Visualization of Search Results from the World Wide Web

Dissertation
zur Erlangung des akademischen Grades
des Doktors der Naturwissenschaften (Dr. rer. nat.)
an der Universität Konstanz
Fachbereich Informatik und Informationswissenschaft
vorgelegt von

Thomas M. Mann

in conformity with the requirements for the degree of
Doctor of Natural Sciences (Dr. rer. nat.)
at the University of Konstanz, Germany
Department of Computer and Information Science

Tag der mündlichen Prüfung: 10. Januar 2002
Referent: Prof. Dr. Harald Reiterer
Referent: Prof. Dr. Wolfgang Pree

Day of the oral examination: 2002-01-10

Visualization of Search Results from the World Wide Web

Thomas M. Mann

Dissertation, Freiburg im Breisgau 2002

University of Konstanz, Germany, Department of Computer and Information Science

Universität Konstanz, Fachbereich Informatik und Informationswissenschaft

Abstract (English)

This thesis explores special forms of presentations of search results from the World Wide Web. The usage of Information Visualization methodologies is discussed as an alternative to the usual arrangement in form of a static HTML-list. The thesis is structured into four main parts. The first part deals with information seeking. It presents ideas from the literature on how to structure the information seeking process and some results from studies of how people search the Web. For the second part visualization ideas, metaphors, techniques, components and systems have been collected. The overview focuses on the visualization of queries or query attributes, document attributes, and interdocument similarities. The reference model for visualization from [Card, Mackinlay, Shneiderman 1999] is used to discuss differences between certain techniques. Visualization components from a number of areas, usage scenarios, and authors are presented using a consistent search example wherever possible. The part about Information Visualization also includes a discussion of multiple coordinated views and some results from empirical evaluations of visualizations by other authors. The third, empirical part of the thesis presents the results of an evaluation of five different user interface conditions of a local meta search engine called INSYDER. An overview covering the INSYDER project in general, the system architecture, and the development of the implemented visualization ideas is included. In a test with 40 users effectiveness, efficiency, expected value, and user satisfaction were measured for twelve tasks. Evaluated user interface conditions were HTML-List, ResultTable, ScatterPlot plus ResultTable, BarGraph plus ResultTable, and SegmentView plus ResultTable. The SegmentView included TileBars and StackedColumns variants. The traditional presentation in the form of an HTML-List performed best in terms of effectiveness and efficiency. In contrast to this, the users preferred the ResultTable and the SegmentView. The last section of the thesis consists of a summary and an outlook.

Abstract (Deutsch)

Diese Dissertation untersucht spezielle Formen der Darstellung von Suchergebnissen aus dem World Wide Web. Diskutiert wird die Nutzung von Methoden der Informationsvisualisierung als Alternative zur üblichen Anordnung in Form einer statischen HTML-Liste. Die Arbeit ist in vier Hauptteile strukturiert. Der erste Teil beschäftigt sich mit der Informationssuche. Er stellt Ideen aus der Literatur vor wie der Suchprozess strukturiert werden kann, sowie einige Resultate aus Studien wie Benutzer im Web suchen. Für den zweiten Teil wurden Ideen, Metaphern, Techniken, Komponenten und Systeme für Visualisierungen gesichtet. Der Überblick ist ausgerichtet auf die Visualisierung von Abfragen oder Abfrageattributen, von Dokumentattributen und von Ähnlichkeiten zwischen Dokumenten. Das Referenzmodell für Visualisierung von [Card, Mackinlay, Shneiderman 1999] wird verwendet, um Unterschiede zwischen bestimmten Techniken zu diskutieren. Visualisierungskomponenten aus bzw. von einer Anzahl von Bereichen, Anwendungsszenarien und Autoren werden dargestellt, indem wo immer möglich ein konsistentes Suchbeispiel verwendet wird. Der Abschnitt über Informationsvisualisierung umfasst auch eine Diskussion über mehrfache, koordinierte Ansichten und einige Resultate aus empirischen Untersuchungen von Visualisierungen durch andere Autoren. Der dritte, empirische Teil der Dissertation stellt die Resultate einer Untersuchung von fünf unterschiedlichen Darstellungs-Szenarien einer lokalen Metasuchmaschine mit dem Namen INSYDER vor. Enthalten ist auch ein Überblick über das INSYDER-Projekt im allgemeinen, die Systemarchitektur und die Entwicklung der umgesetzten Visualisierungen. In einem Test mit 40 Benutzern wurden Effektivität, Effizienz, erwarteter Nutzen und Benutzer-Zufriedenheit für zwölf Aufgaben gemessen. Untersuchte Präsentationsformen waren HTML-Liste, ResultTable, ScatterPlot plus ResultTable, BarGraph plus ResultTable und SegmentView plus ResultTable. Die SegmentView bestand aus TileBar- und StackedColumn-Varianten. Die traditionelle Darstellung in der Form einer HTML-Liste zeigte die besten Ergebnisse bezüglich Effektivität und Effizienz. Im Gegensatz dazu bevorzugten die Benutzer die ResultTable und die SegmentView. Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick.

Zweiseitige Zusammenfassung in Deutsch

Für eine Kurzübersicht über den Inhalt der Arbeit siehe den vorangegangenen deutschen Abstract. Auf den folgenden zwei Seiten erfolgt eine kurze Darstellung der Inhalte der einzelnen Kapitel. Die Einleitung (Introduction) umreißt das Aufgabenfeld Suchen im Web und thematisiert hier insbesondere Informationsüberflutung und Selektion. Als mögliche Lösung von Problemen wird der Einsatz von Techniken der Informationsvisualisierung vorgeschlagen.

Das Hauptkapitel zum Thema Informationssuche (Information seeking) gliedert sich zwei Teile. Nach einer kurzen Darstellung der Unterschiede zwischen Suchprozessen im Web und klassischem Information Retrieval werden Ideen aus der Literatur vorgestellt, wie der Suchprozess strukturiert werden kann. In einem zweiten Teil werden einige Resultate präsentiert wie Benutzer im Web suchen. Die Diskussion von möglichen Strukturierungsansätzen für Suchprozesse präsentiert im wesentlichen Modelle, die im Zusammenhang mit klassischem Information Retrieval entwickelt wurden. Besonderheiten des Suchens im World Wide Web werden dargestellt. Die Diskussion der möglichen Strukturierungsansätze ist gegliedert in drei Granularitätsstufen: a) generelle Ziele, Aufgaben und Strategien, b) Funktionen, Phasen und Schritte des Suchprozesses, sowie c) Detailaufgaben, -ziele und Bedienschritte. Ausgewählt werden mit dem task actions model, dem four-phase framework of information seeking und der TTT data type by task taxonomy drei Ansätze von Shneiderman. Da sich die Darstellungen der Arbeit im wesentlichen auf den Ebenen a) und b) bewegen, spielt die TTT data type by task taxonomy im weiteren Verlauf nur eine untergeordnete Rolle. Die Aufnahme erfolgte zur Abrundung des Gesamtbildes. Nach der theoretischen Auseinandersetzung mit dem Suchprozess erfolgt ein Blick auf empirische Ergebnisse zum realen Suchverhalten. Im Abschnitt zur Frage „wie suchen Benutzer im Web“ werden im wesentlichen die Ergebnisse aus vier Studien vorgestellt, in denen Protokolldateien großer Suchmaschinen analysiert wurden. Es handelt sich dabei um die Excite-Studie von [Jansen, Spink, Bateman et al. 1998], die AltaVista-Studie von [Silverstein, Henzinger, Marais et al. 1999], die 1998er Fireball-Studie von [Hölscher 1998] und die 1999er Fireball-Studie von [Röttgers 1999]. Wichtigste Ergebnisse: eine Suchanfrage enthält im Schnitt etwa zwei Suchbegriffe und die Benutzer gehen nur selten über die erste Ergebnisseite mit zehn Treffern hinaus. Das Kapitel schließt mit einigen Ergebnissen zu Unterschieden bei der Web-Suche zwischen Benutzergruppen.

Nach einer knappen Darstellung der Aufgaben der Informationsvisualisierung (Information Visualization) beginnt das Kapitel mit der Vorstellung eines Referenzmodells von [Card, Mackinlay, Shneiderman 1999]. Die Autoren strukturieren hier den Prozess der Abbildung von Ausgangsdaten über Datentabellen und visuelle Strukturen zu den Ansichten, die der Benutzer letztendlich auf dem Schirm präsentiert bekommt. Das Modell wird im weiteren Verlauf der Arbeit benutzt, um Technikübersichten zu strukturieren, bestimmte Einzelaspekte einzuordnen oder die Datenabbildungen im System INSYDER zu erläutern. Großen Raum in der Arbeit nimmt die Darstellung der Möglichkeiten der Informationsvisualisierung dar. Die Übersicht ist fokussiert auf die Darstellung von Suchergebnissen und beleuchtet das Thema von mehreren Seiten. Als Einstieg wurde der Aspekt der Metaphern gewählt, die ja normalerweise auch dem Benutzer den Zugang zu einem System erleichtern sollen. Es folgt ein Abschnitt, der auf abstraktem Niveau Techniken beschreibt, die im Rahmen der Informationsvisualisierung genutzt werden. Anschließend werden, unter Verwendung eines wo immer möglich durchgehend einheitlichen Beispiels, zahlreiche Ideen präsentiert wie Suchergebnisse visualisiert werden können. Die komponentenorientierte Darstellung ist gegliedert in die Visualisierung von Abfragen oder Abfrageattributen, die Visualisierung von Dokumentattributen und die Visualisierung von Ähnlichkeiten zwischen Dokumenten. Zum Themenbereich Visualisierung von Beziehungen zwischen Dokumenten wird auf andere Arbeiten verwiesen. Die Betrachtung aus unterschiedlichen Blickwinkeln wird abgeschlossen durch eine strukturierte Auflistung der erwähnten Systeme. Es folgt eine Auseinandersetzung mit mehrfachen, koordinierten Ansichten und der Frage, wann und wie solche Konzepte einzusetzen sind. Das Kapitel zum Thema Informationsvisualisierung wird beendet mit der Präsentation einiger Resultate aus empirischen Untersuchungen zum Nutzen ausgewählter Visualisierungsansätze und unter dem Stichwort „5T-Environment“, einer Zusammenfassung von Faktoren, die den Nutzen von Visualisierungen beeinflussen.

