the previous searcher's results. In the third scenario 'Continuing a search', an ongoing search would be automatically stored by SearchTrails and could be resumed after an interruption, at another time or another place.

This thesis has so far been published in four full paper conference publications at national and international conferences. The main qualitative results on the general approach of building search trails during complex search processes from the first user study have been published in (Franken & Norbisrath, 2014b). The results on visual evaluation of given search trails from the first user study have been published in (Franken & Norbisrath, 2014a). The results on the value search trails for collaborative search and their impact on the usability of SearchTrails have been published in (Franken et al., 2015). The results on the value of search trails as collaboration artifacts for improving the quality of collaborative search results have been published in (Franken et al., 2016).

This subchapter provides an overview of the results achieved in this thesis. These results show that the overall research question can be answered positively by the results on the research questions from the two research question groups. The developed system SearchTrails provides practical help in the described scenarios and SearchTrails addresses the challenges that arise from the scenarios. Therefore, the claim raised by the title of my thesis 'Supporting asynchronous, discontinuous, collaborative, complex search tasks by the visualization of search trails' can be fully satisfied.

6.3 Outlook

While the previous subchapters summarize and conclude the most important results of my thesis, this chapter provides an outlook on possible work building upon it.

Both the visualization of individual search trails for users and the utilization of individual search trails as collaboration artifacts are new and have not been done before. The thesis introduces search trails for these purposes, and the first results are very promising. The novelty of the selected approach implies that the different evaluations of the system SearchTrails need to show the technical feasibility, the effectiveness, and the efficiency of the developed approach. This has been achieved during the two user studies. However, further questions remain, which could be solved in future studies or with future improvements of SearchTrails.

Some questions which could not be answered here have been raised in earlier chapters of the thesis. This holds especially for the individual case of research question RQ2.2: 'Does SearchTrails help users with continuing web searches?' where the impact of the search trail visualization on the continuation of own web searches is not clear and could be solved by further investigation of the continuation of own search trails after a period of time.

Similarly, the question whether search trails can prevent users from running into the same dead ends like their collaborators cannot be answered due to a lack of data. However, the question was not one of the main focuses of my thesis. The question points towards further investigation of the way users interact with the search trail. Supervising users interacting with own or given unknown search trails could help finding out which strategies are used for evaluating given search trails and how information is extracted from them. So far, no predominant strategy is reported by the searchers (cf. Chapter 5.2.5.4 'Qualitative results from the questionnaires').

The two user studies do not explicitly focus on the impact of time on the value of the search trails. For own search trails, it would be interesting to find out how easily users can evaluate their own search trails after certain different periods of time. Adding to this, it would also be interesting to find out how users cope with information that has relocated or decayed after a period of time, such as newspaper articles which are not recent anymore, or forum entries that have move to different pages. Due to the highlight function of SearchTrails, the effect of decaying information could be less important compared to plain bookmarking tools, as the piece of information itself may still be present in the highlight overview, and only the source is not accurate anymore.

Further work could also be done with regards to technical improvements of SearchTrails. This includes both the addition of new features, but also the investigation of how users interact with them. New features could be the integration of different resources into the search trails or the highlights, such as thumbnail pictures, videos, or PDFs. It could also be possible to connect trails, or to integrate different trails on similar topics, which would be even closer to the scenarios described by Vannevar Bush. Another point could be enabling synchronous collaboration with SearchTrails. While potential conflicts of multiple users accessing the same search trail need to be solved manually now, a technique could be developed for avoiding these conflicts. Despite the technical implementation work, the interaction of users with the described new features would be interesting. Especially in the case of thumbnail pictures, their influence on the usability of SearchTrails would be interesting.

The problem of discovering, aggregating, and synthesizing information from different sources will stay important in the medium term. This is simply a matter of the business model of search engines. For now, search engines generate revenue from placing ads on search engine result pages, or from website owners for generating traffic on web pages listed high in the search engine's result ranking. These pages again generate revenue from advertisements or directly from the user. Therefore, search engines cannot deliver too detailed answers themselves, as this would result in withholding traffic from search engine customers. At the same time, web pages would be degraded to be the free deliverers of content (cf. (Lewandowski, 2015, p.287 ff.)). For these reasons, search engines as they are modeled today can only be entry points into the web, from which a set of more or less valuable information is discovered and needs to be assembled to a valuable result by the searcher.

In my thesis, I make an impact on facilitating complex search processes in single user and collaborative complex search scenarios by providing support during complex web search tasks by building search trails. I develop the software prototype SearchTrails, which transforms the user's interaction with the Web into a dynamic search trail, which is stored remotely and can be continued by any person at any time. My two user studies show that my approach is both effective and efficient. SearchTrails is effective, as a first qualitative user study shows that it provides a way for visualizing the user's web search process, and eases complex search by supporting aggregation, discovery, and synthesis. A second and quantitative user study shows that SearchTrails even supports collaborative search scenarios and makes the searchers more efficient. It does this by allowing the easy continuation of previous search processes. Especially with the findings on collaborative search processes, this thesis also makes an impact on the research on computer supported collaborative work.

References

Ahmmari, A., Lau, L. & Dimitrova, V. (2012). Distributed Data Mining for User Sensemaking in Online Collaborative Spaces. In: *Mastering Data-Intensive Collaboration through the Synergy of Human and Machine Reasoning (dicoSyn 2012) workshop at CSCW 2012*. 2012, Seattle, Washington: ACM Press NY.

Allsopp, J. (2007). *Microformats: Empowering Your Markup for Web 2.0*. [Online]. Berkeley, CA : New York: Apress. Available from: http://link.springer.com/10.1007/978-1-4302-0195-3.

Altiok, T. & Stidham, S. (1983). The Allocation of Interstage Buffer Capacities in Production Lines. *IIE Transactions*. 15 (4). pp. 292–299.

Amershi, S. & Morris, M.R. (2008). CoSearch: a system for co-located collaborative web search. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. [Online]. 2008, ACM, pp. 1647–1656. Available from: http://dl.acm.org/citation.cfm?id=1357311. [Accessed: 18 September 2015].

Assfalg, R. (1999). Eine hypertextbasierte Autorenumgebung als Redaktionsinstrument eines elektronischen Wissenschaftsforums. In: *Textproduktion. HyperText, Text, KonText*. Frankfurt/Main: Peter Lang Verlag, pp. 181–198.

Aula, A., Jhaveri, N. & Käki, M. (2005). Information search and re-access strategies of experienced web users. In: *Proceedings of the 14th international conference on World Wide Web*. [Online]. 2005, ACM, pp. 583–592. Available from: http://dl.acm.org/citation.cfm?id=1060831. [Accessed: 21 September 2015].

Aula, A. & Russell, D.M. (2008). Complex and Exploratory Web Search. *Information Seeking Support Systems An invitational workshop sponsored by the National Science Foundation*. pp. 23–24.

Awadallah, A.H., White, R.W., Pantel, P., Dumais, S.T. & Wang, Y.-M. (2014). *Supporting Complex Search Tasks*. In: [Online]. 2014, ACM Press, pp. 829–838. Available from: http://dl.acm.org/citation.cfm?doid=2661829.2661912. [Accessed: 18 September 2015].

Bates, M.J. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online Review*. 13 (5). pp. 407–424.

Belkin, N.J., Clarke, C.L., Gao, N., Kamps, J. & Karlgren, J. (2012). Report on the SIGIR Workshop on Entertain Me: Supporting Complex Search Tasks. In: *ACM SIGIR Forum*. [Online]. 2012, ACM, pp. 51–59. Available from: http://dl.acm.org/citation.cfm?id=2093354. [Accessed: 21 September 2015].

Belkin, N.J., Dumais, S., Scholtz, J. & Wilkinson, R. (2004). *Evaluating interactive information retrieval systems: opportunities and challenges*. In: [Online]. 2004, ACM Press, p. 1594. Available from: http://portal.acm.org/citation.cfm?doid=985921.986162. [Accessed: 18 September 2015].

Berners-Lee, T., Hendler, J., Lassila, O. & others (2001). The semantic web. *Scientific American*. 284 (5). pp. 28–37.

Bhargava, R. (2009). *Manifesto for the Content Curator: The Next Big Social Media Job of the Future ?* [Online]. 2009. The Influential Marketing Blog. Available from: http://www.rohitbhargava.com/2009/09/manifesto-for-the-content-curator-the-next-big-social-media-job-of-the-future.html. [Accessed: 22 September 2015].

Bhargava, R. (2011). *The 5 Models of Content Curation*. [Online]. 2011. The Influential Marketing Blog. Available from: http://www.rohitbhargava.com/2011/03/the-5-models-of-content-curation.html. [Accessed: 22 September 2015].

Bible, Ecclesiastes 12:12 (n.d.). *Ecclesiastes 12:12*. [Online]. Bible. Ecclesiastes 12:12. Available from: http://www.bibleserver.com/text/ESV/Ecclesiastes12.

Blair, A.M. (2010a). *Information overload, then and now. The Chronicle Review*. [Online]. 28 November 2010. Information Overload, Then and Now. Available from: http://chronicle.com/article/Information-Overload-Then-and/125479/. [Accessed: 21 September 2015].

Blair, A.M. (2010b). *Too much to know: managing scholarly information before the modern age*. New Haven [Conn.]: Yale University Press.

Bookstein, A. & Cooper, W. (1976). A general mathematical model for information retrieval systems. *The Library Quarterly*. 46 (2). pp. 153–167.

Boswell, J. & Croker, J.W. (1831). *The life of Samuel Johnson, LL.D.: including A journal of a tour to the Hebrides in Five Volumes*. The Life of Samuel Johnson, LL.D.: Including A Journal of a Tour to the Hebrides, by James Boswell. [Online]. London: John Murray. Available from: https://books.google.no/books?id=L3IEAAAAYAAJ.

Breivik, P.S. (1985). A vision in the making: putting libraries back in the information society. *American Libraries*. 16 (10). pp. 723–723.

Briggs, P. & Smyth, B. (2007). *Trusted search communities*. In: [Online]. 2007, ACM Press, p. 337. Available from: http://portal.acm.org/citation.cfm?doid=1216295.1216361. [Accessed: 18 September 2015].

Broder, A. (2002). A taxonomy of web search. In: *ACM SIGIR Forum*. 2002, ACM, pp. 3–10.

Buckminster Fuller, R. & McHale, J. (1967). *World Design Science Decade, 1965-1975*. Phase I, (1965) Document 3 hevner. Souther Illinois University, Carbondale, Illinois,: World Resources Inventory.

Bush, V. (1945). *As we may think*. 176 (July 1945). pp. 101–108.

Campbell, D.J. (1988). Task Complexity: A Review and Analysis. *Academy of Management Review*. 13 (1). pp. 40–52.

Campoy Flores, F., Quint, V. & Vatton, I. (2006). Templates, microformats and structured editing. In: *Proceedings of the 2006 ACM symposium on Document engineering*. [Online]. 2006, ACM, pp. 188–197. Available from: http://dl.acm.org/citation.cfm?id=1166211. [Accessed: 18 September 2015].

Capra, R. (2009). HCI Browser: A tool for studying web search behavior. *Proceedings of the American Society for Information Science and Technology*. 47 (1). pp. 1–2.

Capra, R., Chen, A.T., Hawthorne, K. & Arguello, J. (2012). ResultsSpace: An experimental collaborative search environment. *Proceedings of the American Society for Information Science and Technology*. 49 (1). pp. 1–4.

Capra, R., Chen, A.T., McArthur, E. & Davis, N. (2013). Searcher actions and strategies in asynchronous collaborative search. *Proceedings of the American Society for Information Science and Technology*. 50 (1). pp. 1–10.

Caspi, E., Shankar, J. & Wang, J. (2004). *Reading the classics: As we may think*. [Online]. Available from: http://www.moralfiber.org/eylon/berkeley/cs294-classics/bush-lecture/bush-as-we-may-think.pdf. [Accessed: 21 September 2015].

Chi, E.H. (2009). Information Seeking Can Be Social. *Computer*. 42 (3). pp. 42–46.