Der empirische Teil der Arbeit beginnt mit einer Beschreibung des Projektes INSYDER, in dessen Rahmen die Software entwickelt wurde, die bei der Evaluierung verschiedener Darstellungsformen von Suchergebnissen eingesetzt wurde. Beschrieben werden die Funktionen des Systems im Allgemeinen, seine Softwarearchitektur, die Funktionen der einzelnen Softwaremodule, der prototypengestützte Entwicklungsprozess und erste formative Evaluationen während des Projektes. Es folgt eine ausführliche Darstellung der implementierten Visualisierungen sowie des konkreten Abbildungsprozesses von den Ausgangsdaten zu Ansichten. Hierbei werden auch Probleme thematisiert, die im Rahmen dieses Prozesses auftraten, sowie verschiedene Visualisierungen, die aus unterschiedlichen Gründen in der endgültigen Softwareversion nicht umgesetzt wurden. Die Diskussion der durchgeführten Evaluation beginnt mit einer Beschreibung der Hypothesen und Variablen, sowie des Versuchsablaufs. Untersucht wurden Effektivität, Effizienz, erwarteter Nutzen und Benutzer-Zufriedenheit für die Präsentationsformen HTML-Liste, ResultTable, ScatterPlot plus ResultTable, BarGraph plus ResultTable und SegmentView plus ResultTable. Der Test wurde mit 40 Benutzern und jeweils zwölf Aufgaben im Frühjahr 2000 an der Universität Konstanz durchgeführt. Unabhängige Variablen waren Präsentationsform, Benutzergruppe (Anfänger / Experte), Anzahl der Suchbegriffe (1 / 3 / 8), Anzahl der als Ergebnis präsentierten Dokumente (30 / 500) und Art der Aufgabe (Finden spezifischer Fakten / erweitertes Finden von Fakten). Die Fragebogenauswertung ergab, dass die Benutzer zwar an verschiedenen Stellen Probleme mit der Benutzbarkeit der Visualisierungen hatten, ganz generell aber die Möglichkeiten sehr begrüßten, die von der ResultTable und den Visualisierungen geboten wurden. Die Unterschiede in der Einschätzung zwischen Anfängern und Experten waren gering und bezogen sich, wenn überhaupt, meist auf den ScatterPlot. Wenn positive und negative Bewertungen zusammengefasst werden, schneiden die ResultTable und die SegmentView besser ab als die HTML-Liste. Der BarGraph und speziell der ScatterPlot schneiden schlechter ab als die HTML-Liste. Beim Vergleich von subjektiven Einschätzungen und ermitteltem Erfolg der Komponenten muss beachtet werden, dass im Fragebogen nach den einzelnen Komponenten gefragt wurde, im Versuch für die drei echten Visualisierungen aber immer zusätzlich die ResultTable zur Verfügung stand und von den meisten Probanden auch genutzt wurde. Von einigen sogar mehr als die eigentliche Visualisierung. Bezüglich Effektivität, Aufgabenerledigungszeit und Effizienz zeigte die traditionelle Darstellung in Form einer HTML-Liste generell die besten Werte.

Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick (Summary and Outlook) in dem auch weitergehende Evaluationen der bestehenden Komponenten und veränderte Visualisierungsansätze in Form einer SuperTable und eines verbesserten ScatterPlots diskutiert werden.

Acknowledgements

It must have been at the end of the 1980s when I was studying Information Science at the University of Konstanz (Germany) that I first had the idea of getting a doctors degree. Information Science in Konstanz is unthinkable without Prof. Dr. Rainer Kuhlen, whom I have to thank for my choice of Information Science. For a number of reasons it took several years until I really started to take the next step in the direction of a doctor's degree. Special thanks to Dr. Wolf R. Dombrowsky from the Katastrophenforschungsstelle at the Christian-Albrechts-Universität zu Kiel (Germany), who encouraged me in the mid 1990s to turn my vague plans into a real project.

From all the people who contributed in one or another way to me finishing the project, I have to thank first of all my advisor Prof. Dr. Harald Reiterer, who gave me the chance to be the first member in his team at the University of Konstanz in 1997. He motivated, stimulated and supported me for more than four years in the finally successful project. Without him this thesis would never have been possible. Also thanks to Prof. Dr. Wolfgang Pree from the University of Konstanz for his interest in my work and for taking over the role of a second referee.

From my colleagues I have to thank especially my "doctor sister" Gabriela Mußler, who, together with Harald, was the driving force behind the University of Konstanz getting the chance to participate in the INSYDER project. This was the basis for my evaluation work. Congratulations to her for finishing her thesis about the INSYDER project at her new home in Penistone, Sheffield (England). Also special thanks to Siegfried Handschuh. As chief architect of contributions from the University of Konstanz for the INSYDER system development he was one of the main factors helping to turn my visualization ideas into concrete pieces of software. In this context I also have to thank Georg Odenthal from the University of Konstanz and Laurent Dosdat from Arisem S.A. Paris (France) for the time they spent developing the INSYDER system.

At the University of Konstanz a number of additional colleagues and students helped to work on the INSYDER project, the evaluation performed, or my work in general. Just to list the most important ones: Dagmar Michels, Dr. Marc Rittberger, Dr. Wolfgang Semar, Dr. Ulrik Brandes, Ersin Kurun, and especially the evaluation team Dietmar Ohlmann, Edgar Fiederer, Edgar Sprethuber, Joachim Griesbau, and Ludmilla Bernet. Thanks also to the forty participants of the study.

From the other members of the INSYDER team I want to especially thank Alain Garnier, Olivier Spinelli and Jean Ferrè from Arisem, Rina Angeletti from Innova (Rome, Italy), Flavia D'Auria from Promoroma (Rome, Italy), Carlo Revelli and Guillaume Lory from Cybion (Paris, France) and last but not least the European Commission DG III (Brussels, Belgium) with Patrick Corsi who funded the project.

Thank you to John V. Cugini, Dr. Christoph Hölscher, Heike Röttgers and other of the above mentioned for providing me with helpful material.

Special thanks to Dr. Bertrand Lisbach from Basel (Switzerland) for support in statistics, as well as to Charlie Smith, Malcolm MacLaren, Beate Heckner, and a number of other people for reading preliminary versions of the thesis and providing me with helpful advice.

Last but not least I want to thank Veronika for years of love, patience, and support.

Contents

1. Introduction 11
1.1. Problem 11
1.2. Solution 14
1.3. Structure of the Thesis 16

2. Information seeking 18
2.1. Information Retrieval 18
2.2. Structuring the information-seeking process 19
2.2.1. High-level goals, tasks, and strategies 20
2.2.2. Functions, phases, and steps of searching 24
2.2.3. Low-level tasks, goals, and interface actions 27
2.3. How do users search in the Web? 29
2.3.1. General trends 30
2.3.2. User group differences 40
2.4. Summary of the chapter about Information Seeking 44

3. Information Visualization 46
3.1. The ideas behind Information Visualization 46
3.2. The reference model for visualization 47
3.3. State of the Art: Visualization Ideas, Metaphors, Techniques, Components and Systems 49
3.3.1. Metaphors 51
3.3.2. Techniques 60
3.3.2.1. Brushing and linking 60
3.3.2.2. Panning and zooming 61
3.3.2.3. Focus-plus-context 62
3.3.2.4. Magic Lenses 64
3.3.2.5. Animation 65
3.3.2.6. Overview plus detail 65
3.3.3. Components 66
3.3.3.1. Visualization of queries or query attributes 67
3.3.3.2. Visualization of document attributes 75
3.3.3.3. Visualization of interdocument similarities 88
3.3.3.4. Visualization of interdocument connections 113
3.3.4. Systems 113
3.4. State of the Art: Multiple Coordinated Views 117
3.5. Empirical evaluation of visualizations 121
3.6. Influencing Factors: 5T-Environment 127

4. INSYDER 129
4.1. The INSYDER project 129
4.1.1. Functions of the INSYDER system 129
4.1.2. Architecture and Implementation 133
4.1.3. Software development and prototypes 135
4.1.4. Formative evaluation during the project 138
4.2. The INSYDER visualizations 139

4.2.1. Ideas behind the INSYDER visualization components ... 139
4.2.2. INSYDER and the reference model for visualization ... 142
4.2.3. The INSYDER visualization components ... 147
4.3. Evaluation of the visualizations ... 157
4.3.1. Hypotheses ... 158
4.3.2. Independent Variables ... 160
4.3.2.1. User Interface ... 160
4.3.2.2. Target User Group ... 161
4.3.2.3. Type and number of data ... 163
4.3.2.4. Task ... 165
4.3.3. Static Variables ... 167
4.3.3.1. Technical Environment ... 167
4.3.3.2. Training ... 168
4.3.4. Dependent Variables ... 168
4.3.4.1. Effectiveness ... 169
4.3.4.2. Task time ... 169
4.3.4.3. Temporal efficiency ... 169
4.3.4.4. Expected added value ... 169
4.3.4.5. Satisfaction ... 170
4.3.5. Procedure ... 170
4.3.5.1. Pre-test ... 170
4.3.5.2. Entry Questionnaire ... 171
4.3.5.3. ScreenCam introduction ... 171
4.3.5.4. Warm-up Phase ... 172
4.3.5.5. 12 Tasks ... 172
4.3.5.6. Questionnaire ... 174
4.3.6. Evaluation: results ... 176
4.3.6.1. Expected added value ... 176
4.3.6.2. User Satisfaction ... 178
4.3.6.3. Hard Facts ... 193
4.3.6.4. Summary of the hard facts results ... 221

5. Summary and Outlook ... 225

6. References ... 232

7. Index of figures and tables ... 251
7.1. Figures ... 251
7.2. Tables ... 254

8. Appendix ... 256
8.1. Tasks ... 256
8.2. Additional figures from the hard facts ... 257
8.3. Additional inferential statistics ... 263
8.4. INSYDER function Mindmap ... 264

1. Introduction

1.1. Problem

"Finding the needle in the haystack" is a challenge users of the World Wide Web are often faced to. Despite the fact that there are already several hundreds of search engines available, people still often do not succeed in getting what they need. Maybe the information is not available online or not indexed by the particular search engine(s) used. Maybe the user employed the wrong search strategy, or maybe the user did not identify the needed document in the result set presented by the search engine. Researchers and the Web industry are trying to identify and solve a number of these problems, ranging from improvement of covering, indexing or ranking issues to easier accessible presentation of the search results or better user interfaces in general. The University of Konstanz has been partner in a joint project with companies from Italy and France, in which a number of these questions were addressed for a specific application domain. The project, named INSYDER (Internet Système de Recherche), was partially funded by the European Commission as Esprit project #29232. Its goal was to supply small and medium size enterprises with business information from the Web. One important part of the university's work was the development of ideas for the presentation of search results. In conducting additional user evaluations and further research on a number of questions, the University of Konstanz continued working on issues related with the project after the end of the funding. From the range of aspects examined, this thesis concentrates on the presentation of Web search results. Other important parts of the work, dealing with agent technology, ranking improvements, categorization, and relevance feedback are discussed in [Mußler, Reiterer, Mann 2000], [Mußler 2002].

To be informed is important for our everyday life. Information is one of the most important resources for private and business success. Today an enterprise must know more and more about its customers, its suppliers, its competitors, government agencies, and many other external factors. Private users are looking for information touching all aspects of life. Whereas classical information channels are still important, the Web is increasingly becoming an important information source for most subject areas. Information is readily available on the Web and the amount is being added to every hour with the multiplication of the overall number of Web pages[1]. The drawback of the overall growth of the Web is that finding relevant information becomes more and more difficult. The exponential growth aggravates even more the already often-existing situation of an information overload. With the loss of overview about obtainable and relevant data, the danger proliferates that the relevant information cannot be identified and exploited. This is not only a problem of the Web, but it culminates in this area. In 1997, Zimmer, writing in a German newspaper, noted that the chance to find certain information decreases drastically with the increase of information possibilities. His conclusion was that information overload is the key word, information rejection is the already necessary action[2].

---

[1] While the size of the Internet is not exactly measurable in March 2000 it amounted to more than one billion pages when taking the index size of common search engines as an indicator [Sullivan 2000]. [Moore, Murray 2000] reported more than two billion unique, publicly accessible pages in July 2000 and estimate four billion pages by early 2001.

[2] „*Schon jetzt, ganz am Anfang des Informations- und Kommunikationszeitalters zeigt sich, daß die Aussicht, an eine bestimmte Information zu kommen, mit der Vermehrung der Informationsmöglichkeiten drastisch abnimmt. Informationsüberlastung heißt das Stichwort, Informationsabwehr bereits das Gebot der Stunde*." [Zimmer 1997]

The question is whether people can handle so much data, or if users will be overburdened instead of supported by these huge amounts of information. With a focus on business use of the Internet, the theoretical benefits of using information from the Web for business intelligence[3] are great. In practice while a few Web resources are used as data sources, the immense resources of the Internet are largely untapped. Some problems of the past, like the availability of sufficient fast and reliable internet access for reasonable prices even for small and medium size enterprises (SMEs) or private use, are solved nowadays – at least in the majority of western industrial countries. Other points, such as Internet literacy for users of the Web or the availability of adequate tools to avoid information overload, seem to be far behind the bare technical access itself. Education of users will clearly be one of the success factors for effective usage of the Web, but is not discussed in this thesis. For an impression of its importance see [Pollock, Hockley 1997]. When concentrating on the technical aspects, there are powerful tools needed to support a continuous and systematic information-seeking approach to make use of these untapped Web resources. Besides the successful search, the effective selection of information will be one of the most important points in this process of information seeking.

Information seeking - especially in the Web - is an imprecise process. Information seekers often have only a vague understanding of how they can get the information they want. [Shneiderman, Byrd, Croft 1997] divide the information seeking into the four phases shown in Table 1 (for a closer look at structuring models for information-seeking processes see 2.2 Structuring the information-seeking process).

Phase	Description
Formulation	Selecting the sources; expressing the search
Action	Launching the search
Review of results	Presentation of the search results
Refinement	Refining the search based on the insights reviewing the results

Table 1: Four-phase framework of information seeking according to [Shneiderman, Byrd, Croft 1997] [4]

Good information-seeking systems should help to find the needed information but avoid an information overload by supporting the users in formulating their queries, selecting among available information sources, understanding search results, keeping track of the progress of their search, and reformulating their queries for a new search. If the users choices in the formulation phase lead to a small result set, information overload won't be a problem. The only problem could be that the user does not get the requested information. If the users choices lead to a large result set, adequate presentation will be crucial to support the user in not getting lost. Having in mind the list as the traditional format in which Web search results are presented, the sequence of presentation, which is normally based on the ranking, will surely be one of the most important factors to lead the user to the requested information. This is even more important, as a study of data from the search en-

[3] "*A business intelligence system ... provides a set of technologies and products for supplying users with the information they need to answer business questions, and make tactical and strategic business decisions*." [IBM 2000]

[4] Descriptions taken from [Shneiderman 1998] because of their conciseness. Originally in [Shneiderman, Byrd, Croft 1997] the authors explained: formulation: what happens before the user starts a search (sources, fields, what to search for, variants); action: starting the search; review of results: what the user sees resulting from the search; refinement: what happens after review of results and before the user goes back to formulation with the same information need.

gine Excite [Jansen, Spink, Bateman et al. 1998][5] showed that users normally do not have a look at more than the first 20 or 30 results presented[6, 7] in a session. Other studies report similar measures[8] or even lower numbers of hit pages viewed by the users[9] when looking at the query level.

People seem to do what Zimmer demands: if the result set is too large, rejection is the reaction. Regarding the information-seeking process as a multiple step selection process - where the user decides to look for the needed information in the internet, selects a search or meta search engine, chooses the keywords and search options, launches the search - in the step of reviewing the result set, the next selections are highly dependent on one dimension of the attributes of the results: the ordering of the result set, which is in most cases the relevance measure calculated by the search engine. Especially for large, unstructured result sets with intransparent ranking criteria, the distillation of relevant information will be more or less a result of a pure rejection, instead of a logic based selection in this step of the search. Due to the fact that they are all based on examinations of the search engines log files the studies about Web searching cited above say nothing about the question of which of the documents of the first three result pages are really viewed by the users. So the selection from this maximum of 10 to 30 documents could be based on a number of other dimensions showed in the result pages like title, abstract, size or age of the document, the server where it resides or others, but in any case most of the users rejected all documents in the result set ranked 31 or higher. The numbers regarding Web searching should not be over interpreted due to a number of limitations these studies have[10]. But taking it as an assumption that people do not examine all hits of large result sets and despite all efforts to improve the process of getting the result set and the ranking of items in the result set, the ranking could be a bottleneck for the selection or rejection decision of the user. This is independent from the question how many criteria or dimensions are taken into account when calculating the relevance value.

In the INSYDER project which is the basis for the work discussed here, a lot of effort has been spent to support the user on his way from his information demand to the result set and the best possible ranking of the documents in the result set (for details see [Mußler, Reiterer, Mann 2000], [Mußler 2002]). But despite all the work in this area undertaken in this project and many others, the question remains as to whether different presentations of the result set to the user, which break up the traditional sequential ordering mostly based on relevance ranking, will help the user to satisfy his information demand faster, better or in a more satisfying way.

---

[5] 86% of 18,113 users viewed not more than three result pages from Excite with 10 hits each, 77% not more than two and 58% not more than one

[6] Preliminary Version of [Jansen, Spink, Bateman et al. 1998a] cited by [Amento, Hill, Terveen et al. 1999]: "*showed that 86% of all users looked at no more than 30 pages*"

[7] [Jansen, Spink, Bateman et al. 1998] cited by [Heidorn, Cui 2000]: "*study showed that 58% of users do not look beyond the first 10 titles and 77% do not look beyond the first 20*"

[8] [Xu 1999] cited by [Spink, Xu 2000] from 1996 to 1999 over 70% of Excite users viewed not more than one result page with 10 hits each

[9] [Silverstein, Henzinger, Marais et al. 1999] in 95.7% of nearly 1 billion requests the users viewed not more than three result pages from AltaVista with 10 hits each, 92,7% not more than two and 85,2% not more than one

[10] So is the frequently cited study of Jansen et al. based on data collected from one search engine during a couple of hours on a single day, or Silverstein, Henzinger, Marais et al. mention that they could not distinguish requests by robots from requests by humans.

1.2. Solution

The goal is to find and verify a way of helping users when handling result sets of searches in the World Wide Web. The main targets are large result sets which, based on the findings of the above-mentioned studies are here defined as result sets with more than 10 to 30 hits. It is assumed that it is no problem for the user to thoroughly examine smaller sets, because at least they look at the listed presentation of the hits. Reducing the size of the result set is not the intended direction here. It may be a good way of preventing the information overload, and it is definitely worth examining, but as explained above the methods of doing this are in the scope of others works. Nevertheless the impact of the size of the result set will be discussed later in this thesis. Taking the size of the result set and the results itself as given, the only way to help the user is to ease the access to the material. There are number of possibilities of doing this. Among them are:

- The use of direct manipulation
- Shorter response times
- Structure the result set
- Extensify the use of the human visual perception system

[Shneiderman 1982] introduced the concept of "direct manipulation"[11] in 1982 at a time where command line based or ASCII-menu / form fill-in interfaces dominated the Human-Computer Interaction (HCI). In the literature a number of advantages of direct manipulation are listed since many years[12]. However, looking at a number or traditional Information Retrieval (IR) interfaces still in use or thinking about the "direct manipulation" attitude of HTML-based interfaces of common search engines there seem still to be room for improvements in this application domain.

Handling the result sets of common search engines is usually a click-and-wait process because the Web itself is used as a medium between user-interface (usually a browser) and the server, where the result set is handled. The still common low transmission rates of the Web are therefore a bottleneck that leads to "long" user-interface response times. The fact that systems with shorter response times lead in general to higher user satisfaction and shorter task performance times (despite sometimes higher error rates during the task solution process) has been shown years ago [Shneiderman 1987]. It would be interesting to see if the viewed portion of large result sets will increase from the reported 10 to 30 hits to higher numbers, if response times at the user interface will be decreased. In any case the response time will be an important factor for the success of a user interface for handling Web search result sets.

Another point is to add structure to the presentation of the result set. Since years studies have shown that structuring the data presented on the screen in an appropriate way can significantly improve the accessibility of information. The possibilities for doing this range from the spatial arrangement format of text[13] to ideas used to group or cluster the elements of result sets. An impressive example how a modified usage of the cluster hypothesis [Van Rijsbergen 1979] can be used to successfully guide user when examine result sets of document searches can be found in

[11] Direct manipulation is characterized by: continuous representation of the object of interest, physical actions or button presses instead of complex syntax, and rapid incremental reversible operations whose impact on the object of interest is immediately visible. [Shneiderman 1998]

[12] [Jacob 1989], [Shneiderman 1987], [Triebe, Wittstock, Schiele 1987], [Ziegler, Fähnrich 1988]

[Hearst, Pedersen 1996a]. Hearst and Pedersen performed the clustering step dynamically on the retrieved result set instead of just doing a static clustering of the entire collection. With theoretical analysis and a small test with four users they showed that a combination of document clustering and traditional ranking could outperform pure ranked lists of documents. Main factor that made this possible was the observation that relevant documents tend to fall together in one or two clusters and therefore allow users to concentrate on subsets of the result set, instead of the whole result set.

A further possibility to support users to get insights into great amounts of abstract data is to extensify the use of the human visual perception system. One way of doing this is the use of adequate visualizations of the result set. Using visualization to support information seeking will employ the enormous capacity for human visual information processing [Ahlberg, Shneiderman 1994] and allow the user to reach his goals aided by visual information seeking. "*By presenting information visually and allowing dynamic user control through direct manipulation principles, it is possible to traverse large information spaces and facilitate comprehension with reduced anxiety.*" [Ahlberg, Shneiderman 1994]. Doing this, hopefully the information rejection when handling large result sets changes to a selection supported by visualizations. [Ahlberg, Shneiderman 1994] list a number of principles that are important for visual information-seeking systems:

- Visual representation of the world of action including both objects and actions
- Rapid, incremental and reversible actions
- Selection by pointing
- Immediate and continuous display of results
- Support of browsing
- Progressive refinement of search parameters
- Continuous reformulation of goals
- Visual scanning to identify results.

The usage of visualization to support the exploration of large volumes of abstract data with computers has been known for a number of years as Information Visualization (IV). IV can be defined as "*The use of computer-supported, interactive, visual representations of abstract data to amplify cognition.*" [Card, Mackinlay, Shneiderman 1999]. Working with the visualization of result sets of Web searches for the purposes listed above is a typical use case for Information Visualization. [Card, Mackinlay, Shneiderman 1999] and a lot of other authors show ideas or results how the usage of IV can help to explore result sets. But which visualization will be the best for the usage scenario examined here?

As has been shown, there are a number of theoretical possibilities regarding how to help users when handling result sets of searches in the World Wide Web. And this is also the case when neglecting the most obvious ones like reducing the size of the result set or improving the ranking algorithms. Here, the use of Information Visualization is very interesting. The human perceptual system is highly adjusted to the effective processing of visual coded information [Tufte 1983], [Larkin, Simon 1987], [Card, Mackinlay, Shneiderman 1999]. The amount of literature regarding

---

[13] [Guastello, Traut, Korienek 1989] list a number of relevant studies.

ideas of how to use visualization for information handling purposes has exploded over the last few years. On the other hand the number of experimental verifications of how helpful these ideas really are is relatively low. Additionally the usage of information visualizations inherently carries or requires some of the other possibilities like direct manipulation or short response times. This is also evident when looking at the above listed principles for visual information-seeking by Ahlberg and Shneiderman, which by the way were derived by taking the principles of direct manipulation as a starting point. The above mentioned INSYDER project offered an ideal test bed to implement some ideas out of the huge field of IV ideas, and really test their effects when used to support users in handling result sets of Web searches. The theoretical background, the rationales behind the user interface design choices and implementation, the design of the performed user study and its results will be described in the remainder of this thesis.

1.3. Structure of the Thesis

The remainder of this thesis will start in Chapter 2 with a brief discussion of the information-seeking process. The relation to classical information retrieval will be exposed. Different models used to structure the information-seeking process in phases or tasks will be shown. Focusing on the application domain information seeking in the Web, the chapter will close with some notes regarding what is known about how users search in the Web.

Chapter 3 is dedicated to Information Visualization. An introduction of a reference model for information visualization is followed by an overview about the state-of-the-art of Information Visualization structured in metaphors, techniques, components, and systems. The chapter is focused on visualizations of abstract data. The special case of multiple coordinated views will be addressed in a separate sub-chapter. The main chapter about IV will close with a discussion of empirical evaluations of visualization ideas and a compilation of crucial factors for the usefulness of visualizations.