Choo, C.W., Detlor, B. & Turnbull, D. (1998). A Behavioral Model of Information Seeking on the Web–Preliminary Results of a Study of How Managers and IT Specialists Use the Web. *1998 ASIS Annual Meeting*.

Claypool, M., Le, P., Wased, M. & Brown, D. (2001). Implicit interest indicators. In: *Proceedings of the 6th international conference on Intelligent user interfaces*. [Online]. 2001, ACM, pp. 33–40. Available from: http://digitalcommons.wpi.edu/computerscience-pubs/84/.

Clough, P. & Berendt, B. (2009). Report on the TrebleCLEF query log analysis workshop 2009. In: *ACM SIGIR Forum*. [Online]. 2009, ACM, pp. 71–77. Available from: http://dl.acm.org/citation.cfm?id=1670578. [Accessed: 21 September 2015].

Cronbach, L.J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*. 16 (3). pp. 297–334.

Cutrell, E. & Dumais, S.T. (2006). Exploring personal information. *Communications of the ACM*. 49 (4). p. 50.

Czerwinski, M., Van Dantzich, M., Robertson, G. & Hoffman, H. (1999). The contribution of thumbnail image, mouse-over text and spatial location memory to web page retrieval in 3D. In: *Proceedings of INTERACT*. [Online]. 1999, pp. 163–170. Available from: http://research.microsoft.com/en-us/um/people/marycz/interact99.pdf. [Accessed: 18 September 2015].

Danilov, D. & Vainikko, E. (2013). *Search excavator: the knowledge discovery tool*. In: [Online]. 2013, ACM Press, pp. 2525–2528. Available from: http://dl.acm.org/citation.cfm?doid=2505515.2508203. [Accessed: 18 September 2015].

Delgado, J., Ishii, N. & Ura, T. (1998). Content-based collaborative information filtering: Actively learning to classify and recommend documents. In: M. Klusch & G. Weiß (eds.). *Cooperative Information Agents II Learning, Mobility and Electronic Commerce for Information Discovery on the Internet*. [Online]. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 206–215. Available from: http://link.springer.com/10.1007/BFb0053686. [Accessed: 21 September 2015].

Diekmann, A. (2014). *Empirische Sozialforschung: Grundlagen, Methoden, Anwendungen*. Rororo Rowohlts Enzyklopädie 55678. Orig.-Ausg., vollst. überarb. und erw. Neuausg.

2007, 9. Aufl., [26. Aufl. der Gesamtausg.]. Reinbek bei Hamburg: Rowohlt-Taschenbuch-Verl.

Dumitriu, S., Gîrdea, M. & Buraga, S.C. (2007). Knowledge Management in a Wiki Platform via Microformats. In: *FLAIRS Conference*. [Online]. 2007, pp. 278–283. Available from: http://www.aaai.org/Papers/FLAIRS/2007/Flairs07-058. [Accessed: 18 September 2015].

Dziadosz, S. & Chandrasekar, R. (2002). Do thumbnail previews help users make better relevance decisions about web search results? In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. [Online]. 2002, ACM, pp. 365–366. Available from: http://dl.acm.org/citation.cfm?id=564446. [Accessed: 18 September 2015].

Eades, P. & Huang, M.L. (2000). Navigating clustered graphs using force-directed methods. *J. Graph Algorithms Appl.* 4 (3). pp. 157–181.

Eason, K. (1982). The process of introducing information technology. *Behaviour & Information Technology*. 1 (2). pp. 197–213.

Eid, M., Gollwitzer, M. & Schmitt, M. (2013). *Statistik und Forschungsmethoden: Lehrbuch*. 3., korrigierte Aufl. Weinheim: Beltz.

Eisenberg, M.B. & Berkowitz, R.E. (1988). *Curriculum initiative: an agenda and strategy for library media programs*. Information management, policy, and services. Norwood, N.J: Ablex Pub. Corp.

Eisenberg, M.B. & Brown, M.K. (1992). Current Themes Regarding Library and Information Skills Instruction: Research Supporting and Research Lacking. *School Library Media Quarterly*. 20 (2). pp. 103–110.

Ellis, D. (1989). A Behavioural Approach to Information Retrieval System Design. *Journal of Documentation*. 45 (3). pp. 171–212.

Ellis, D. & Haugan, M. (1997). Modelling the information seeking patterns of engineers and research scientists in an industrial environment. *Journal of Documentation*. 53 (4). pp. 384–403.

Eppler, M.J. & Mengis, J. (2004). The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines. *The Information Society*. 20 (5). pp. 325–344.

Evans, B.M. & Chi, E.H. (2008a). *Towards a model of understanding social search*. In: [Online]. 2008, ACM Press, p. 485. Available from: http://portal.acm.org/citation.cfm?doid=1460563.1460641. [Accessed: 18 September 2015].

Evans, B.M. & Chi, E.H. (2008b). *Towards a model of understanding social search*. In: [Online]. 2008, ACM Press, p. 83. Available from: http://portal.acm.org/citation.cfm?doid=1458583.1458600. [Accessed: 18 September 2015].

Farnham, S.D., Lahav, M., Raskino, D., Cheng, L. & Laird-McConnell, T. (2012). So. cl: An Interest Network for Informal Learning. In: *ICWSM*. [Online]. 2012. Available from: http://www.aaai.org/ocs/index.php/ICWSM/ICWSM12/paper/viewFile/4674%26lt%3B/4971. [Accessed: 21 September 2015].

Floyd, C. & Keil, R. (1983). Softwaretechnik und Betroffenenbeteiligung. In: P. Mambrey & R. Oppermann (eds.). *Beteiligung von Betroffenen bei der Entwicklung von Informationssystemen*. Franfurt, New York: Campus Verlag, pp. 137–167.

Fluit, C., Sabou, M. & Van Harmelen, F. (2006). Ontology-based information visualization: toward semantic web applications. In: *Visualizing the semantic web*. [Online]. Springer, pp. 45–58. Available from: http://link.springer.com/chapter/10.1007/1-84628-290-X_3. [Accessed: 18 September 2015].

Fox, E.A., Neves, F.D., Yu, X., Shen, R., Kim, S. & Fan, W. (2006). Exploring the computing literature with visualization and stepping stones & pathways. *Communications of the ACM*. 49 (4). p. 52.

Fox, S., Karnawat, K., Mydland, M., Dumais, S. & White, T. (2005). Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems (TOIS)*. 23 (2). pp. 147–168.

Franken, S. & Norbisrath, U. (2014a). Supporting the evaluation of complex search tasks with the SearchTrails tool. In: *Proceedings of 24th Annual International Conference on Computer Science and Software Engineering*. 2014, Toronto, Canada: IBM Corp., pp. 262–274.

Franken, S. & Norbisrath, U. (2014b). Trail Building During Complex Search Tasks. In: A. Butz, M. Koch, & J. Schlichter (eds.). *Mensch & Computer 2014 - Tagungsband*. 2014, München, Germany: De Gruyter Oldenbourg, pp. 135–144.

Franken, S., Norbisrath, U. & Prinz, W. (2015). Search Trails as Collaboration Artifacts – Evaluating the UX. In: S. Diefenbach, N. Henze, & M. Pielot (eds.). *Mensch und Computer 2015 – Proceedings*. 2015, Berlin, Germany: De Gruyter Oldenbourg, pp. 23–32.

Franken, S., Norbisrath, U. & Prinz, W. (2016). The Impact of SearchTrails on the Quality of Collaborative Search. In: W. Prinz, J. Borchers, M. Jarke, & M. Ziefle (eds.). *Mensch und Computer 2016 – Proceedings*. 2016, Aachen, Deutschland: De Gruyter Oldenbourg.

Franken, S. & Prinz, W. (2009). FacetBrowse: Ein facettenbasierter Dateibrowser für ein Groupware-System. In: S. Kain, D. Struve, & H. Wandke (eds.). *Workshop-Proceedings der Tagung Mensch & Computer 2009*. 2009, Berlin, Germany: Logos Verlag, pp. 214–216.

Gade, M., Hall, M., Huurdeman, H., Kamps, J., Koolen, M., Toms, E.G. & Walsh, D. (2014). *Supporting Complex Search Tasks*. [Online]. Available from: http://humanities.uva.nl/~mkoolen1/SCST15/#/overview. [Accessed: 9 December 2015].

Gapski, H. & Tekster, T. (2009). *Informationskompetenz in Deutschland - Überblick zum Stand der Fachdiskussion und Zusammenstellung von Literaturangaben, Projekten und Materialien zu einzelnen Zielgruppen*. [Online]. Düsseldorf, Germany: Landesanstalt für

Medien Nordrhein-Westfalen (LfM) Bereich Tagungen und Öffentlichkeitsarbeit. Available from: http://www.lfm-nrw.de/fileadmin/lfm-nrw/Aktuelle_Forschungsprojekte/Informationskompetenz_in_Deutschland_August_09.pdf. [Accessed: 21 September 2015].

Garfield, E. (1970). When is a negative search result positive. *Essays of an Information Scientist*. 1. pp. 117–118.

George, D. & Mallery, P. (2014). *IBM statistics 21 step by step: a simple guide and reference*. Thirteenth edition. Boston: Pearson.

Gersh, J., Lewis, B., Montemayor, J., Piatko, C. & Turner, R. (2006). Supporting insight-based information exploration in intelligence analysis. *Communications of the ACM*. 49 (4). p. 63.

Goldberg, D., Nichols, D., Oki, B.M. & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*. 35 (12). pp. 61–70.

Golder, S.A. & Huberman, B.A. (2006). Usage patterns of collaborative tagging systems. *Journal of information science*. 32 (2). pp. 198–208.

Golovchinsky, G., Adcock, J., Pickens, J., Qvarfordt, P. & Back, M. (2008). Cerchiamo: a collaborative exploratory search tool. *Proceedings of Computer Supported Cooperative Work (CSCW)*. pp. 8–12.

Golovchinsky, G., Diriye, A. & Dunnigan, T. (2012). The future is in the past: designing for exploratory search. In: *Proceedings of the 4th Information Interaction in Context Symposium*. [Online]. 2012, ACM, pp. 52–61. Available from: http://dl.acm.org/citation.cfm?id=2362738. [Accessed: 18 September 2015].

Golovchinsky, G., Qvarfordt, P. & Pickens, J. (2009). Collaborative Information Seeking. *Computer*. 42 (3). pp. 47–51.

González-Ibáñez, R. & Shah, C. (2011). Coagmento: A system for supporting collaborative information seeking. *Proceedings of the American Society for Information Science and Technology*. 48 (1). pp. 1–4.

Goodman, D. (1988). *Danny Goodman's HyperCard developer's guide*. Macintosh performance library. Toronto ; New York: Bantam Books.

Gross, B.M. (1964). *The managing of organizations: The administrative struggle*. New York, NY: The Free Press of Glencoe.

Gross, T. (1998). CSCW3: Transparenz-und Kooperationsunterstützung für das WWW. *Groupware und organisatorische Innovation. Tagungsband der D-CSCW'98*. pp. 37–50.

Gwizdka, J. (2008). Revisiting search task difficulty: Behavioral and individual difference measures. *Proceedings of the American Society for Information Science and Technology*. 45 (1). pp. 1–12.

Halvey, M., Vallet, D., Hannah, D., Feng, Y. & Jose, J.M. (2010). An asynchronous collaborative search system for online video search. *Information Processing & Management*. 46 (6). pp. 733–748.

Hanani, U. & Frank, A.J. (2001). *The parallel evolution of search engines and digital libraries: their convergence to the Mega-Portal*. In: [Online]. 2001, IEEE Comput. Soc, pp. 211–218. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=942177. [Accessed: 18 September 2015].

Hearst, M.A. (2006). Clustering versus faceted categories for information exploration. *Communications of the ACM*. 49 (4). pp. 59–61.

Hendrickson, L.E., Sorooshian, S. & Brazil, Larry E. (1988). Comparison of Newton-Type and Direct Search Algorithms for Calibration of Conceptual Rainfall-Runofi 'Models. *Water Resources Research*. 24 (5). pp. 691–700.