Chapter 4 begins with a description of the INSYDER project and software as a framework for the evaluations which are the basis for the results presented in this thesis. The implemented visualizations are presented and discussed in detail. After a description of the ideas behind the evaluation, the hypothesis, the variables, and the procedure, the findings will be thoroughly presented and discussed.

The thesis will conclude with a summary and outlook in Chapter 0. A reference list in Chapter 0, an index of the figures and tables in Chapter 0, and some additional information in the appendix follow from this.

Figure 1 shows the structure of the thesis with its main parts.

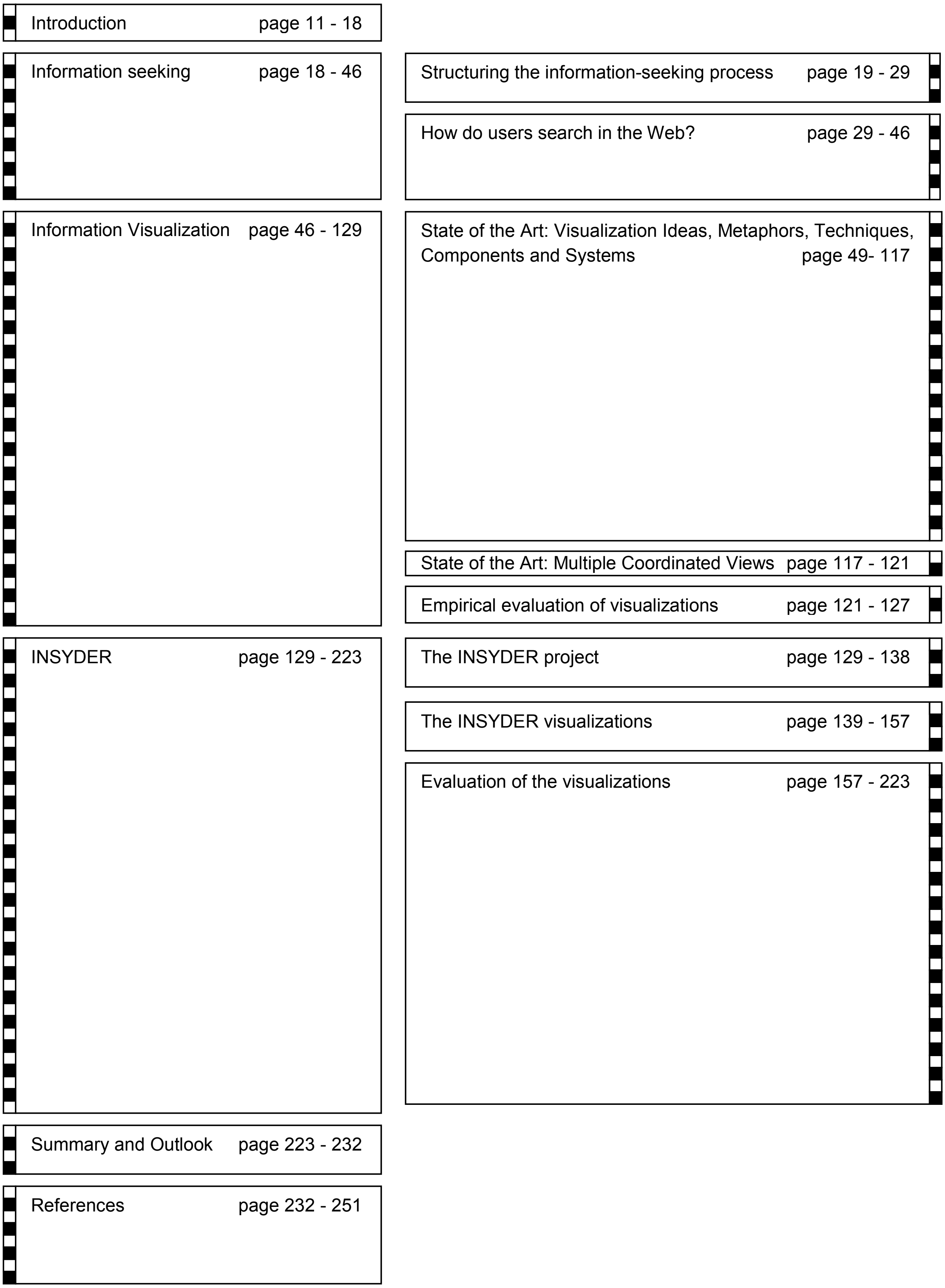


Figure 1: Structure of the thesis (main parts)

2. Information seeking

2.1. Information Retrieval

The search of information in the World Wide Web today has a number of elements in common with classical Information Retrieval (IR). Basically in both cases the user has an information need that is being satisfied by using a (online) search system. In Chapter 2.2 “Structuring the information-seeking process”, we will see what the structural differences are when we try to model the information-seeking process for classical Information Retrieval or Internet searching. In harmony with the common elements of the search process, Internet search engines use a number of principles and methods developed in the long history of IR. Anyhow there are also a number of important differences that have to been taken into account when working in this field. “*Internet searching is very different then IR searching as traditionally practiced and researched. Internet IR is a different IR.*” [Jansen, Spink, Bateman et al. 1998]. Especially when looking for research results from Information Retrieval to draw conclusions for Internet searching, there are a number of points which have to be regarded[14]:

- Classical IR in the past often dealt with bibliographic citations. Internet searching is mainly full text searching[15].
- Many of the classical IR studies in the past were performed with systems using pure Boolean logic. Internet search engines mainly use statistical ranking methods[16].
- A near miss in classical IR was often a miss, due to absent hyperlink possibilities in the document collection. Searching the Internet a near miss can sometimes lead to a needed document by following a hyperlink.
- Precision may play another role in Web retrieval, than in classical IR [Eastman 1999][17].
- Many of the classical IR studies focus on professional intermediaries like librarians. Internet searching is mainly end user searching.
- IR systems used in earlier times in classical IR studies often had command line based interfaces. Internet searching nowadays means at least form fill-in or hyperlink-environments, sometimes even direct manipulation interfaces. A number of studies in the classical IR-environments during the last few years also used these types of interfaces. Here it is important to assess under which conditions reported results and conclusions arose.

---

[14] Most of the points taken from [Hearst 1999]. Complemented with my own considerations. The goal of Hearst’s listing is a comparison between earlier IR interface studies and “modern information access”. Nevertheless many of the points are true for a comparison of a large part of the IR-research described in the literature and “internet searching”.

[15] “Full text” does not mean the full text of the Internet, but the full text of the documents in the fraction of the Internet covered by the used search engine(s).

[16] Many of them have additional Boolean options, but the statistical ranking is nearly always present. Sometimes these statistical ranking methods are not only concentrated on the query-document-relation itself, but also process information like the number of references from other pages or sites.

[17] [Eastman 1999] made her students perform exercises in Web search, to demonstrate a well-known effect from classical IR: more precise and narrower searches lead to fewer hits and better results. This was not always reliable for the searches the students performed using popular Web search engines. A reexamination of a number of searches confirmed this observation.

- Classical IR studies in the past were often performed with systems where moving from one collection to another was not easy, because of the time and effort needed to switch or the additional knowledge which was necessary. The Internet itself can be regarded as a huge collection of numerous collections of documents that can be searched at one time. Nearly every server can be seen as a single collection. If we regard the fraction of the internet covered by a certain search engine as a collection, switching collections in the Internet is often just one click away, because search engines sometimes offer links to other engines, or meta-search-engines offer easy selection possibilities for underlying engines to use.
- In classical IR the user will very seldom have the possibility to perform a search on the same or nearly same document collection with completely different search engines. It is relatively easy when searching the Internet to use specialized search engines or specialized directories for special purposes.

With regard to user behavior, there are also a number of differences between classical IR and searching in the Internet. Examples are the number of search terms used or the number of queries per session.

- Whereas in traditional IR systems the average number of terms used in a query range from about 7 to 15 terms [Jansen, Spink, Bateman et al. 1998], for Web search the average is between 1.7 and 3.6[18].
- When searching the Web, in more than 2/3 of cases[19] users only have one query in a session, which is a significant contrast to searches using classical IR systems, where modification of queries often occurs. [Jansen, Spink, Bateman et al. 1998].

Despite these differences it is interesting to see how the "information-seeking process" can be structured with classical IR in mind or a more general view of information-seeking, and what the modifications or specialties for internet searching are. This will be discussed in the following chapter.

2.2. Structuring the information-seeking process

In recent years the number of documents published on the Web has increased dramatically. This has brought research into information-seeking systems within the focus of people dealing with the Web. When trying to create a software system it is essential to have a model concerning the process(es) which should be supported by the system. As with any other software system this is also true of a system supporting an information-seeking process like searching the Web. So one of the first steps when dealing with information-seeking systems is to get an idea of how to describe the information-seeking process best. As listed above, searching the Web differs in a number of ways from other information-seeking processes like, for example, classical Information Retrieval. Nevertheless, when trying to structure the Web search process, a lot of things can be learned from studying classical IR process models. Most of the models which can be found in the literature try to structure a search process in terms of *goals* and/or *strategies* and/or *phases* and/or *tasks* and/or *steps*. The usage of these terms differs from author to author. What is labeled a "phase" in one

[18] See Chapter 2.3 for details

[19] 67% according to [Jansen, Spink, Bateman et al. 1998]; 77.6% according to [Silverstein, Henzinger, Marais et al. 1999]

model can be a “step” in another, and what one author classifies as a “goal” is a “task” for another. When talking about specific terms like “tasks”, the granularity of a certain task can range from “information retrieval” in general to “compare within entities” as a specific low-level task. The same is true for goals, where the level can range from “monitoring a well known topic over time” to “accurate value lookup”. The next three sub-chapters try to structure the field and distill out a framework that can be used as a guideline for system design and evaluation.

2.2.1. High-level goals, tasks, and strategies

The common starting point of nearly all interaction-process- or phase-models of the information-seeking process is that there is always a user information need at the beginning. This starting situation is often characterized in the IR literature as an anomalous state of knowledge (ASK) [Belkin 1980] / [Belkin, Oddy, Brooks 1982] / [Belkin, Oddy, Brooks 1982a]. Derived from the information need, the user will have one or more goals explicitly formulated, or implicitly in mind behind his actions. [Hearst 1999] lists “*finding a plumber*”, “*keeping informed about a business competitor*”, “*writing a publishable scholarly article*”, and “*investigating an allegation of fraud*” as examples for goals. Hearst comes from her goals to information access tasks that are used to achieve these goals. These tasks can span from asking specific questions to exhaustively researching a topic. A task example she cites from [O’Day, Jeffries 1993] is “*monitoring a well-known topic over time*”. This task could, for example, be developed from the goal to be kept informed about a business competitor. From the tasks Hearst comes to a model of interaction, where the information need is the starting point that is to be followed by different steps like “*select a system and collection to search on*” or “*formulate a query*”.

Whereas Hearst’s tasks are dependent on the user’s goals, [Goldstein, Roth 1994] developed a model for data exploration where the goals are dependent on the user’s task. However the authors write: *“... we classified the types of interactive data exploration tasks (goals) that users will perform ...*”. They list for example under data manipulation tasks goals such as “controlling scope” or “choosing level of detail”. Goals at the same level of detail can also be found in other contexts too, like for example “*accurate value lookup*” or “*comparison of values*” in [Roth, Mattis 1990]. This type of goals will be classified here as low-level tasks, and will be discussed later in Chapter 2.2.3 Low-level tasks, goals, and interface actions.

On the same granularity of information access tasks listed by Hearst, [Shneiderman 1998] differentiates four types of “task actions” listed in Table 2.

Task actions
Specific fact-finding (known-item search)
Extended fact-finding
Open-ended browsing
Exploration of availability.

Table 2: Task actions according to [Shneiderman 1998]

The two fact-finding tasks both produce clear and replicable outcomes. The main difference between these two types is that in the first case there is a clear stop criterion, when the user finds a document to answer the question. In the second case there is no such clear abort criterion to stop the examination of a result set or the overall search, and therefore the investigation process of a result set or the complete information-seeking process will be much broader in scope and possibly of longer duration. Even more open and unstructured are the remaining two task actions open-ended browsing and exploration of availability. Trying to fit Hearst's goal examples in this classification, "*finding a plumber*" can lead to a specific fact-finding task. Shneiderman's corresponding example is "*Find the telephone number of Bill Clinton*". Hearst's "*keeping informed about a business competitor*" could lead to an extended fact-finding task or open-ended browsing. Here the corresponding examples from Shneiderman are "*What genres of music is Sony publishing?*" for extended fact-finding and "*Is there new work on voice recognition being reported from Japan?*" for open ended browsing. Taking the remaining example goals from Hearst "*writing a publishable scholarly article*" and "*investigating an allegation of fraud*" the first task action will probably be an exploration of availability, eventually later followed by more specific task actions. A comparison of the information access tasks by [Hearst 1999] and the task actions by [Shneiderman 1998] is shown in Figure 2.

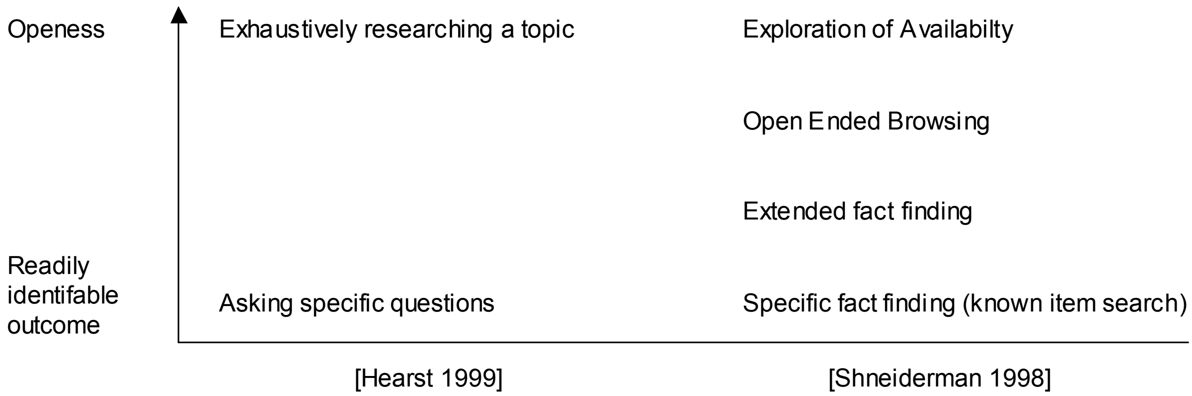


Figure 2: High-level tasks by [Hearst 1999] and [Shneiderman 1998]

[Shneiderman 1998] points out that the task actions are broken down into browsing or searching. In a next step browsing and searching are represented by interface actions like scrolling or zooming. But before we reach this level of detail two other points should be discussed in more depth: information-seeking strategies and phases or steps of searching.

Using again the "finding a plumber" example, there are different possibilities to fulfill the information need. [Baeza-Yates, Ribeiro-Neto 1999] emphasize, when using a retrieval system for ASK-situations, the distinction between two different types of strategies: information or data retrieval on the one hand and browsing on the other. In fact, they categorize retrieval and browsing as two different types of *tasks*. The general distinction between searching (sometimes also named direct querying or retrieval by specification) and browsing (sometime also named scanning or retrieval by recognition) is very common in the literature. As shown above, Shneiderman makes the same distinction, however not directly using the term "task" on this level. Because the term *task* is used in such an inflationary way by many authors, it seems to be more appropriate to classify these different types of behavior as *strategies* like for example done by [Henninger, Belkin 1996]. Having a closer look at information-seeking strategies [Belkin, Marchetti, Cool 1993] and [Belkin, Cool, Stein et al. 1995] try to structure the field by defining a multi-dimensional space of information-seeking strategies. For this purpose they use four dimensions: method of interaction (scanning ⇔ searching), mode of retrieval (recognition ⇔ specification), goal of interaction (learning ⇔ selecting), and resource considered (information ⇔ meta information). With these dimensions they create a matrix that shows the possible combinations in the form of sixteen different Information-Seeking Strategies (ISS). Table 3 shows a selection of the most interesting ISSs in the context of this thesis.

ISS	Method of Interaction	Mode of Retrieval	Goal of Interaction	Resource Considered
ISS5	Scan	Recognize	Select	Information
ISS7	Scan	Specify	Select	Information
ISS13	Search	Recognize	Select	Information
ISS15	Search	Specify	Select	Information

Table 3: Examples of Information-Seeking Strategies ISS according to [Belkin, Marchetti, Cool 1993] and [Belkin, Cool, Stein et al. 1995]

The goal of interaction as a dimension of the matrix created by Belkin et al. focuses on the retrieval system used. The two modes are “learn” and “select”. For the resource considered, the distinction between “information” and “meta information” is a classical IR category. The subtle differentiation between method of interaction and mode of retrieval is particularly interesting. The authors point out that scanning is typically associated with retrieval by recognition, and searching with retrieval by specifications, but they present examples where this typical connection is broken up. Another important point Belkin et al. emphasize is possible changes of the ISS during an information-seeking episode. Depending on previous knowledge, the user will start an information-seeking process with a certain strategy. Getting the first results may cause him to change this strategy. The next set of results may cause another change and so on. The idea that information seeking is not always a straightforward process with one best strategy can also be found in other models. One of the most famous ones, which also emphasizes the diversity of strategies, is the berrypicking model of [Bates 1989]. She also points out that it is not only the strategy that may change, but also the information need itself. Another important message from Bates is that the information need may not be satisfied by a single, final retrieved set of documents. All or part of the information chunks found on the way may also contribute to satisfying the information need(s). Bates lists six widely used information-seeking strategies: footnote chasing or backward chaining, citation searching or forward chaining, journal run, area scanning, subject search in bibliographies and abstracting and indexing services, and author searching. These strategies as parts of the berrypicking model were observed when people used manual sources. At the end of the 1980s, Bates had great expectations that hypertext approaches would be ideal for berrypicking. What was true for hypertext will also be true for the Web as the biggest hypertext so far formed.

The findings of Bates are supported by a number of authors like [O’Day, Jeffries 1993] or [Hearst 1999]. The former studied the use of information search results by fifteen regular clients of professional intermediaries. As shown above, Web searching is mainly end-user-searching. Nevertheless, the patterns they found for mediated searches may also occur in Internet searching. They classified three basic search modes: monitoring, planned, and exploratory. Or in more detail: monitoring a well-known topic or set of variables over time, following an information-gathering plan suggested by a typical approach to the task at hand, and exploring a topic in an undirected fashion. In addition they identified patterns of interconnected searches. They established that the accumulation of search results had value for the end-users - not only the final result set – and this even for mediated searches. It may be even more the case for end-user searching.

Focusing back on the internet [Baeza-Yates, Ribeiro-Neto 1999] expand their above listed two different *tasks* retrieval and browsing to three basic forms of searching for information in the Web: the use of search engines, that index a portion of the Web documents as a full-text database, the use of Web directories, which classify selected Web documents by subject, and the exploitation of the hyperlink structure of the Web for search purposes. In fact we have three different strategies here where the use of search engines corresponds strongly to the classical search strategy, and the two other ones are both varied forms of classical browsing.

Also appealing is an approach from [Choo, Detlor, Turnbull 1998] / [Choo, Detlor, Turnbull 1999] combining Aguilars's four modes of organizational scanning [Aguilar 1967] with the six categories of information seeking behavior defined by [Ellis 1989], to a new model of modes and moves for information seeking in the Web. The modes are: undirected viewing, conditioned viewing, informal search, and formal search. For every mode they attach a number of moves (information seeking categories) shown as categories in Table 4. The authors verified the model by analyzing 61 Web information seeking episodes of 34 Web users from different professions. The strength of the model is its clear and simple structure; however, its main weakness is that not all of the real-world possibilities can be adequately placed in a cell of the model. Chaining, for example, is only attached to undirected viewing, but can surely also sometimes be found in formal search mode (even when not found in this combination in the 61 episodes). What is in any case interesting is their comparison of literature search moves from [Ellis 1989] with their Web moves equivalents shown in Table 4.

Category	Literature Search Moves	Anticipated Web Moves
Starting	Identifying sources of interest	Identifying websites/pages containing or pointing to information of interest
Chaining	Following up references found in given material	Following links on starting pages to other content-related sites
Browsing	Scanning tables of contents or headings	Scanning top-level pages: lists, headings, site maps
Differentiating	Assessing or restricting information according to their usefulness	Selecting useful pages and sites by bookmarking, printing, copying and pasting, etc. Choosing differentiated, pre-selected site
Monitoring	Receiving regular reports or summaries from selected sources	Receiving site updates using e.g. push, agents, or profiles Revisiting 'favorite' sites
Extracting	Systematically working a source to identify material of interest	Systematically searches a local site to extract information of interest at that site

Table 4: Comparison of literature search and Web moves according to [Choo, Detlor, Turnbull 1999] Fig. 2.

Other models in the area of Web information seeking try to cope with special artifacts of the process. An example for this is the work of [Navarro-Prieto, Scaife, Rogers 1999]. After a study performed with 10 Computer Science and 13 Psychology students, they defined different Web search models for users with high and low experience to make predictions about the participants' searches. The model for experienced searches is much more complex than the one for novices.

As we have seen there are a number of different high-level models available which look at how to structure the information-seeking process in the form of goals, tasks or strategies. More detailed overviews and discussions can be found in [Hearst 1999] or [Morse 1999]. For the context used here the following four most important conclusions can be drawn out of the different approaches:

- Classical search is just one of the possible ways to fulfill an information need
- Goals and strategies are not static, but may change during an information-seeking episode
- Not only the final result set is important, a number of factors contributing to fulfilling the information need may also come along the way
- Strategies may depend on user experience

Shneiderman's task action model [Shneiderman 1998] shown in Table 2 on page 20 will be focused on as a concrete task model in the remainder of this thesis. The content area of this thesis is the visualization of search results; therefore the next chapter, discussing lower levels of abstraction, will concentrate mainly on the aspects of *searching* as a strategy, despite the fact that there are a number of other possibilities which can be used in fulfilling an information need.

2.2.2. Functions, phases, and steps of searching

When concentrating on searching, the information-seeking process can be broken down into a number of finer granulated functions, phases or steps. A famous model of doing this, especially targeted on end-user information seeking, is proposed by [Marchionini 1992]. It consists of the following five functions: Define the problem, Select the source, Articulate the problem, Examine the results, and Extract information. Like many other authors[20] Marchionini points out that the overall process is iterative. To accentuate this, he represents the functions in the corresponding figure in a nonlinear way as shown in Figure 3.

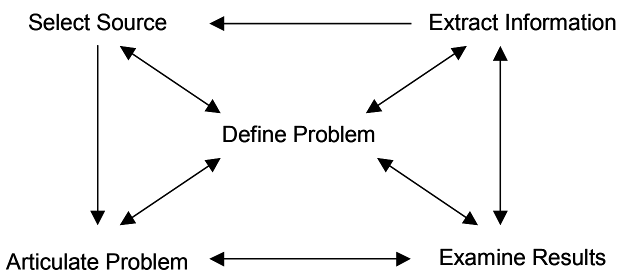


Figure 3: Information seeking functions according to [Marchionini 1992] p. 157 FIG. 1.