Hevner, A.R., March, S.T. & Ram, S. (2004). Design Science In Information Systems Research. *MIS Quarterly*. 28 (1). pp. 75–105.

Höchstötter, N. (2007). Suchverhalten im Web-Erhebung, Analyse und Möglichkeiten. *Information Wissenschaft und Praxis*. 58 (3). pp. 135–140.

Ingold, M. (2005). Das Bibliothekarische Konzept der Informationskompetenz - Ein Überblick. *Berliner Handreichungen zur Bibliothekswissenschaft*. [Online]. 128. Available from: http://www.ib.hu-berlin.de/~kumlau/handreichungen/h128/h128.pdf. [Accessed: 21 September 2015].

Ingwersen, P. & Järvelin, K. (2005). *The Turn*. The Information Retrieval Series. [Online]. Berlin/Heidelberg: Springer-Verlag. Available from: http://link.springer.com/10.1007/1-4020-3851-8. [Accessed: 8 December 2015].

Irving, A. (1985). *Study and information skills across the curriculum*. Portsmouth, New Hampshire: Heinemann Educational Books.

Jansen, B.J. (2006). Using temporal patterns of interactions to design effective automated searching assistance. *Communications of the ACM*. 49 (4). p. 72.

Jansen, B.J., Ramadoss, R., Zhang, M. & Zang, N. (2006). Wrapper: An application for evaluating exploratory searching outside of the lab. *EESS 2006*. [Online]. 14. Available from: http://www.researchgate.net/profile/Jim_Jansen/publication/228830541_Wrapper_An_application_for_evaluating_exploratory_searching_outside_of_the_lab/links/02e7e51f8ad289d769000000.pdf. [Accessed: 21 September 2015].

Jansen, B.J., Spink, A. & Saracevic, T. (2000). Real life, real users, and real needs: a study and analysis of user queries on the web. *Information processing & management*. 36 (2). pp. 207–227.

Kelly, R. & Payne, S.J. (2014). *Collaborative web search in context: a study of tool use in everyday tasks*. In: [Online]. 2014, ACM Press, pp. 807–819. Available from: http://dl.acm.org/citation.cfm?doid=2531602.2531617. [Accessed: 18 September 2015].

Khare, R. (2006). Microformats: The next (small) thing on the semantic web? *Internet Computing, IEEE*. 10 (1). pp. 68–75.

Khare, R. & Çelik, T. (2006). Microformats: a pragmatic path to the semantic web. In: *Proceedings of the 15th international conference on World Wide Web*. [Online]. 2006, ACM, pp. 865–866. Available from: http://dl.acm.org/citation.cfm?id=1135917. [Accessed: 18 September 2015].

Kline, P. (2000). *The handbook of psychological testing*. 2nd ed. London ; New York: Routledge.

Klöckner, K., Mambrey, P., Sohlenkamp, M., Prinz, W., Fuchs, L., Kolvenbach, S., Pankoke-Babatz, U. & Syri, A. (1995). POLITeam bridging the gap between Bonn and Berlin for and with the users. In: *Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work ECSCW'95*. 1995, Springer, pp. 17–32.

Koshman, S. (2006). Exploratory search visualization: identifying factors affecting evaluation. *EESS 2006*. Proceedings of the ACM SIGIR 2006 Workshop on 'Evaluating Exploratory Search Systems'. pp. 20–23.

Kramer, S.N. & Sandor, G.N. (1975). Selective precision synthesis—a general method of optimization for planar mechanisms. *Journal of Manufacturing Science and Engineering*. 97 (2). pp. 689–701.

Kruchten, P. (2001). From Waterfall to Iterative Development—A Challenging Transition for Project Managers. *Rational Edge, Rational Software*. [Online]. Available from: http://www.itu.dk/~hesj/SUTI/Artikler/waterfall2iterative.pdf. [Accessed: 21 September 2015].

Kuhlthau, C.C. (1988). Developing a model of the library search process: Cognitive and affective aspects. *RQ*. 28 (2). pp. 232–242.

Kuhlthau, C.C. (1991). Inside the search process: Information seeking from the user's perspective. *Journal of the American Society for Information Science (JASIS)*. 42 (5). pp. 361–371.

Kules, B. & Capra, R. (2008). Creating Exploratory Search Tasks for a Faceted Search Interface. *Proceedings of the 2nd Workshop on Human Computer Interaction (HCIR 2008)*. pp. 18–21.

Kules, B. & Capra, R. (2009). *Designing exploratory search tasks for user studies of information seeking support systems*. In: [Online]. 2009, ACM Press, p. 419. Available from: http://portal.acm.org/citation.cfm?doid=1555400.1555492. [Accessed: 21 September 2015].

Laugwitz, B., Held, T. & Schrepp, M. (2008). Construction and Evaluation of a User Experience Questionnaire. In: A. Holzinger (ed.). *HCI and Usability for Education and Work*. [Online]. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 63–76. Available from: http://link.springer.com/10.1007/978-3-540-89350-9_6. [Accessed: 23 September 2015].

Lewandowski, D. (2006). Query types and search topics of German Web search engine users. *Information Services & Use*. 26 (4). pp. 261–269.

Lewandowski, D. (2015). *Suchmaschinen verstehen*. Xpert.press. Berlin: Springer Vieweg.

Lewandowski, D. (2008). The retrieval effectiveness of web search engines: considering results descriptions. *Journal of Documentation*. 64 (6). pp. 915–937.

Li, Y. (2004). Task type and a faceted classification of tasks. *Poster presented at the American Society of Information Science and Technology, Providence, RI*.

Malone, T.W., Grant, K.R., Lai, K.-Y., Rao, R. & Rosenblitt, D. (1987). Semistructured messages are surprisingly useful for computer-supported coordination. *ACM Transactions on Information Systems (TOIS)*. 5 (2). pp. 115–131.

Manning, C.D., Raghavan, P. & Schütze, H. (2008). *Introduction to information retrieval*. [Online]. New York: Cambridge University Press. Available from: http://nlp.stanford.edu/IR-book/. [Accessed: 9 December 2015].

Marchionini, G. (2006). Exploratory search: from finding to understanding. *Communications of the ACM*. 49 (4). p. 41.

Marchionini, G. (1989). Information-seeking strategies of novices using a full-text electronic encyclopedia. *Journal of the American Society for Information Science*. 40 (1). pp. 54–66.

McEneaney, J.E. (2001). Graphic and numerical methods to assess navigation in hypertext. *International Journal of Human-Computer Studies*. 55 (5). pp. 761–786.

Minoli, D. (1975). Combinatorial graph complexity. *Atti. Accad. Naz. Lincei, VIII. Ser., Rend., Cl. Sci. Fis. Mat. Nat*. 59 (6). pp. 651–661.

Moore, G.E. (1965). Cramming more components onto integrated circuits. *Electronics*. 38 (8). pp. 114–117.

Morris, M.R. & Horvitz, E. (2007). *SearchTogether: an interface for collaborative web search*. In: [Online]. 2007, ACM Press, p. 3. Available from: http://portal.acm.org/citation.cfm?doid=1294211.1294215. [Accessed: 21 September 2015].

Neel, D.L. & Orrison, M.E. (2006). The linear complexity of a graph. *the electronic journal of combinatorics*. 13 (1). p. R9.

Nelson, M.R. (1994). We have the information you want, but getting it will cost you!: held hostage by information overload. *Crossroads*. 1 (1). pp. 11–15.

Nelson, T. (1960). *Project Xanadu® - The original Hypertext project*. [Online]. 1960. Available from: http://xanadu.com/. [Accessed: 23 September 2015].

Nielsen, J. (1993). Iterative User-Interface Design. *Computer*. 26 (11). pp. 32–41.

Pariser, E. (2011). *The filter bubble: what the Internet is hiding from you*. New York: Penguin Press.

Pauleen, D. (2009). Personal knowledge management: Putting the 'person' back into the knowledge equation D. Pauleen (ed.). *Online Information Review*. 33 (2). pp. 221–224.

Paul, S.A. & Ringel Morris, M. (2009). CoSense: enhancing sensemaking for collaborative web search. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. [Online]. 2009, ACM, pp. 1771–1780. Available from: http://dl.acm.org/citation.cfm?id=1518974. [Accessed: 18 September 2015].

Peffers, K., Tuunanen, T., Rothenberger, M.A. & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*. 24 (3). pp. 45–77.

Pirolli, P. (2009). Powers of 10: Modeling Complex Information-Seeking Systems at Multiple Scales. *Computer*. 42 (3). pp. 33–40.

Pirolli, P. & Card, S. (2005). The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In: *Proceedings of international conference on intelligence analysis*. 2005, pp. 2–4.

Pitkow, J. & Recker, M. (1994). Results from the First World-Wide Web user survey. *Computer Networks and ISDN Systems*. 27 (2). pp. 243–254.

Prinz, W. & Gräther, W. (2000). Das Social Web Cockpit: Ein Assistent für virtuelle Gemeinschaften. *Verteiltes Arbeiten–Arbeit der Zukunft. Tagungsband der D-CSCW 2000*.

Reeder, R.W., Pirolli, P. & Card, S.K. (2001). *WebEyeMapper and WebLogger: tools for analyzing eye tracking data collected in web-use studies*. In: [Online]. 2001, ACM Press, p. 19. Available from: http://portal.acm.org/citation.cfm?doid=634067.634082. [Accessed: 21 September 2015].

Ringel Morris, M. (2013). *Collaborative search revisited*. In: [Online]. 2013, ACM Press, p. 1181. Available from: http://dl.acm.org/citation.cfm?doid=2441776.2441910. [Accessed: 18 September 2015].

Ringel Morris, M., Lombardo, J. & Wigdor, D. (2010). WeSearch: supporting collaborative search and sensemaking on a tabletop display. In: *Proceedings of the 2010 ACM conference on Computer supported cooperative work*. [Online]. 2010, ACM, pp. 401–410. Available from: http://dl.acm.org/citation.cfm?id=1718987. [Accessed: 18 September 2015].

Robertson, S.E. (1977). Theories and Models in Information Retrieval. *Journal of Documentation*. 33 (2). pp. 126–148.

Robins, D. (2000). Interactive information retrieval: Context and basic notions. *Informing Science*. 3 (2). pp. 57–62.

Romano, N.C., Nunamaker, J.F., Roussinov, D. & Chen, H. (1999). Collaborative information retrieval environment: Integration of information retrieval with group support systems. In: *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*. [Online]. 1999, IEEE, pp. 10–19. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=772729. [Accessed: 21 September 2015].

Rose, D.E. & Levinson, D. (2004). Understanding user goals in web search. In: *Proceedings of the 13th international conference on World Wide Web*. [Online]. 2004, pp. 13–19. Available from: http://dl.acm.org/citation.cfm?id=988675. [Accessed: 3 May 2013].

Rotman, D., Procita, K., Hansen, D., Sims Parr, C. & Preece, J. (2012). Supporting content curation communities: The case of the Encyclopedia of Life. *Journal of the American Society for Information Science and Technology*. 63 (6). pp. 1092–1107.

Russell, D.M., Stefik, M.J., Pirolli, P. & Card, S.K. (1993). The cost structure of sensemaking. In: *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. [Online]. 1993, ACM, pp. 269–276. Available from: http://dl.acm.org/citation.cfm?id=169209. [Accessed: 21 September 2015].

Schmidt-Mänz, N. (2007). Erkenntnisse aus dem Suchverhalten im Web „Muster in Suchanfragen“. In: J. Eberspächer & S. Holtel (eds.). *Suchen und Finden im Internet*. [Online]. Springer Berlin Heidelberg, pp. 21–28. Available from: http://link.springer.com/10.1007/978-3-540-38157-0_3. [Accessed: 22 September 2015].

Schmidt-Mänz, N. (2005). *Suchverhalten im Web - Empirische Ergebnisse*. [Online]. Available from: http://nbn-resolving.de/urn:nbn:de:swb:ch1-200500637.

Schmidt-Mänz, N. & Bomhardt, C. (2005). Wie suchen Onliner im Internet? *Science Factory/Absatzwirtschaft*. 2. pp. 5–8.

Schmidt-Mänz, N. & Koch, M. (2006). *A General Classification of (Search) Queries and Terms*. In: [Online]. 2006, IEEE, pp. 375–381. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1611622. [Accessed: 18 September 2015].

Schmitt, N. (1996). Uses and abuses of coefficient alpha. *Psychological Assessment*. 8 (4). pp. 350–353.

schraefel, m. c., Wilson, M., Russell, A. & Smith, D.A. (2006). mSpace: improving information access to multimedia domains with multimodal exploratory search. *Communications of the ACM*. 49 (4). p. 47.

Schubert, S. & Schwill, A. (2011). Didaktik der Informatik. In: *Didaktik der Informatik*. [Online]. Heidelberg: Spektrum Akademischer Verlag, pp. 1–30. Available from: http://link.springer.com/10.1007/978-3-8274-2653-6_1. [Accessed: 18 September 2015].

Sebrechts, M.M., Cugini, J.V., Laskowski, S.J., Vasilakis, J. & Miller, M.S. (1999). Visualization of search results: a comparative evaluation of text, 2D, and 3D interfaces. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. [Online]. 1999, ACM, pp. 3–10. Available from: http://dl.acm.org/citation.cfm?id=312634. [Accessed: 18 September 2015].

Shneiderman, B. (1997). Designing information-abundant web sites: issues and recommendations. *International Journal of Human-Computer Studies*. 47 (1). pp. 5–29.

Shneiderman, B. (1992). *Designing the user interface: strategies for effective human-computer interaction*. 2nd Ed. Addison-Wesley Reading, MA.

Shneiderman, B., Byrd, D. & Croft, W.B. (1997). Clarifying search: A user-interface framework for text searches. *D-lib magazine*. 3 (1). pp. 18–20.

Singer, G. (2012). *Web search engines and complex information needs*. [Online]. Tartu, Estonia: University of Tartu, Estonia. Available from: https://www.google.de/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0ahUKEwiEvsDp1czJAhUFRg8KHTRJAXsQFggsMAI&url=http%3A%2F%2Fdspace.ut.ee%2Fbitstream%2Fhandle%2F10062%2F26463%2Fsinger_georg.pdf&usg=AFQjCNHlS6QxAO4bv6lfXc0Fm3y6pepSfw.

Singer, G., Danilov, D. & Norbisrath, U. (2012a). Complex search: aggregation, discovery, and synthesis. *Proceedings of the Estonian Academy of Sciences*. 61 (2). p. 89.

Singer, G., Norbisrath, U. & Lewandowski, D. (2012b). Impact of gender and age on performing search tasks online. *H. Reiterer & O. Deussen (Eds.): Proceedings of Mensch & Computer 2012*. pp. 23–32.

Singer, G., Norbisrath, U. & Lewandowski, D. (2012c). Ordinary search engine users assessing difficulty, effort, and outcome for simple and complex search tasks. In: *Proceedings of the 4th Information Interaction in Context Symposium*. [Online]. 2012, ACM, pp. 110–119. Available from: http://dl.acm.org/citation.cfm?id=2362746. [Accessed: 21 September 2015].

Singer, G., Norbisrath, U. & Lewandowski, D. (2013). Ordinary search engine users carrying out complex search tasks. *Journal of Information Science*. 39 (3). pp. 346–358.

Singer, G., Norbisrath, U., Vainikko, E., Kikkas, H. & Lewandowski, D. (2011). Search-logger analyzing exploratory search tasks. In: *Proceedings of the 2011 ACM Symposium on Applied Computing*. [Online]. 2011, ACM, pp. 751–756. Available from: http://dl.acm.org/citation.cfm?id=1982350. [Accessed: 21 September 2015].

Singer, K., Singer, G., Lepik, K., Norbisrath, U. & Pruulmann-Vengerfeldt, P. (2012d). Search Strategies of Library Search Experts. *Qualitative and Quantitative Methods in Libraries (QQML)*. 1. pp. 89–97.

Singla, A., White, R. & Huang, J. (2010). *Studying trailfinding algorithms for enhanced web search*. In: [Online]. 2010, ACM Press, p. 443. Available from: http://portal.acm.org/citation.cfm?doid=1835449.1835524. [Accessed: 21 September 2015].

Smalheiser, N.R. (2002). Informatics and hypothesis-driven research. *EMBO reports*. 3 (8). pp. 702–702.

Stripling, B.K. & Pitts, J.M. (1988). *Brainstorms and blueprints: Teaching library research as a thinking process*. Englewood, Colorado: Libraries Unlimited Press.

Stuckenschmidt, H., Van Harmelen, F., De Waard, A., Scerri, T., Bhogal, R., Van Buel, J., Crowlesmith, I., Fluit, C., Kampman, A., Broekstra, J. & others (2004). Exploring large document repositories with RDF technology: the DOPE project. *Intelligent Systems, IEEE*. 19 (3). pp. 34–40.

Toffler, A. (1970). *Future shock*. New York: Random House.

Toms, E.G., Freund, L. & Li, C. (2004). WiIRE: the Web interactive information retrieval experimentation system prototype. *Information Processing & Management*. 40 (4). pp. 655–675.

Turnbull, D. (1998). WebTracker: A tool for understanding web use. *Unpublished report*. pp. 19–23.

UEQ Evaluation Sheet (2015). *UEQ Evaluation Sheet*. [Online]. Available from: http://www.ueq-online.org/. [Accessed: 9 December 2015].

Ullrich, M., Maier, A. & Angele, J. (2003). Taxonomie, thesaurus, topic map, Ontologie—ein Vergleich. *Ontoprise GmbH*. [Online]. Available from: http://www.ullri.ch/download/Ontologien/ttto13.pdf. [Accessed: 18 September 2015].

Vaishnavi, V. & Kuechler, W. (2004). *Design research in information systems*. [Online]. 2004. Available from: http://www.desrist.org/design-research-in-information-systems/. [Accessed: 21 September 2015].

Villa, R., Cantador, I., Joho, H. & Jose, J.M. (2009). An aspectual interface for supporting complex search tasks. In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. [Online]. 2009, ACM, pp. 379–386. Available from: http://dl.acm.org/citation.cfm?id=1572007. [Accessed: 18 September 2015].

Voyloshnikova, E. & Storey, M.-A. (2014). Towards understanding digital information discovery and curation. In: *Proceedings of 24th Annual International Conference on Computer Science and Software Engineering*. 2014, IBM Corp., pp. 247–261.

Weitz, S. (2009). *Search isn't 'Search': A look at how search engines can better serve what people are actually doing on the web*. In: 22 April 2009, Munich, Germany: Keynote held by S. Weitz (Microsoft) at the Search Marketing Expo (SMX).

Westerman, S.. & Cribbin, T. (2000). Mapping semantic information in virtual space: dimensions, variance and individual differences. *International Journal of Human-Computer Studies*. 53 (5). pp. 765–787.

White, R.W. (2009). Designing Information Seeking Support Systems. *Information Seeking Support Systems An invitational workshop sponsored by the National Science Foundation*. pp. 55–59.

White, R.W., Kules, B., Drucker, S.M. & others (2006a). Supporting Exploratory Search, Introduction, Special Issue, Communications of the ACM. *Communications of the ACM*. 49 (4). pp. 36–39.

White, R.W., Marchionini, G. & Muresan, G. (2008). Evaluating exploratory search systems: Introduction to special topic issue of information processing and management. *Information Processing & Management*. 44 (2). pp. 433–436.

White, R.W., Muresan, G. & Marchionini, G. (2006b). Report on ACM SIGIR 2006 workshop on evaluating exploratory search systems. *ACM SIGIR Forum*. 40 (2). p. 52.

White, R.W. & Roth, R.A. (2009). *Exploratory search beyond the query-response paradigm*. [Online]. [San Rafael, Calif.]: Morgan & Claypool. Available from: http://www.morganclaypool.com/doi/abs/10.2200/S00174ED1V01Y200901ICR003. [Accessed: 9 January 2013].

Widhalm, R. & Mück, T. (2002). *Topic Maps: Semantische Suche im Internet ; mit 98 Tabellen*. Xpert.press. Berlin: Springer.

Xu, W., Dainoff, M.J. & Mark, L.S. (1999). Facilitate Complex Search Tasks in Hypertext by Externalizing Functional Properties of a Work Domain. *International Journal of Human-Computer Interaction*. 11 (3). pp. 201–229.

Zhong, C., Shah, S., Sundaravadivelan, K. & Sastry, N. (2013). Sharing the Loves: Understanding the How and Why of Online Content Curation. In: *ICWSM*. [Online]. 2013. Available from: http://www.inf.kcl.ac.uk/pg/czhong/papers/icwsm13.pdf. [Accessed: 21 September 2015].

Table of Figures

Table of Tables

Appendix

The appendix consists of the questionnaires developed for the first and second user study, and a set of the search trail visualizations from the second user study.

A1 – Questionnaire from the first user study

I developed this questionnaire for the evaluation of the first user study. The participants received this questionnaire before study started, and were asked to answer the questions after each search task.

SearchLogger Evaluation

You are about to work with the SearchLogger application, which supports you while performing different information search tasks on the web. This questionnaire should help you to express your experiences with the SearchLogger and the search tasks you performed with it.

The evaluation will be done in 6 steps:
1 - You will be introduced into the different functionalities of the SearchLogger and the procedure of this evaluation.
2 – Enter your username and give a title to your search.
3 - Read and perform the first search task and answer the questions.
4 - Read the following three search tasks and select the two that you like most.
5 - For each search task, perform the task and answer the subsequent-questions.
6 - Please answer the general questions after you performed all three tasks.

Search Task 1

Try to find 9 films, using Beethoven's 7th symphony as a score. Which part of this symphony is normally used as a score and what does it sound like?

Have a guess: How difficult will this task be?

very difficult						very easy

After Search Task 1
(Test person used the logger: Yes / No)

How familiar were you with this subject when you began this task?

not familiar						very familiar

How difficult was it to accomplish this task?

very difficult						very easy

Did you discover information that was new to you?

not much						a lot

Did you learn something about your search topic?

not much						a lot

I am confident that I fulfilled the task asked of me

strongly disagree						strongly agree

To what extent did completing this task involve finding a single vs. multiple resources / pages?

multiple resources						single resource

To what extent did your search goal change based on the results you found?

not at all						a lot

Were the documents you considered important at early stages of your search still important for you at the end of your search?

No, not any						Yes

If you used the SearchLogger: Was it helpful for you to see the course of your navigation?

Not, not at all						Yes, very

What was the SearchLogger especially good for while performing this search task?

What did you like about the task?

* * *

Of the following three search tasks, please select the two that interest you most. Perform the two tasks and answer the subsequent questions.
Make sure to perform the tasks in the right order with and/or without the SearchLogger.

Search Task 2

(crypto) Crypto currencies: What are they, what is their current status, how do you obtain them?

Search Task 3

(ad) Check on the manyfold of German (or other countries) classified-ad-portals if there is a recent sale of a glassed frame in DIN A0 format, which you could bring home somehow. How much would it cost?

Search task 4

(video) What is the state-of-the-art for video-glasses, and how would it be possible to build some on your own?

I chose: (crypto) (ad) (video)

Have a guess: How difficult will this task be?

very difficult						very easy

After this search task
(Test person used the logger: Yes / No)

How familiar were you with this subject when you began this task?

not familiar						very familiar

How difficult was it to accomplish this task?

very difficult						very easy

Did you discover information that was new to you?

not much						a lot

Did you learn something about your search topic?

not much						a lot

I am confident that I fulfilled the task asked of me

strongly disagree						strongly agree

To what extent did completing this task involve finding a single vs. multiple resources / pages?

multiple resources						single resource

To what extent did your search goal change based on the results you found?

not at all						a lot

Were the documents you considered important at early stages of your search still important for you at the end of your search?

No, not any						Yes

If you used the SearchLogger: Was it helpful for you to see the course of your navigation?