The representation is without doubt nonlinear, but it lacks a little bit in terms of showing what Marchionini himself explains as: *"recognizing and defining an information problem initiates information seeking*" [Marchionini 1992]. This initiation as a starting point is better depicted by a revision of this model undertaken in [Marchionini 1997], and shown in Figure 4. The fact that the process starts at a certain point with an information need is also shown in a figure used by [Hearst 1999] to show a standard process as a sequence of steps. It is reproduced here in Figure 5. The revised model by [Marchionini 1997] contains the following steps: Recognize and accept an information problem => Define and understand the problem => Choose a search system => Formulate a query => Execute search => Examine results => Extract information => Reflect / Iterate / Stop. Comparing the figures from Marchionini and Hearst the main functions from Marchionini can be found as steps in Hearst's diagram, except "select source". Interestingly enough, in her textual description the step is listed: *"(1) Start with information need. (2) Select a system and collections to search on. (3) Formulate a query. (4) Send the query to the system. (5) Receive the results in the form of information items. (6) Scan, evaluate, and interpret the results. (7) Either stop, or, (8) Reformulate the query and go to step 4.*" [Hearst 1999]. After introducing the "standard" process Hearst too emphasizes the non-linearity of the overall process, and furthermore, points out that there are a number of points like the role of scanning and navigation not represented in the model. Supporting Bates, she also de-emphasizes the role of the final result set and states that accumulated learning and acquisition of information occurring during the search process is the main value of the search.

[20] E.g. [Shneiderman 1998] or [Hearst 1999]

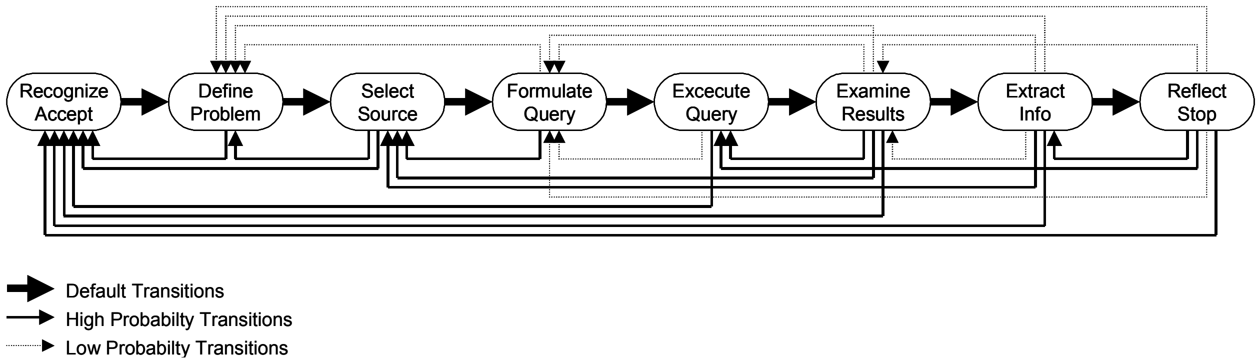


Figure 4: Information-seeking process according to [Marchionini 1997] p. 50 Figure 3.3

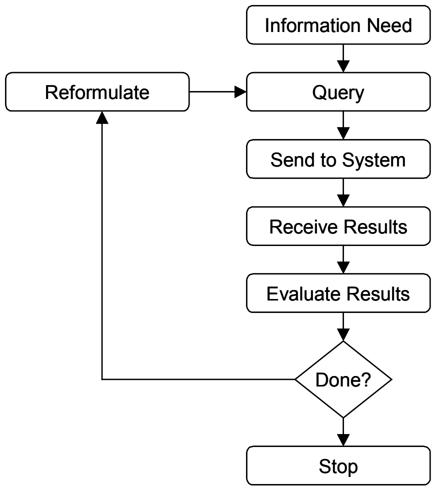


Figure 5: Simplified diagram of the standard model of the information access processes according to [Hearst 1999] p. 263, Figure 10.2

[Shneiderman, Byrd, Croft 1997] divide a search into the four phases listed in Chapter 1.1: Formulation => Action => Review of results => Refinement. The classification is similar to the functions of [Marchionini 1992] or the standard process described by [Hearst 1999]. The same is true of another model by [Veerasamy, Heikes 1997], describing a typical user interaction with current IR systems. The differentiation of a first and a second stage in the display of results is particularly interesting. Table 5 shows a comparison of the four models. Marchionini's 1992 figure was linearized for the comparison. The model of [Shneiderman, Byrd, Croft 1997] is taken in the version by [Shneiderman 1998].

[Marchionini 1992]	[Marchionini 1997]	[Hearst 1999]	[Veerasamy, Heikes 1997]	[Shneiderman 1998]
	Recognize and accept an information problem	Start with information need	User in anomalous state of knowledge expresses information need as query	
Define the problem	Define and understand the problem			
Select the source	Choose a search system	Select a system and collections to search on		Formulation: selecting the sources; expressing the search
Articulate the problem	Formulate a query	Formulate a query		
	Execute search	Send the query to the system		Action: launching the search

[Marchionini 1992]	[Marchionini 1997]	[Hearst 1999]	[Veerasamy, Heikes 1997]	[Shneiderman 1998]
		Receive the results in the form of information items	System matches query with stored documents and retrieves a set of documents	
Examine the results	Examine results	Scan, evaluate, and interpret the results	First stage of display: surrogates are displayed	Review of results: presentation of the search results. Read, view, sort, …
			User inspects surrogates and requests more information (second stage of display)	
Extract information	Extract information			Refinement: refining the search based on the insights reviewing the results. Save, send or take results as input for other programs
	Reflect / Iterate / Stop	Either stop, or Reformulate the query	After going through a sufficient number of documents quit session or reformulation	

Table 5: Comparison of the search models by [Marchionini 1992], [Marchionini 1997], [Hearst 1999], [Veerasamy, Heikes 1997], and [Shneiderman 1998]

The authors cited here, and many others too, emphasize the importance of the steps from the popup of an information need to the articulation of the problem. In the context of this thesis, which has its main focus on the visualization of search results, it may be permissible to summarize this process as “formulation” as done by [Shneiderman, Byrd, Croft 1997]. The most important point here is that the user has to select sources in terms of where to search and to transform his information need into a query. The query is a representation of the information need, and thus must be expressed in a language understood by the system [Belkin, Croft 1992]. The next logical step is to launch the search. In the remainder of this thesis, the short term “action” will be used for this. As with “formulation”, what happens during this action-phase is a whole world in itself. In the user-centered models, this phase plays only a small role. In practice, this step with all its crawling, ranking, or processing is also a important factor influencing what the user will be presented with in the next step, the review of results. This last mentioned step is the really interesting one when talking about the visualization of search results. Going back to the four-phase framework, the last step of the information-seeking process is the refinement. It must be emphasized that refinement is not the last step in searching the Web, which is usually an iterative process. After examining the results, the user will be able to refine his initial formulation. From the point of visualization, the refinement step has elements from the formulation and result phase, and some new elements that do not need to be discussed here[21]. In the context of this thesis “refinement” stands as a summary of the steps after the review of results, and as a reminder of the iterative nature of the overall process. Because the visualization of search results is clearly a part of the review of results, in the rest of this thesis “formulation” will be used to describe a placeholder for the users input part of the process[22], “action” for the systems part, and “refinement” for the next steps.

[21] A point of particular interest is for example visualization support for relevance feedback. An interesting discussion about the pros, cons and potential problems of relevance feedback in general can be found in [Hearst 1999]

[22] Despite the fact that the system should support the user in the formulation.

2.2.3. Low-level tasks, goals, and interface actions

As mentioned before, the terms "task" and "goal" can be used in a number of quite different granularities. Concentrating now on the review of result sets with the one or other step left or right to show the scope of possibilities, diverse approaches in structuring low-level tasks a user has to perform in this step of the overall information-seeking process can be found. Some of these low-level tasks are so generic that they can also occur in other steps of the process. In addition we are now approaching the visual part of visual information seeking. So this is a good moment to introduce the visual-information-seeking mantra by [Shneiderman 1998]: "*Overview first, zoom and filter, then details on demand.*" This guideline is a summary of lessons learned from a number of projects, and introduces some of the interface actions Shneiderman uses in his Data Type by Task Taxonomy (TTT). The idea behind the TTT is to identify visualization data types and the tasks that need to be supported. The tasks he lists there are shown in Table 6.

Task	Description
Overview	Gain an overview of the entire collection.
Zoom	Zoom in on items of interest.
Filter	Filter out uninteresting items.
Details-on-demand	Select an item or group and get details when needed.
Relate	View relationships among items.
History	Keep a history of actions to support undo, replay, and progressive refinement.
Extract	Allow extraction of subcollections and of the query parameters

Table 6: Tasks of the TTT data type by task taxonomy according to [Shneiderman 1998] p. 524 Box 15.2

Much briefer is a classification by [Rohrer, Swing 1997] who view visualization as an integral component in data analysis and identify two main tasks to be supported by this as shown in Table 7: analytic discovery and data reduction.

Task	Description
Analytic discovery	Highlights a key knowledge nugget from a large corpus of data.
Data reduction	Cull away uninteresting portions and then apply more conventional tools to the remaining data.

Table 7: Main tasks supported by visualizations in data analysis according to [Rohrer, Swing 1997]

Another, completely different task or low-level goal classification is used by [Roth, Mattis 1990] to describe user information seeking goals. The elements are shown in Table 8. The authors use this classification to show the influence of differences in goals on the effectiveness of graphical techniques or their combinations. Actually their classification is much more focused on fact-retrieval than Shneiderman's, which has a broader scope. Nevertheless, it is interesting to see which low-level goals or tasks they identify.

Goal
Accurate value lookup
Comparison of values within, but not among different relations
Pair wise or n-wise comparison of relations for the same data set
Distributions of values for a relation
Functional correlations among attributes
Indexing-needs for one or both data sets within a relation

Table 8: User information seeking goals according to [Roth, Mattis 1990]

In the context of the visualization of Web search results, the tasks listed by [Roth, Mattis 1990] are interesting when dealing with the metadata of the documents or facts drawn from the documents. But even when we look at broader scope information searches, like those undertaken in the previously cited study of [O'Day, Jeffries 1993], we can identify some comparable tasks or low-level goals. [O'Day, Jeffries 1993] refer to these as analysis techniques, and state that about 80% of 80 analysis examples they researched fell into one of the six categories listed in Table 10.

Analysis technique
Looking for trends or correlations
Making comparisons of the different pieces of the data set
Experimenting with different aggregates and/or scaling
Identifying a critical subset of relevant or unique items
Making assessments
Interpreting data to find meaning in terms of domain or problem concepts

Table 9: Analysis techniques according to [O'Day, Jeffries 1993]

Focusing on the data manipulation aspects of interactive data exploration, [Goldstein, Roth 1994] categorized users' goals by examining users tasks as shown in Table 10. Additionally they break down the goals into operations as shown in the third column.

Task	Goal	Operation
Data manipulation	Controlling scope	Filter data using attribute(s)
		Select multiple disjunctive subsets
	Selecting focus of attention	Select attribute(s) for viewing operations
		Select attribute(s) for level of detail operations
		Select attribute(s) from existing attributes
	Choosing level of detail	Predefined aggregation & decomposition
		Flexible aggregation & decomposition
Data analysis	No detailed goals or operations defined by the authors. An example listed is obtaining statistics on portions of the data.	
Data visualization	No detailed goals or operations defined by the authors. Examples listed are requirements and specifications for viewing the data through appropriate visualizations.	

Table 10: User tasks and goals in interactive data exploration according to [Goldstein, Roth 1994]

Another model, especially for visual environments comes from [Wehrend, Lewis 1990]. They describe domain-independent operations classes that users might perform. The operation classes they list are shown in Table 11.

Task
Identify
Locate
Distinguish
Categorize
Cluster
Distribution
Rank
Compare within relations
Compare between relations
Associate
Correlate

Table 11: Operation classes in a visual environment according to [Wehrend, Lewis 1990], Table 2

To complete the overview about the scope within which tasks can be structured, a last model specially dedicated to document spaces will be introduced: Navigation tasks taken as an assumption by [Spring, Morse, Heo 1996] to develop the CASCADE system are shown in Table 12.

Navigation tasks
Finding groups of objects of interest
Finding specific objects of interest
Following interesting paths
Tentative exploration of objects of given attributes

Table 12: Navigation tasks in a document space taken as an assumption by [Spring, Morse, Heo 1996] to develop the CASCADE system.

What is the insight from this chapter? On higher levels of abstraction of the information-seeking process it was, with a certain amount of effort, still possible to draw a common set of conclusions from the different models and findings or to compare them. On the plane of low-level tasks, goals and interface actions this is much more difficult. In the rest of this thesis Shneiderman's task classification [Shneiderman 1998] will be used where necessary. On the one hand its level of granularity is high enough to grasp a broad scope of problems. On the other hand it is seamlessly integrated in the higher levels of abstraction by Shneiderman et al. already chosen above, like the four-phase framework or the task actions model. In the remainder of this work, the discussion will concentrate mainly on the area of high-level goals, tasks, and strategies or functions, phases, and steps of searching. Low-level tasks, goals, and interface actions will play only a subordinate role, except for a later introduction of visualization techniques. Nevertheless, this chapter has been included to provide a summary of the overall view.

2.3. How do users search in the Web?

After the introduction of theoretical models for the information-seeking process it will be interesting to know what people are really doing when they search the Web. A number of findings have already been mentioned in the introduction in establishing the framework for this thesis, and in Chapter 2.1 "Information Retrieval" to show the differences between classical Information Retrieval and Web searching. In the following chapter we will have a closer look at a number of studies about people searching the Web.

Research on Web searching is in its infancy [Jansen, Pooch 2000]. When investigating search behavior of World Wide Web users, a number of interesting questions arise, including who the users of the Web are, what they are looking for, where they are looking, how they are looking? Where do they search?, and How do they search? As well as being of scientific interest, some of the answers to these questions are of high economic value, due to the high economic impact of the World Wide Web. So it is, for example, very interesting for the advertising industry to know what the characteristics of the user population of a certain search engine are or which keywords people use the most. Due to this economic value, not all of the material collected is affordable for someone who is writing a thesis. This is made more frustrating by the fact that things are changing fast. This change does not only involve the size of the Web or the number of users. With a growth in user population, a change in its assembly follows. In summary, we are looking here at results from a research discipline which is at an incipient stage, where not all of the already collected material is available, and where we are trying to study a fast moving and evolving target. Therefore the subsequent discussed results can only be a spotlight on the overall field of Web searching. Nevertheless, some of the findings are quite interesting for the visualization of search results. The visualizations evaluated for this thesis were used in a dedicated system, customized for a dedicated user population and their information needs. So a number of the questions listed above, such as "Who are the users of the Web?" are of minor relevance in terms of this thesis. The following compilation is therefore focused on the question "How do people search the Web?". For a broader overview of studies looking at how people navigate through the Internet see [Hölscher 2000].

2.3.1. General trends

[Jansen, Pooch 2000] conducted an extensive literature review and identified only three large scope Web-searching studies, all done by analyzing log files from one of the large search engines:

- The Excite study, done by [Jansen, Spink, Bateman et al. 1998][23] using a part[24] of a one day data set with about 51,000[25] transaction records from Excite, collected in 1997 (March). The study is part of an ongoing series of studies of Excite data, which has other (up to now) less documented parts using data sets with 1.2 million and 30 billion queries [Spink, Xu 2000].
- The 1998 Fireball study, done by [Hölscher 1998][26] using a one month data set with about 16 million queries from the German search engine Fireball[27], collected in 1998 (July) [Hölscher 1998a]
- The AltaVista study, done by [Silverstein, Henzinger, Marais et al. 1999][28] using a 1.5 month data set with nearly 1 Billion requests from AltaVista, collected in 1998 (Aug/Sep)

Besides the Fireball study by [Hölscher 1998] mentioned by [Jansen, Spink, Bateman et al. 1998], another study using log file data from Fireball also exists. The results are described in a German FH-Diplomarbeit by [Röttgers 1999].

- The 1999 Fireball study, done by [Röttgers 1999] using two one-week log files from Fire-

[23] See also [Jansen, Spink, Bateman et al. 1998a], [Jansen, Spink, Saracevic 2000]

[24] [He, Göker 2000], who analyzed the same log file with another goal, mention that the data was collected in a 49 minute time interval from 00:00:00 to 00:49:19 on 10th March 1997. Whether 00:00:00 is the local time of the Excite log server or something other like Greenwich Mean Time (GMT) is not mentioned.

[25] The exact number of queries examined seems to be 51,474, 51,473 or 51,453: [Jansen, Pooch 2000] report 51,473 queries in the description and 54,573 in Table 1. The sum when counting together the session length data in Table 1 is 54,595. [Jansen, Spink, Bateman et al. 1998] originally reported 51,473 queries seven times in their abstract, text and Table 1, one time on page 11 they mention 51,474 queries (which is the sum when summing up the number of queries in their Table 5), and when summing up the number of queries in their Table 2, the sum is 51,453. In [Jansen, Spink, Bateman et al. 1998a] there can be found some times 51,453, some times 51,473, and one time 51,474. However the variation in the number of queries is very consistent over time. In [Jansen, Spink, Saracevic 2000] we find again in the abstract "*We analyzed transaction logs containing 51,473 queries …*" and in the text p. 211 "*We classified the 51,474 queries as …*". In [Spink, Bateman, Jansen 1998] the study is referenced with 51,472 queries. [He, Göker 2000] examined the same data set with another focus. They report 51,474 queries.

[26] Parts of the results are also published in [Hölscher, Strube 2000] and [Hölscher 2000]. It is important to note that [Jansen, Pooch 2000] cite a number of facts from [Hölscher 1998] which are not included in the two-page-WebNet'98-paper and the 13-slides-WebNet'98-presentation from Hölscher. Hölscher reported these facts in his talk and included them in a document sent to Jansen. This document has also been made available to the author and is referenced as [Hölscher 1998a].

[27] Besides its German database, Fireball also uses AltaVista's database [Hölscher 2000]. [Röttgers 1999] reports that about 10% of the requests in the 1999 Fireball log file she analyzed had been routed to AltaVista

[28] The papers [Silverstein, Henzinger, Marais et al. 1998] and [Silverstein, Henzinger, Marais et al. 1999] are nearly identical. The 1999 SIGIR version has some additional clarifications compared to the 1998 Technical Report.

ball, with more than 6 Million requests each, collected in 1999 (Jan/Feb). Both log files were processed separately because they were in different formats. There had been an overlap of one day.

From these studies, we can only get statistical information about Web usage, because only log files are analyzed, or in the case of the Fireball studies only summarized data from log files. In addition, there are also some methodical problems when interpreting the log data. Among them are general questions like how to define or detect a user-session, or how to interpret the data not knowing which queries came from end-users, professional information seekers, meta-search engines[29] or softbots (intelligent software agents for the internet). Other points concern specific problems with the log data, like for example the fact that with the Excite data it was not possible to make a distinction between null queries[30] and relevance feedback queries. Nevertheless, there are some general trends worth mentioning:

- The average search session contains roughly two queries
- The average length of a query is around two keywords, with an increasing tendency
- The majority of the queries do not contain Boolean operators or modifiers like "+" or "NEAR".
- In the majority of cases, people do not go beyond the first page of results
- Topics people are looking for come from all conceivable areas, including sexual topics. They also seem to be influenced by trends.

Additionally, the studies contain information about a number of other points like query modification, the use of relevance feedback, or correlation of searched items. There are a number of other studies which follow specific goals, and also work with log files from search engines. Two examples are [Lau, Horvitz 1999] and [He, Göker 2000]. Parts of their findings will be cited in the context of the large scope Web-searching studies in order of commenting their results.

- [He, Göker 2000] analyzed the same 51,000 activity[31] Excite log file as Jansen et al. and an additional eight-day log file from the intranet search engine[32] of Reuters Ltd., collected in 1999 (March), and containing 9,534 activities. Their goal was to develop a methodology for deriving reasonable session breaks in Web queries.
- [Lau, Horvitz 1999] analyzed a 200 KB portion of a 48 MB one-day log file from Excite with 4,690 queries, collected on Tuesday, September 16 1997, and containing approximately one million transaction records. Hand-tagging this data, they constructed probabilistic models focusing on temporal patterns of query refinement.

[29] Taking the INSYDER system as an example for a meta-search engine, a five-term-query submitted to INSYDER will, due to the underlying query model, automatically result in six queries to every used search engine like AltaVista or Excite: one five-term-query containing all five terms, and one additional query for every single term. So taking this five-term-query entered by the user as the whole sample, it will result in an average query length of 1.66 terms in AltaVista or Excite despite the fact that it was entered as a 5-term query in the INSYDER system. Other meta-search engines may use other query models, but statistical falsifications are nevertheless likely.

[30] [Jansen, Pooch 2000] cite [Peters 1993] who shows that users enter null queries during the normal search process. [Kirsch 1998] reported for Infoseek "*Actually, our most popular query is really just an empty query box.*"

[31] Like [Silverstein, Henzinger, Marais et al. 1999] differentiate between "requests" and "queries", [He, Göker 2000] differentiate between "activities" and "queries".

[32] A local version of AltaVista.

In the literature, the usage of concepts like "query", "request" or "activity" is not always 100% consistent. A number of authors try to clarify their wording for their studies. Unfortunately this clarity is not always really present. In what follows, the attempt will nevertheless be made to homogenize the usage of the terms, at least in the context of this thesis. "Request" will be used for one transaction record in a log file (called "activity" by [He, Göker 2000]), being a query, a unique query, a modified query, an identical query, a null query, or a request for additional result screens. "Query" will be used in the way proposed by [He, Göker 2000] only for "*forming and modifying a search statement*". Sending it to the search engine as a request is included in this definition. So the broader usage of "query", like that of Jansen et al., is narrowed. One search with no change of the search string and three result screens viewed will be one query but three requests. For cases where it is not clear whether the authors are talking about queries or requests, the wording "queries / requests" is used.

[Jansen, Spink, Bateman et al. 1998] report an average of "*2.84 queries per user*". Ignoring identical queries[33] the average was 1.6 [Jansen, Spink, Saracevic 2000]. In terms of the previously mentioned homogenization, they reported 2.84 requests per user, or 1.6 queries per user. [Silverstein, Henzinger, Marais et al. 1999] report an average of 2.02 queries / requests per session[34]. The possible distortion of results due to automatic search agents can be seen in the results of the AltaVista study with a very high standard deviation of 123.4 and a maximum number of 172,325 queries in one session[35]. [Hölscher 2000][36] and [Röttgers 1999] do not report the number of queries per user or session. Ignoring the methodological problems involved in defining a session, and involved in making a distinction between requests and queries, we can get the impression of the number of queries per session shown in Figure 6. The log file hand-tagged by [Lau, Horvitz 1999] revealed in this context that the users performed an average of 3.27 queries / requests[37] per goal[38], and 4.28 queries / requests per day[39].

---

[33] In the examined Excite log data no differentiation was possible between an identical query entered by the user and a request for further result pages of an already displayed query, which had also been logged as an identical query.

[34] It's not clear if this 2.0 is with or without identical queries. It may be 2.0 queries per session, but could theoretically also be 2.0 requests per session. Interestingly enough [Silverstein, Henzinger, Marais et al. 1999] compare their 2.0 with the 2.8 from [Jansen, Spink, Bateman et al. 1998]. [Jansen, Pooch 2000] do the same comparison using the 2.0 and the 1.6.