Not, not at all						Yes, very

What was the SearchLogger especially good for while performing this search task?

What did you like about the task?

I chose: (crypto) (ad) (video)

Have a guess: How difficult will this task be?

very difficult						very easy

After this search task
(Test person used the logger: Yes / No)

How familiar were you with this subject when you began this task?

not familiar						very familiar

How difficult was it to accomplish this task?

very difficult						very easy

Did you discover information that was new to you?

not much						a lot

Did you learn something about your search topic?

not much						a lot

I am confident that I fulfilled the task asked of me

strongly disagree						strongly agree

To what extent did completing this task involve finding a single vs. multiple resources / pages?

multiple resources						single resource

To what extent did your search goal change based on the results you found?

not at all						a lot

Were the documents you considered important at early stages of your search still important for you at the end of your search?

No, not any						Yes

If you used the SearchLogger: Was it helpful for you to see the course of your navigation?

Not, not at all						Yes, very

What was the SearchLogger especially good for while performing this search task?

What did you like about the task?

General questions

Your age: 20-29 30-39 40-49 50-59

Your profession: ____________

Did the search engines you used support the way you are searching?

not at all						very good

What do you like about the way you can enter queries into search engines? What do you miss?

What do or don't you like about the way search engines present the results to you?

Did you have enough time to perform the tasks in this evaluation?

not enough						too much

If you used the SearchLogger during the evaluation

Did you get enough information to work with the SearchLogger?

not enough						too much

Was the type of visualization clear to you?

too complex						too simple

Do you need more visual information? If yes, which?

The visualisation is…

too slow						too fast

Did the SearchLogger help you performing your search tasks?

No, not at all						Yes, very

Has creating the synthesis document been fun?

No, not at all						Yes, very

Is it easy to get back to the pages where you found the information?

No, not at all						Yes, very

Did you use the keyword list and did it help you performing your search tasks?

No, not at all						Yes, very

Did the clustering help you performing your search tasks?

No, not at all						Yes, very

Did the highlight-overview help you performing your search tasks?

No, not at all						Yes, very

Does the tool have enough features?

not enough						too many

Which ones are missing?

Does this experience change how you attack complex search problems in the future?

No, not at all						Yes, sure

How?

Could you think of other situations where you would like to use the SearchLogger?

I would want this type of visualization more often!

No, not at all						Yes, sure

General comments: What did you like about the tool, what could be better? Is something not necessary, is something missing?

Thank you for your support!

A2 – Questionnaires from the second user study

For the second user study, I developed a set of six documents, which I handed out in the different phases of the user study. For the user study, the questionnaires were distributed without the names of the question groups. The following documents were developed for the user study:

1. **Introduction**
   This document introduces the overall process of the user study, the most important features of SearchTrails, and assigns the user ID and the topic to the study participants.
2. **Form of consent**
   The form of consent was signed by the study participants and ensures them data confidentiality.
3. **Pre-evaluation**
   The study participants filled out the pre-evaluation questionnaire in class before the study started. It mainly asks for demographical data and about the participant's normal ways of searching.
4. **Intermediate evaluation**
   In the intermediate evaluation, I ask about the experiences during the search tasks with or without the use of SearchTrails.
5. **Introduction to phase 2**
   This document introduces the study participants to the new topics.
6. **Final evaluation**
   In the final evaluation, I ask for the experiences during the evaluation of the given artifact and the experiences with SearchTrails. This questionnaire was subsided by a standard UEQ questionnaire, which is not presented here.

In the following, I present the questionnaires 3, 4, and 6 for group SR3, which used SearchTrails (S) in phase 1 and received a report as a given artifact for phase 2 (R). The group works on the topic '3D printing' (3) in phase 1, and 'Home automation' in phase 2.

Pre-evaluation questionnaire

This is a preliminary questionnaire to be filled out before the first phase of the evaluation. This questionnaire covers some general details relevant for the evaluation.

A – Demographics		
(1) Your age	________ years	
(2) Your gender	☐ Female	☐ Male
(3) Your course		
(4) Did you use the Internet yesterday for searching something?	☐ Yes	☐ No
(5) How often do you use the Internet for searching per day (approximately)?	________ times	

(6) How often did you search for something within the last week, so starting from last Wednesday (approximately)?	________ times

B – Setting. How do you perform web search normally?		
(1) How would you rate your level of experience with web search?	☐ professional ☐ better than average ☐ intermediate ☐ not very skilled ☐ newbie	
(2) Does your browser open old tabs when starting?	☐ Yes	☐ No
(3) Do you use browser plug ins to improve your search experience?	☐ Yes	☐ No
(4) If yes, which ones?		
(5) How often did you experience complex/unspecific /tedious search tasks since last Wednesday?	________ times	
(6) What tools do you use to solve them? (You can check multiple tools) ☐ Browser ☐ Handwritten notes ☐ Text editing ☐ Spreadsheets ☐ PowerPoint ☐ Browser bookmarks ☐ Social bookmarks (delicious etc.) (which?): ________________________ ☐ Mobile devices ☐ Browser plug-ins / Extensions (which?): ___________________________ ☐ Others (which?): ____		
(7) Please describe your approach for solving complex/unspecific /tedious search tasks:		
(8) Do you take notes during these kinds of searches?	☐ Yes	☐ No
(9) If yes, how do you take notes?		

(10) How long does it take you to do these kinds of searches normally (approximately)?	________ minutes	
(11) Do you use bookmarks, tags, social tagging, social bookmarking?	☐ Yes	☐ No
(12) Do you feel well-supported by current search engines during these tasks?	☐ Yes	☐ No
(13) Did you want to go back to older searches within the last two weeks?	☐ Yes	☐ No
(14) Do you share search results with your colleagues?	☐ Yes	☐ No
(15) If yes, how do you do this? Describe your approach:		
(16) If yes, how easy was this?	☐ Very easy ☐ Easy ☐ Moderate ☐ Hard ☐ Very hard	
(17) Have you ever wanted to hand over your complete search results on a topic to someone else?	☐ Yes	☐ No
(18) If yes, in which situation?		

C - You got the topic '3D-Printing' for phase 1.	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) How long do you expect to need for that search?	________ minutes				
(2) How many sessions do you expect to need for that search?	________ sessions				
(3) I feel confident to cover the topic thoroughly.					
(4) I am looking forward to have technical support while performing the search task.					

Thank you!

Intermediate questionnaire

You have now performed the first half of the evaluation study of SearchTrails. Thanks a lot for that!
You did a search with SearchTrails and prepared a report about your findings based on your search results.
You are now requested to answer the following questions, in which you should comment on your experiences with SearchTrails.

D – You got the topic '3D-Printing' for phase 1.	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) What operating system did you use?					
(2) How long did you need for that search?	___________ minutes				
(3) How many sessions did you need for that search?	___________ sessions				
(4) I feel confident to have covered the topic thoroughly.					
(5) I enjoyed working with SearchTrails while performing the search task.					

E – Search topic	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) I liked the search topic.					
(2) I found the topic interesting.					
(3) The search task was motivating.					
(4) I learned something new during my search on that topic.					

F – SearchTrails – Features

Features / Opinion	I used this feature.	How important was the feature during your search?	This feature is a good idea, but a bad implementation.
	Check / leave blank	Please rate with a number from 1 'very important' to 5 'unimportant/useless'	Check / leave blank
Search trail visualization as a graph			
Keyword table			
Select and Search for keywords from the keywords table			
Blue Nodes / Highlight pages in the graph			

Red nodes / Irrelevant pages in the graph			
Hulls around nodes / Clusters			
Collapsing clusters			
Deleting nodes			
Continuing a search by entering the trails ID			
Adding highlights by pressing 'ALT'			
Adding handwritten high-lights			
Deleting highlights			
Marking pages irrelevant by pressing '-' (minus)			
Clicking on URLs/Links connected to the nodes in the visualization to get back to a page			

G – Self-Awareness Evaluation	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) Sometimes, I visited pages that were irrelevant for my search (e.g. news pages).					
(2) The need to create the report made my search more focused.					
(3) Seeing the search graph helped me to be aware of my search results.					
(4) I became aware of pages I visit frequently.					
(5) List 3 insights that you gained when you were monitoring your own search behavior: 1) 2) 3)					

H – Aggregation Evaluation	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) At least once I felt overwhelmed by the amount of new information during my search.					
(2) If you agree: How often (approximately)?	___________ times				

(3) The graph visualization behaved like expected.					
(4) The search graph obviously resembles the course of my search.					
(5) Seeing the course of my search was interesting.					
(6) It was interesting to see all visited pages on a glance.					
(7) It was interesting to see how I came to search results and where I went thereafter.					
(8) The search trail helped to see all pages where I found relevant information for my search.					
(9) The search graph visualization helps to remember relevant facts of the search topic.					
(10) At least once I wanted to get back to pages visited earlier.					
(11) SearchTrails helped me to get back to previous pages.					
(12) At least once I had the feeling to be moving in circles, often seeing the same pages.					

I – Synthesis Evaluation	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) SearchTrails was helpful to capture relevant search results.					
(2) It felt safe to capture search results.					
(3) I looked into the search highlights after the actual search.					
(4) The highlights helped to gather all relevant resources for the report.					
(5) Being able to interact with the graph and its highlights helped to condense the relevant information.					
(6) At least once I forgot to highlight things that turned out to be important later.					
(7) I had SearchTrails and the tabs used for browsing both always visible (e.g. on two screens).	☐ Yes			☐ No	
(8) Switching between the SearchTrails-Tab and the tabs used for browsing was cumbersome.					

J – Discovery Evaluation	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) I think I have found all relevant facts that can be found on my search task.					

(2) At least once I got stuck during my search.					
(3) If you agree: How often?	________ times				
(4) SearchTrails enabled me to discover additional facts for my search task.					
(5) If you agree, why?					
(6) The keywords column helped to see the context in which I was browsing.					
(7) The keywords helped to find relevant search terms that I would not have come up with.					

K – Improving motivation	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) Searching with SearchTrails was motivating.					
(2) I searched longer with SearchTrails than I would have searched without the tool.					
(3) Using SearchTrails made me more efficient than I would have been without the tool.					

L – Can you think of more application cases that SearchTrails is suited for?

M – General comments or remarks on the first phase of the evaluation:

N – You got the topic 'Home automation' for phase 2.	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) How long do you expect to need for that search?	________ minutes				
(2) How many sessions do you expect to need for that search?	________ sessions				
(3) I feel confident to cover the topic thoroughly.					
(4) I am looking forward to have technical support while performing the next search task.					

Thank you!

Final evaluation

These questions deal with your experiences in phase 2 of this study, after you received your colleagues report. You read your colleagues report and tried to extend it to create an even better report on the more specific question.

O – You got the topic 'Home automation' for phase 2.	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) What operating system did you use?					
(2) How long did you need for that search?	__________ minutes				
(3) How many sessions did you need for that search?	__________ sessions				
(4) I feel confident to have delivered a thorough answer to the question.					
(5) I enjoyed working with SearchTrails while performing the search task.					

P – Search topic	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) I liked the new search topic.					
(2) I found the new topic interesting.					
(3) I learned something new during my search on the new topic.					

Q – Evaluating your colleagues' report	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) Finding out what the given report was about was hard.					
(2) I already learned something about the topic by reading the report.					
(3) Evaluating the report is easy.					
(4) The report helped to avoid redundant searching.					
(5) The amount of information of the given report was confusing.					
(6) The amount of information of the given report was helpful.					
(7) It was helpful / interesting to look through the report of someone else.					
(8) It was fun / motivating to look through the report of someone else.					
(9) The report helped me to dive into the given topic faster than without.					
(10) Evaluating the given report helped to save a lot of time.					

(11) The given report can be extended by important information.					
(12) What was your strategy to excerpt the information from the given report?					

R – Extending the report	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) Did you find additional relevant information?					
(2) I could use many facts from the given report for my report on the topic.					
(3) I could find facts of the search topic that my colleague did not already cover.					
(4) I could improve or extend the given report.					
(5) Compared to my first search, it was easier to write this report.					
(6) My colleague and I have produced a sufficient result on this topic. **(Collaboration)**					
(7) SearchTrails enhances collaboration between colleagues / teammates. **(Collaboration)**					

S – SearchTrails – Features

Features / Opinion	I used this feature.	How important was the feature during your search?	This feature is a good idea, but a bad implementation.
	Check / leave blank	Please rate with a number from 1 'very important' to 5 'unimportant/useless'	Check / leave blank
Search trail visualization as a graph			
Keyword table			
Select and Search for keywords from the keywords table			
Blue Nodes / Highlight pages in the graph			
Red nodes / Irrelevant pages in the graph			
Hulls around nodes / Clusters			
Collapsing clusters			
Deleting nodes			

Continuing a search by entering the trails ID			
Adding highlights by pressing 'ALT'			
Adding handwritten highlights			
Deleting highlights			
Marking pages irrelevant by pressing '-' (minus)			
Clicking on URLs/Links connected to the nodes in the visualization to get back to a page			

T – Self-Awareness Evaluation	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) Sometimes, I visited pages that were irrelevant for my search (e.g. news pages).					
(2) The need to create the report made my search more focused.					
(3) Seeing the search graph helped me to be aware of my search results.					
(4) I became aware of pages I visit frequently.					
(5) List 3 insights that you gained when you were monitoring your own search behavior: 1) 2) 3)					

U – Aggregation Evaluation	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) At least once I felt overwhelmed by the amount of new information during my search.					
(2) If you agree: How often (approximately)?	__________ times				
(3) The graph visualization behaved like expected.					
(4) The search graph obviously resembles the course of my search.					
(5) Seeing the course of my search was interesting.					
(6) It was interesting to see all visited pages on a glance.					

(7) It was interesting to see how I came to search results and where I went thereafter.					
(8) The search trail helped to see all pages where I found relevant information for my search.					
(9) The search graph visualization helps to remember relevant facts of the search topic.					
(10) At least once I wanted to get back to pages visited earlier.					
(11) SearchTrails helped me to get back to previous pages.					
(12) At least once I had the feeling to be moving in circles, often seeing the same pages.					

V – Synthesis Evaluation	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) SearchTrails was helpful to capture relevant search results.					
(2) It felt safe to capture search results.					
(3) I looked into the search highlights after the actual search.					
(4) The highlights helped to gather all relevant resources for the report.					
(5) Being able to interact with the graph and its highlights helped to condense the relevant information.					
(6) At least once I forgot to highlight things that turned out to be important later.					
(7) I had SearchTrails and the tabs used for browsing both always visible (e.g. on two screens).	☐ Yes			☐ No	
(8) Switching between the SearchTrails-Tab and the tabs used for browsing was cumbersome.					

W – Discovery Evaluation	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) I think I have found all relevant facts that can be found on my search task.					
(2) At least once I got stuck during my search.					
(3) If you agree: How often?	__________ times				
(4) SearchTrails enabled me to discover additional facts for my search task.					
(5) If you agree, why?					

(6) The keywords column helped to see the context in which I was browsing.					
(7) The keywords helped to find relevant search terms that I would not have come up with.					

X – Improving motivation	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) Searching with SearchTrails was motivating.					
(2) I searched longer with SearchTrails than I would have searched without the tool.					
(3) Using SearchTrails made me more efficient than I would have been without the tool.					

Y – Can you think of more application cases that SearchTrails is suited for?

Z – General comments or remarks on the second phase of the evaluation:

1 – Lightweightness	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) The concept of SearchTrails is immediately clear.					
(2) SearchTrails is a lightweight tool.					
(3) The amount of features is sufficient.					

2 – General questions	Strongly agree.	Agree.	Neither agree nor disagree.	Disagree.	Strongly disagree.
(1) I liked working with SearchTrails.					
(2) SearchTrails has an innovative concept.					
(3) I felt supported well by search engines during the **first** search task.					
(4) I felt supported well by search engines during the **second** search task.					
(5) SearchTrails is suited for handing over complex search results.					
(6) I would like to use SearchTrails for my own searches in the future.					

(7) I would recommend SearchTrails to my friends.					
(8) I have an idea on what could also be done with SearchTrails:					
(9) Are there features missing? If yes, which ones:					

3 – This is your chance to comment on your general impression of SearchTrails. It would help a lot if you could comment on everything that was not asked in the previous questions.
Some example questions may help you, but should not limit you: What did you like about SearchTrails, what could be better? Is something not necessary, or is something missing?

Thanks a lot for your support!

A3 – Search trail visualizations from the second user study

I present here some exemplary search trails from the first phase of second user study which have been produced by the study participants with the support of SearchTrails. For convenience reasons, all nodes from the cluster 'Google' have been reduced to a single node, to better see the actual result pages.

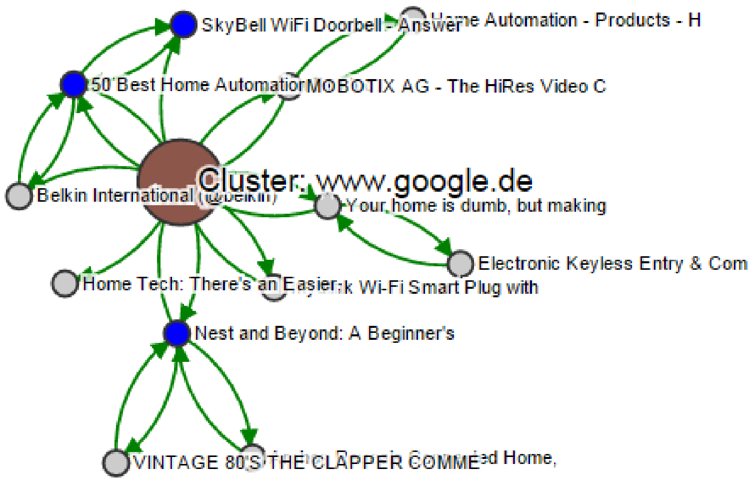


Figure 96: Search trail from user 05 on the topic 'Home automation' from the first phase.

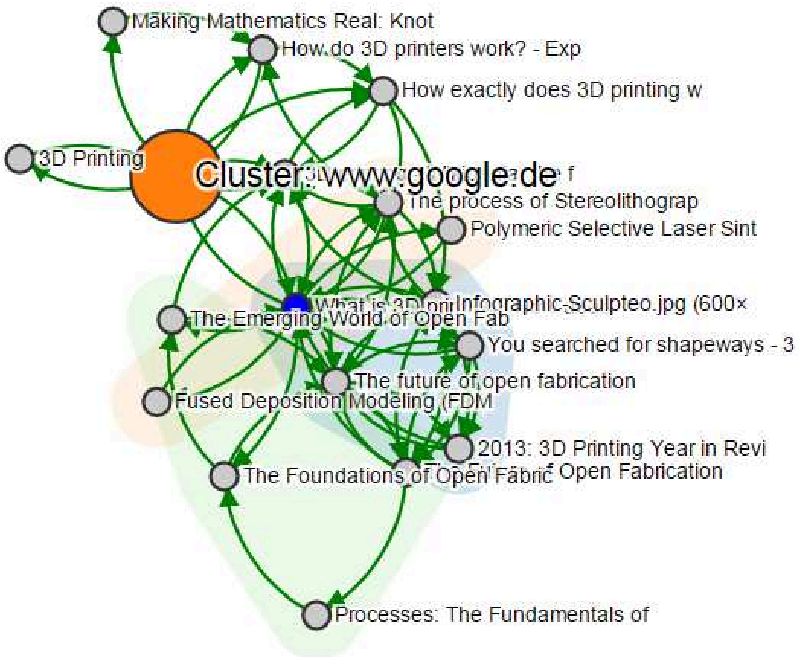


Figure 97: Search trail from user 14 on the topic '3D printing' from the first phase.

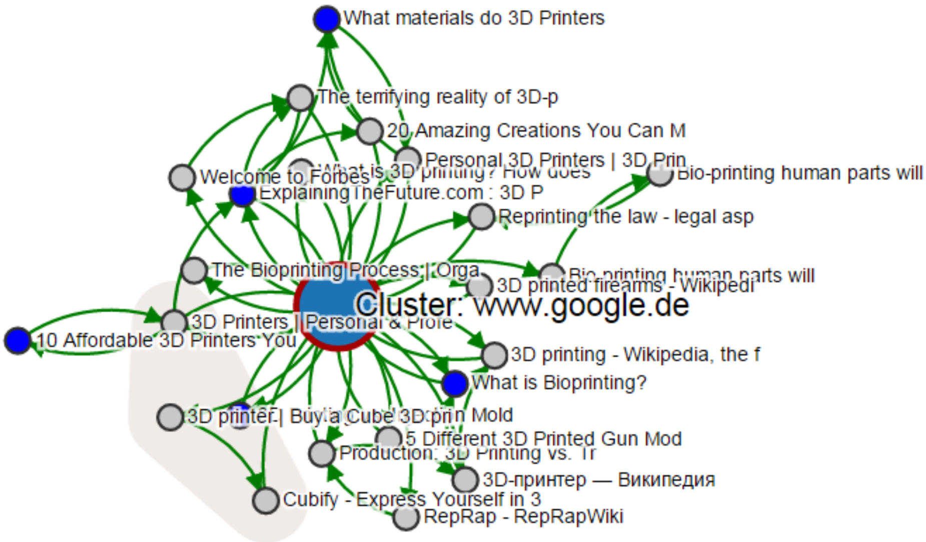


Figure 98: Search trail from user 18 on the topic '3D printing' from the first phase.

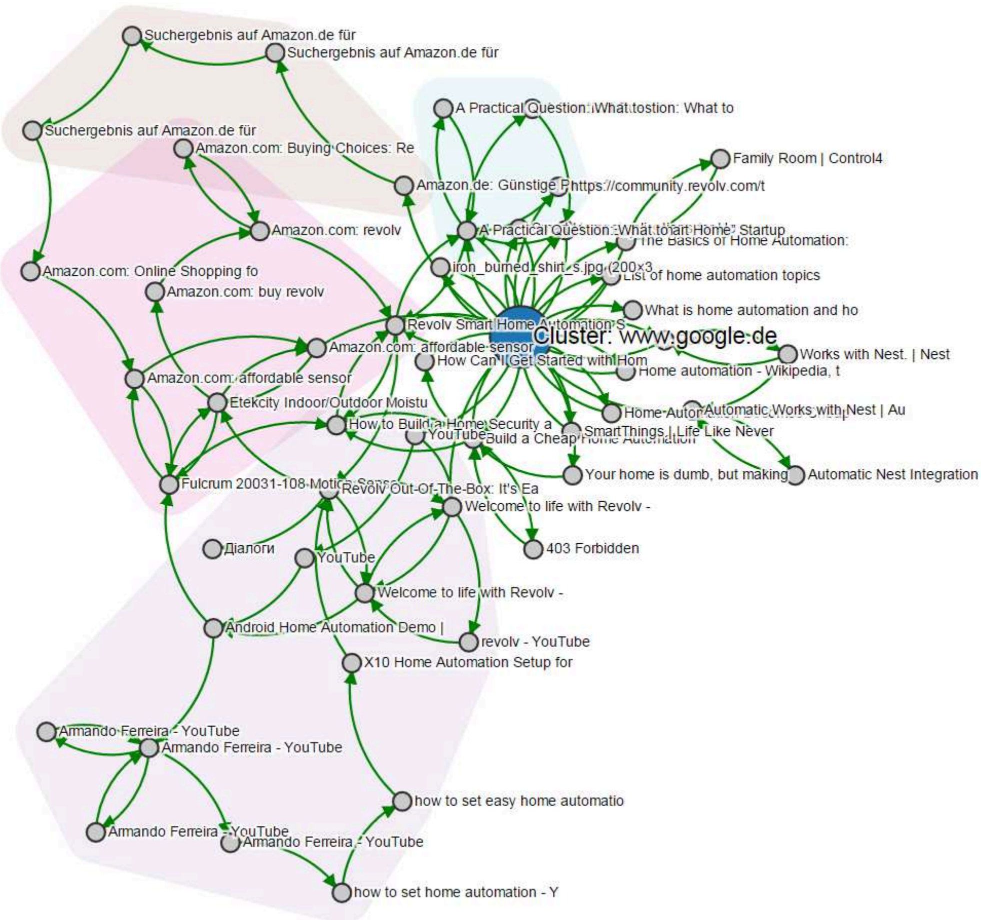


Figure 99: Search trail from user 25 on the topic 'Home automation' from the first phase.

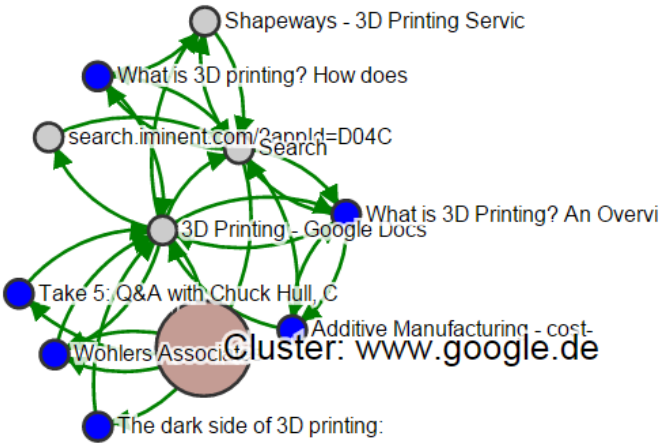


Figure 100: Search trail from user 26 on the topic '3D printing' from the first phase.

I present here some exemplary search trails from the second phase of second user study which have been produced by the study participants with the support of SearchTrails. For convenience reasons, all nodes from the cluster 'Google' have been reduced to a single node, to better see the actual result pages.

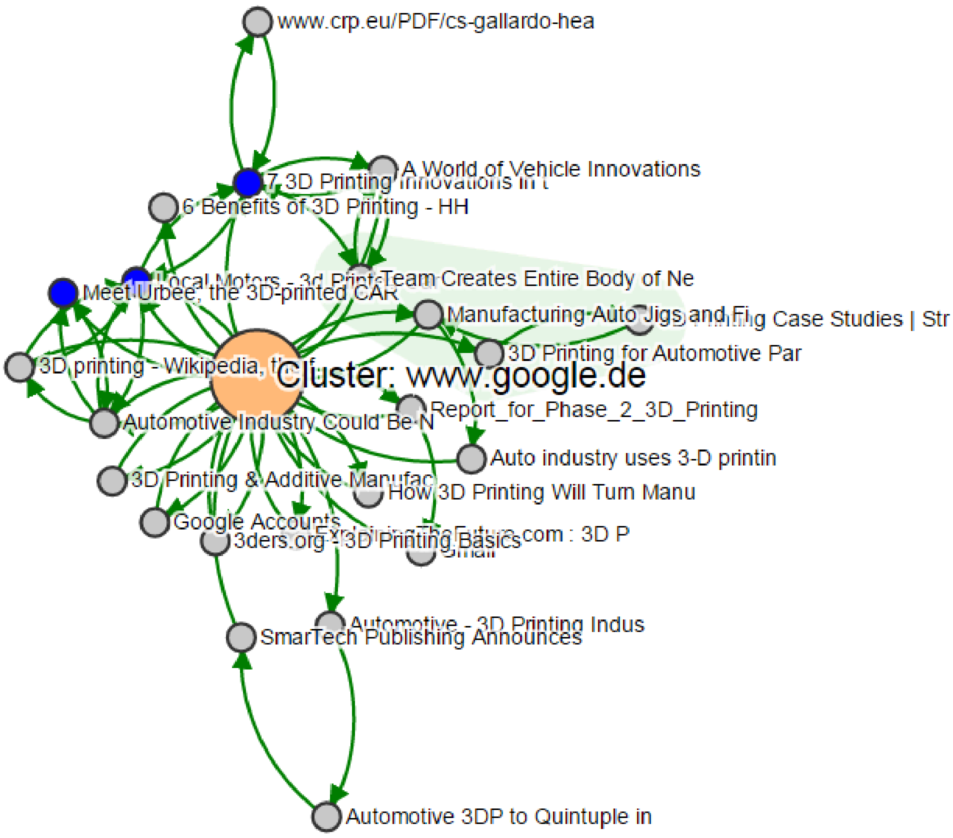


Figure 101: Search trail from the second phase from user 05 on the topic '3D printing' with a report as given artifact.

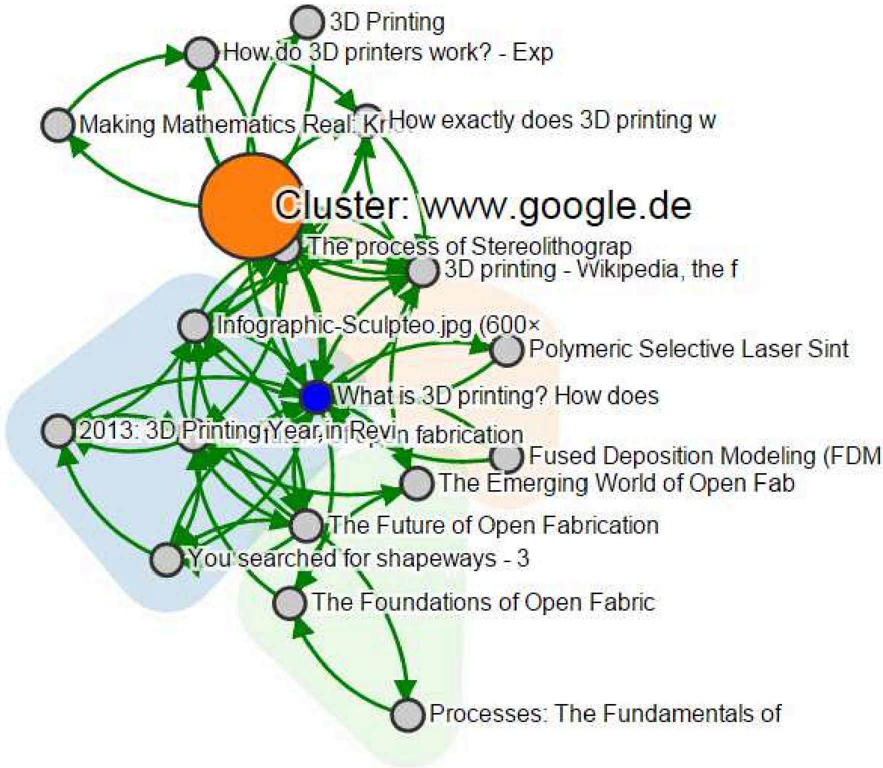


Figure 102: Search trail from the second phase from user 07 on the topic '3D printing' with a search trail as given artifact.

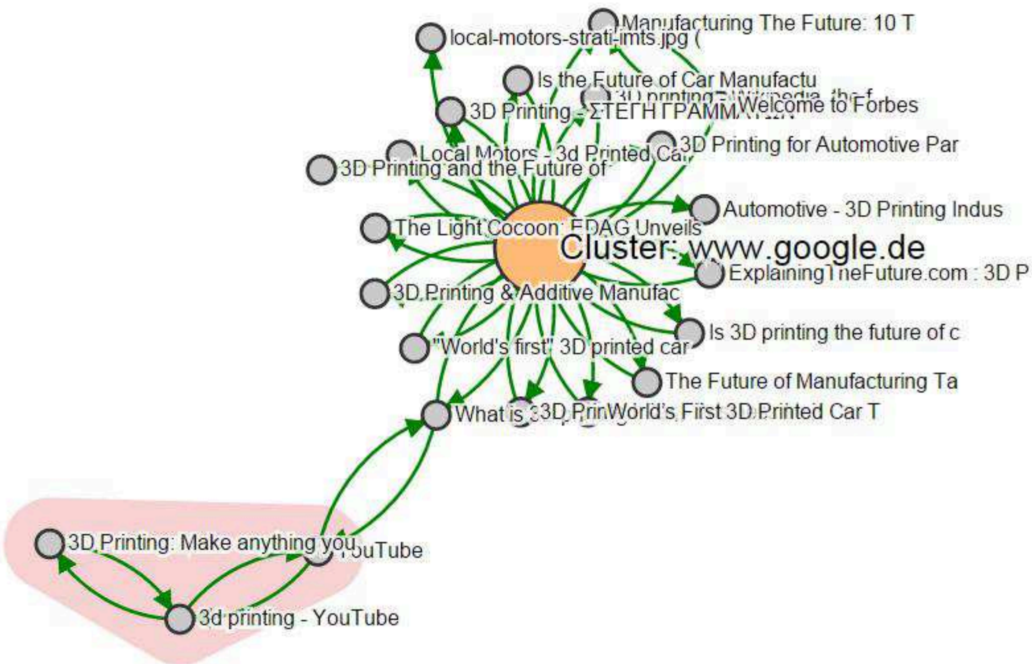


Figure 103: Search trail from the second phase from user 09 on the topic '3D printing' with a report as given artifact.

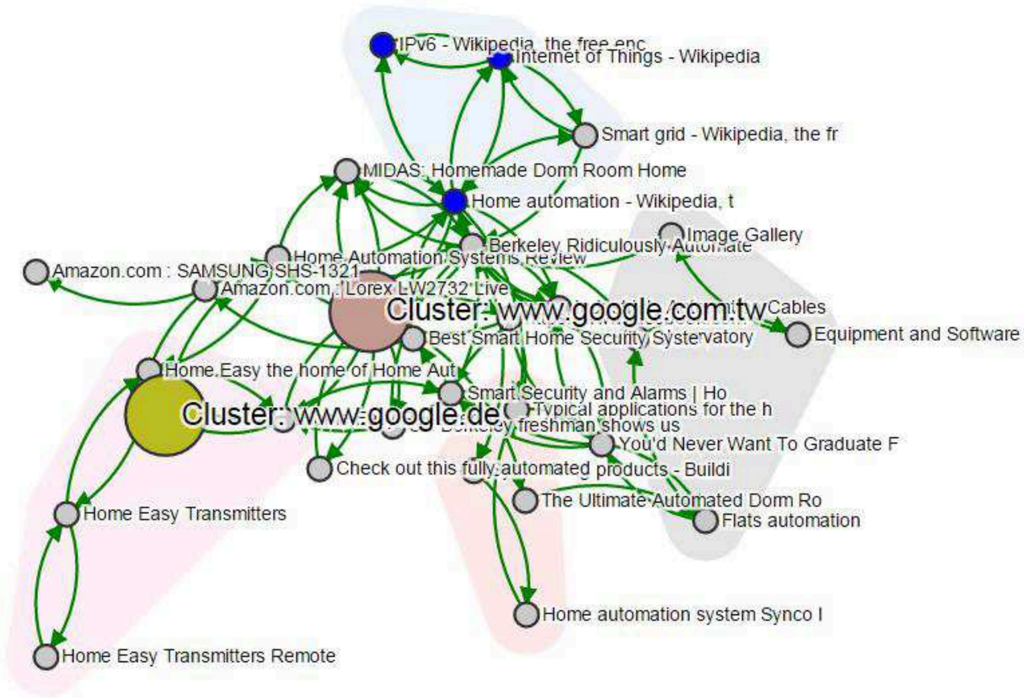


Figure 104: Search trail from the second phase from user 12 on the topic 'Home automation' with a search trail as given artifact.

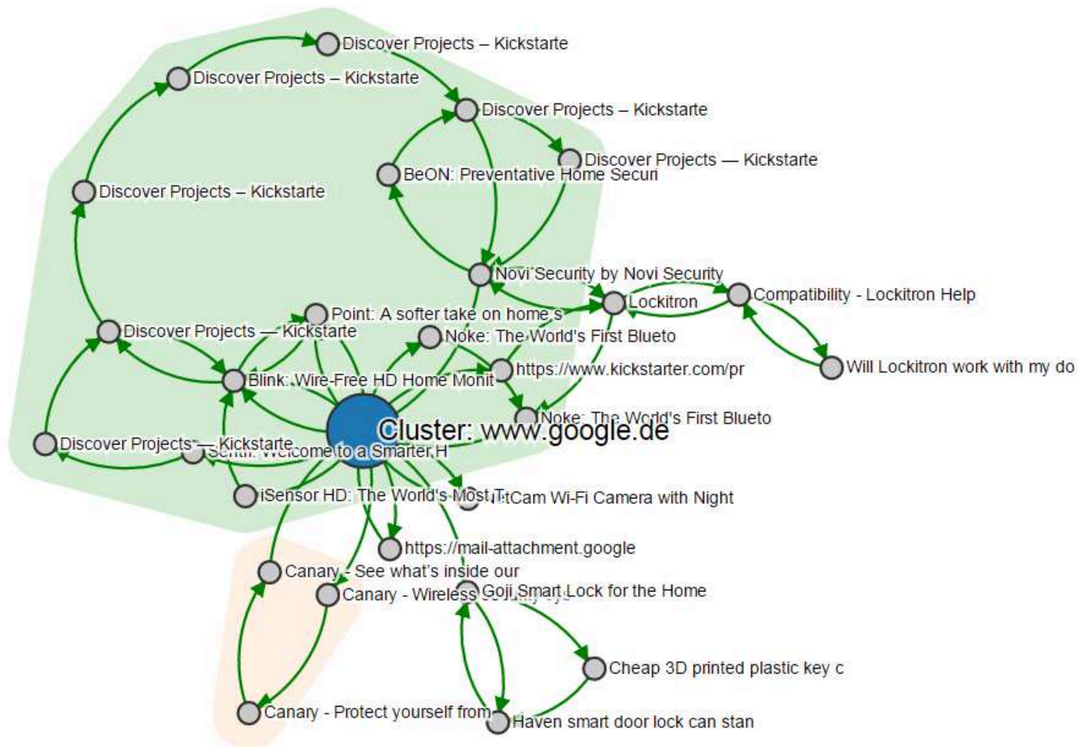


Figure 105: Search trail from the second phase from user 22 on the topic 'Home automation' with a report as given artifact.

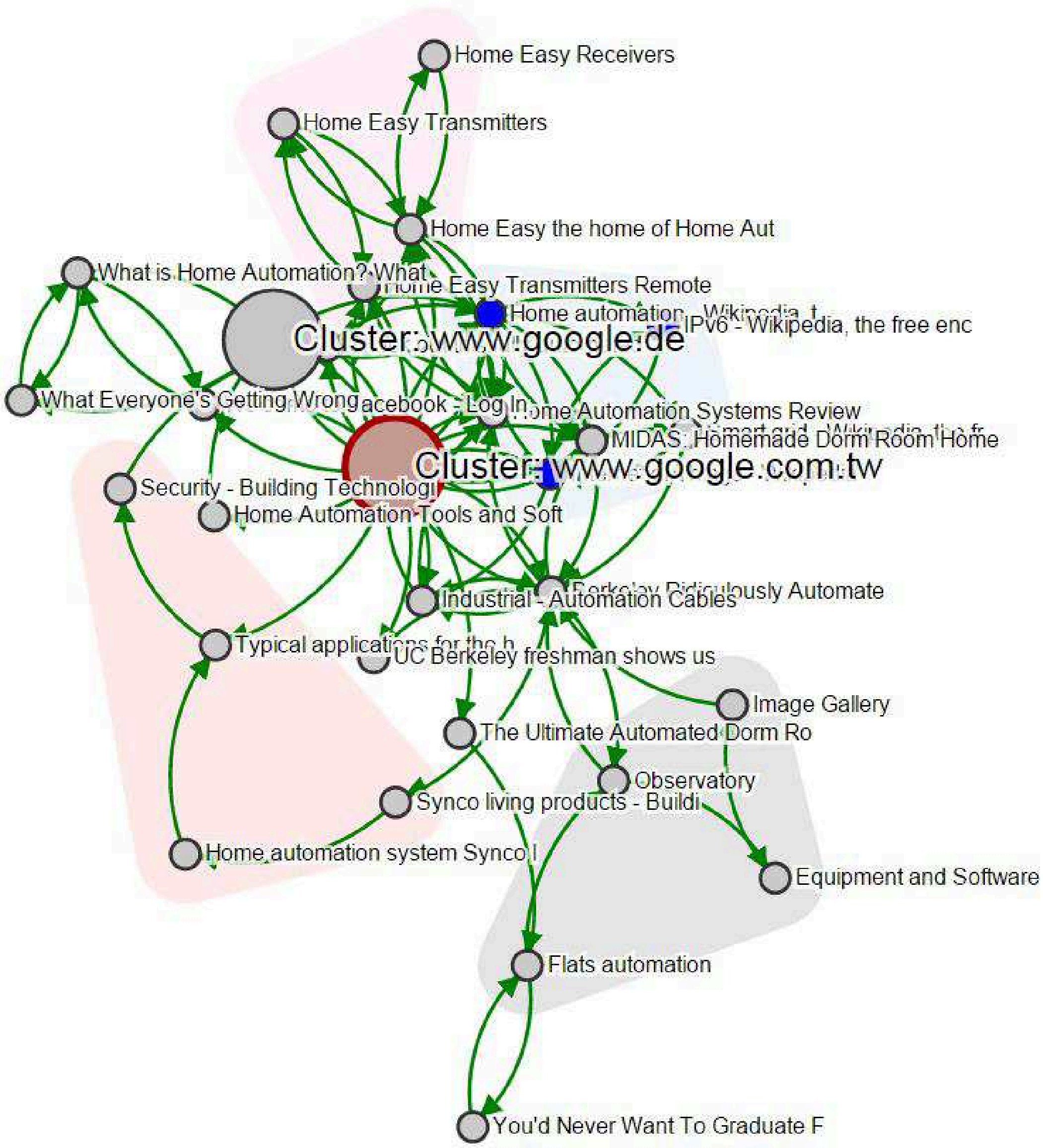


Figure 106: Search trail from the second phase from user 24 on the topic 'Home automation' with a search trail as given artifact.

Curriculum vitae

General

Name	Sebastian Andreas Franken
Academic degrees	Diplom-Informatiker Diplom-Wirtschaftsinformatiker
Date of birth	August 18, 1984
Place of birth	Geilenkirchen, Germany
Address	Fraunhofer Institute for Applied Information Technology FIT Schloss Birlinghoven 53754 Sankt Augustin Germany
Phone	0049 2241 14 2416

Education

Sep 2008 – Jul 2010	Second degree Study Programme on Business Informatics at RWTH Aachen University **Diploma thesis** *'Performance pay in family-owned firms'* under guidance of Professor Dr. Alwine Mohnen. Overall diploma grade 'Good'
Oct 2003 – Feb 2009	University studies of Computer Science at RWTH Aachen University **Diploma thesis** *'Conception and evaluation of a web-based faceted browsing application for a groupware system'* under guidance of Professor Wolfgang Prinz, Ph. D. Overall diploma grade 'Very good'
Aug 1994 – Jun 2003	High school 'Bischöfliches Gymnasium St. Ursula, Geilenkirchen'. Intensive courses in Mathematics and Physics.
Sep 1990 – Jun 1994	Elementary school 'Katholische Grundschule Birgden'

Professional experience

Oct 2013 - date	Researcher at the Fraunhofer Institute for Applied Information Technology FIT • Worked on the EU projects GreenCom, IMPReSS, and CloudTeams.

- Developed power grid visualization for a Danish distribution system owner.
- Developed a sensor supervision interface for a Brazilian opera house.
- Developed a Software-as-a-Service platform for enhancing collaboration between software developers and customers.
- Participant of the Fraunhofer mentoring programme 'StepForward'.
- Supervision of Master theses and student practical courses.
- Review for a number of conferences related to CSCW.
- Development and evaluation of the SearchTrails application.
- Certified Usability Engineer in January 2016.

Sep 2010 – Sep 2013 Researcher at the
Chair of Computer Science 5 - Information Systems

- Worked on the project 'SurgeryNet' by the German Federal Ministry of Education and Research.
- Developed a Software-as-a-Service platform for the further education of surgeons.
- Supervision of Master theses and student practical courses.

Publications related to the thesis

Franken, S. & Norbisrath, U. (2014a). Supporting the evaluation of complex search tasks with the SearchTrails tool. In: Proceedings of 24th Annual International Conference on Computer Science and Software Engineering. 2014, Toronto, Canada: IBM Corp., pp. 262–274.

Franken, S. & Norbisrath, U. (2014b). Trail Building During Complex Search Tasks. In: A. Butz, M. Koch & J. Schlichter (eds.). Mensch & Computer 2014 - Tagungsband. 2014, München, Germany: De Gruyter Oldenbourg, pp. 135–144.

Franken, S., Norbisrath, U. & Prinz, W. (2015c). Search Trails as Collaboration Artifacts – Evaluating the UX. In: S. Diefenbach, N. Henze & M. Pielot (eds.). Mensch und Computer 2015 – Proceedings. 2015, Berlin, Germany: De Gruyter Oldenbourg, pp. 23–32.

Franken, S., Norbisrath, U. & Prinz, W. (2016). The Impact of SearchTrails on the Quality of Collaborative Search. In: W. Prinz, J. Borchers, M. Jarke & M. Ziefle (eds.). Mensch und Computer 2016 – Proceedings. 2016, Aachen, Germany: De Gruyter Oldenbourg.

Further publications

Franken, S. & Prinz, W. (2009a). FacetBrowse: Ein facettenbasierter Dateibrowser für ein Groupware-System. In: S. Kain, D. Struve, & H. Wandke (eds.). Workshop-Proceedings der Tagung Mensch & Computer 2009. 2009, Berlin, Germany: Logos Verlag, pp. 214–216.

Franken, S. & Prinz, W. (2009b). FacetBrowse: Facettenbasiertes Browsen im Groupware Kontext. In: H. Wandke, S. Kain, & D. Struve (eds.). Wandke, H., Kain, S., Struve, D. Mensch & Computer. 2009, pp. 123–132.

Franken, S. & Jeners, N. (2011). Challenges of Social Software in Clinical Environments. In: ECSCW 2011 Workshop Proceedings. 2011, Aarhus, Denmark: Springer.

Franken, S. & Jeners, N. (2012). SurgeryNet-Eine Kooperationsplattform zur individuellen Aus- und Weiterbildung in der Chirurgie. In: Proceedings of the 11th Annual meeting of the German Society for Computer- and Robot Assisted Surgery. 2012, Duesseldorf, Germany, pp. 79–82.

Franken, S., Kolvenbach, S. & Prinz, W. (2013). Social Media integrated into Groupware. In: ECSCW 2013 Adjunct Proceedings. 2013, Paphos, Cyprus.

Jeners, N., Prinz, W. & Franken, S. (2013). A Meta-Model for Cooperation Systems. In: L. M. Camarinha-Matos & R. J. Scherer (eds.). Collaborative Systems for Reindustrialization. [Online]. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 239–246. Available from: http://link.springer.com/10.1007/978-3-642-40543-3_26. [Accessed: 24 September 2015].

Franken, S., Prinz, W. & Jeners, N. (2014). Making Groupware Social: The Case of a Cooperative Platform for Surgeons. In: Proceedings of the International Conference on Intelligent Networking and Collaborative Systems. [Online]. September 2014, Salerno, Italy: IEEE, pp. 49–56. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7057069. [Accessed: 24 September 2015].

Franken, S., Kolvenbach, S. & Prinz, W. (2015a). CloudTeams: Enabling Tight Collaboration between Customers and Software Developers. In: Poster presentation at the European Conference of Computer Supported Cooperative Work (ECSCW). Unpublished.

Franken, S., Kolvenbach, S., Prinz, W., Alvertis, I. & Koussouris, S. (2015b). CloudTeams: Bridging the Gap Between Developers and Customers During Software Development Processes. In: Procedia Computer Science. [Online]. 2015, Pisa, Italy, pp. 188–195. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1877050915030793. [Accessed: 5 November 2015].