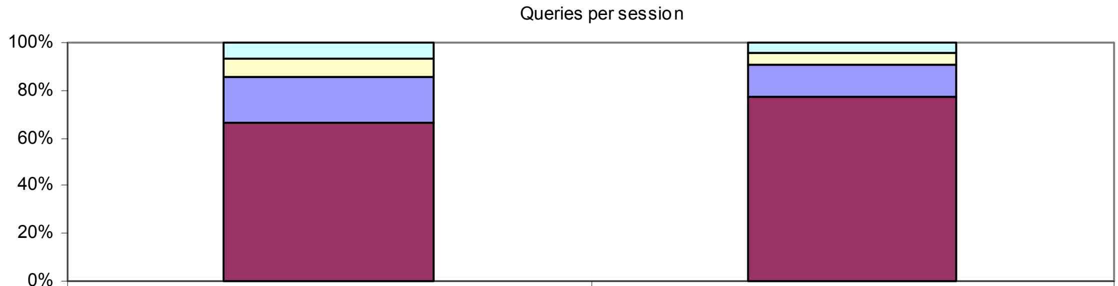
[35] This single session contains 3 times more queries than the whole Excite study, but only 0.017% of the number of queries of the AltaVista study.

[36] For here and the remainder "*[Hölscher 2000] does not report* " stands also for [Hölscher 1998], [Hölscher 1998a], [Hölscher, Strube 1999], and [Hölscher, Strube 2000]

[37] They report the average number of queries, and it is not clear if this is done using Jansen et al.'s broader method, or the more narrowed method used in this thesis.

[38] Information goals were defined, and the researchers detected changing of goals by using an ontology, inspecting the Excite log file, and interpreting the sequences of the query terms of the users.

[39] The authors do not describe how they extracted the 200 kB from the 48 MB log file. So the basis upon which this average of 4.28 queries or requests per day and user was discovered is not completely clear.



	Excite [Jansen, Spink, Bateman et al. 1998]	AltaVista [Silverstein, Henzinger, Marais et al. 1999]
> 3 queries per session	1,214	4.5%
3 queries per session	1,321	4.4%
2 queries per session	3,501	13.5%
1 query per session	12,068	77.6%

Figure 6: Number of queries per session according to [Jansen, Spink, Bateman et al. 1998], [Silverstein, Henzinger, Marais et al. 1999]

Discussing the number of queries per session, there are some interesting results from [He, Göker 2000]. Their results show that changing the threshold between two activities of the same user to define a new session start can influence the interpretation of the log file. In the case of the Excite log file, also used by Jansen et al., a threshold of 5 minutes leads to 58% of sessions with one single query and one single result screen. Using no threshold for the log file representing 49 minutes of data, the percentage of sessions with one query and one result page drops to 44%[40], whilst with a threshold of one minute it increases to 84%. [He, Göker 2000] recommend using thresholds of between 10 and 15 minutes. This leads, in the case of the Excite log file, to a percentage of 52.7% to 48.5% of sessions with just one query and one result screen. The threshold for the AltaVista-study used by [Silverstein, Henzinger, Marais et al. 1999] was five minutes leading to a result of 63.7% sessions with just one request and one result screen. Using the threshold recommended by [He, Göker 2000] must lead to a smaller number of this type of sessions.

The average length of a Web-query has already been mentioned in the comparison between traditional IR and Internet searching. Taking the large studies discussed here, we find values of 1.66 [Hölscher 2000], 1.65 [Röttgers 1999][41], 2.21 [Jansen, Spink, Bateman et al. 1998] and 2.35 [Silverstein, Henzinger, Marais et al. 1999][42] terms per query. On the same large-scale data level, [Kirsch 1998] reported an average query length of approximately 2.2 words for Infoseek users. If we look at the detailed data displayed in Figure 7, we see that in most of the cases shown, people enter one or two keywords. Whereas in the Excite and the AltaVista study two keywords are slightly more often used than one, in most of the cases in the Fireball studies just one keyword is

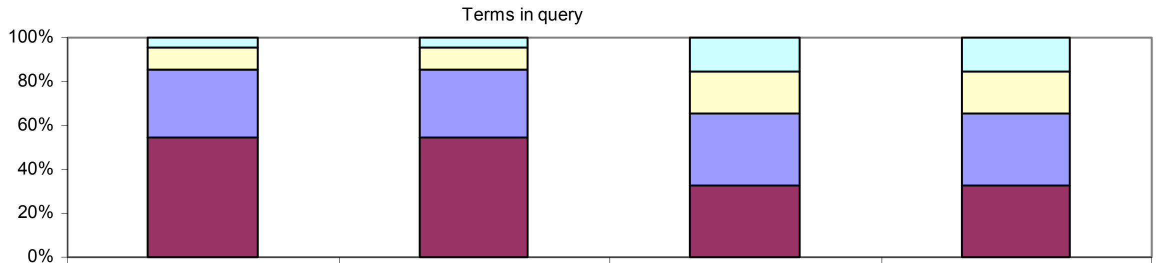
---

[40] This 44% from [He, Göker 2000] does not fit together with the 58% from [Jansen, Spink, Bateman et al. 1998]. The 44% without threshold from [He, Göker 2000] stands for sessions with just one activity, which means one query, one result screen. [Jansen, Spink, Bateman et al. 1998] report that 58% of the users viewed one page. The number of users seems to correspond to the number of sessions, including an error. Sentences like "*Some users used only one query in their session,* […]"[Jansen, Spink, Bateman et al. 1998] supports the impression that no users had more than one session. [Jansen, Spink, Bateman et al. 1998] do not mention a threshold for their study, and [Jansen, Pooch 2000] even criticize [Silverstein, Henzinger, Marais et al. 1999] for using a five-minute threshold, because this "*has the effect of 'shortening' the sessions, reducing the query per session count.*" [Jansen, Pooch 2000].

[41] Calculated by dividing the 9,327,458 non-unique words of the simple searches through the 5,649,571 simple searches.

[42] [Silverstein, Henzinger, Marais et al. 1999] write „*The same average query length* [2.35] *was found by [Jansen et al. 1998].*" Here they took the value listed in the text of [Jansen, Spink, Bateman et al. 1998], instead of the probably more likely value of 2.21 from their Table 1. See here also [18] and [46]. The 2.35 from [Jansen, Spink, Bateman et al. 1998] instead of the 2.21 is also cited by a number of other authors like [Hawking, Craswell, Thistlewaite et al. 1999], or [Röttgers 1999]

used. Whilst the nearly equal distribution for the two Fireball studies is understandable, it is astonishing for the Excite and the AltaVista study. Empty requests are not displayed in the table. [Hölscher 2000] and [Röttgers 1999] do not report the number of empty requests. [Jansen, Spink, Bateman et al. 1998] list null queries[43] as being 5% of all requests. [Silverstein, Henzinger, Marais et al. 1999] report empty requests as 15% of all requests, and null queries as 20.6% of all queries. As explained above in the case of Excite, these null queries could either be user null queries or relevance feedback queries. For AltaVista no interpretation is provided.



	1998 Fireball [Hölscher 2000]	1999 Fireball [Röttgers 1999]	Excite [Jansen, Spink, Bateman et al. 1998]	AltaVista [Silverstein, Henzinger, Marais et al. 1999]
more than 3 terms	691,119	4.25%	7,582	12.6%
3 terms in query	1,683,129	10.36%	9,242	15.0%
2 terms in query	5,005,653	30.80%	16,191	26.0%
1 term in query	8,873,001	54.50%	15,854	25.8%

Figure 7: Terms in query according to [Hölscher 2000], [Röttgers 1999][44], [Jansen, Spink, Bateman et al. 1998][45], and [Silverstein, Henzinger, Marais et al. 1999]

Do the null queries and the identical queries have any impact on the reported average terms per query? Yes, they do – at least in the case of the null queries. [Silverstein, Henzinger, Marais et al. 1999] explain their Table 2 (statistics concerning the number of terms per query) thus: "*The mean and standard deviation are calculated only over queries with at least one term.*" [Röttgers 1999] does not report an average query length. [Hölscher 2000] does not report null queries. [Jansen, Spink, Bateman et al. 1998] do not explain their method, but taking their Table 2 it can be worked out that the average of 2.21 must be calculated including the null queries and the identical queries. Omitting the null queries, the average query length for the Excite study is 2.33[46]. Turning now from null queries to identical queries [Hölscher 2000] mentions that his average of 1.66 was calculated including identical queries. For [Röttgers 1999] the same algorithm was used. For [Jansen, Spink, Bateman et al. 1998] we have seen that they must also have included them. [Silverstein, Henzinger, Marais et al. 1999] do not explain whether they included identical queries in the calculation or not. Their reported data shows that they were able to make a distinction between requests for a new result screen (31.8 % of all non-empty requests, 27.0 % of all requests) and exact-same-as-before requests (5.0 % of all non-empty requests, 4.2 % of all requests)[47]. It is not implausible that short queries, say with one term only, tend to lead to larger result sets and that users may look

[43] Because a null query has no additional result screens, one null query normally corresponds to one empty request

[44] [Röttgers 1999] considered only requests using the simple interface ("*Express-Suche*") of Fireball, which represent 90% of the overall requests. The requests from the advanced interface ("*Detailsuchmaske*") are not discussed.

[45] [Jansen, Spink, Saracevic 2000] list 15,874 queries with one term instead of the 15,854 from [Jansen, Spink, Bateman et al. 1998]. The other figures are identical.

[46] And so the above-criticized statement of [Silverstein, Henzinger, Marais et al. 1999] about the same average query length is at least nearly correct, despite the fact that the maximum number of terms per query was 10 in the case of the Excite study and 393 for the AltaVista study.

[47] The 43% identical queries from [Jansen, Spink, Bateman et al. 1998], should correspond to 27.0 % + 4.2 % = 31.2 % identical queries from [Silverstein, Henzinger, Marais et al. 1999]

at more result pages when getting larger result sets. So if [Silverstein, Henzinger, Marais et al. 1999] calculated their average, in contrast to the others, without requests for a new result screen, a comparable value to the 1.66 of [Hölscher 2000], the 1.65 of [Röttgers 1999], or the 2.33 without null queries for [Jansen, Spink, Bateman et al. 1998] could be lower than the reported 2.35.

Another factor influencing the terms per query number reported is the definition of a "term". For [Jansen, Spink, Bateman et al. 1998] "*A term is any unbroken string of characters (i.e. no space between characters)*". [Silverstein, Henzinger, Marais et al. 1999] "*use the term* query term *to denote a word or a quotation-mark-enclosed phrase*." So a query like `"Visualization of Search Results from the World Wide Web"` would be a one-term query for [Silverstein, Henzinger, Marais et al. 1999], but a nine-term query for [Jansen, Spink, Bateman et al. 1998]. On the other hand for Web specific queries [Silverstein, Henzinger, Marais et al. 1999] simplified the analysis by treating a query like `host:www.acompany.com` as a four term query containing the words `host, www, acompany,` and `com`. For [Jansen, Spink, Bateman et al. 1998] this would be a one-term query. [Hölscher 2000] and [Röttgers 1999] do not explain their algorithms.

When one looks at some smaller scaled studies focusing on Web search, the range of values broadens up. Sometimes similar figures can be found for specific use cases like single site search services. So [Croft, Cook, Wilder 1995] got an average query term number of 2.32 in an investigation of 25,321 queries from the THOMAS system (US Congress Bills and Congressional Record). Sometimes the figures differ greatly for general Web search scenarios. [Hölscher, Strube 2000] found an average query length of 3.64 words per query in a controlled experiment with 12 Web experts, compared to the 1.66 received from the analysis of the log data of the Fireball search engine. In a second 2x2 matrix experiment, they got a non-significant difference of 2.61 words per query for 12 Web experts vs. 2.32 for 12 Web novices. [Körber 2000] performed a two-task-experiment with 9 Web experts and 9 Web novices and got an average query length of 6.33 and 3.78 for the experts vs. 3.11 and 3.89 for the novices.

Returning to the analysis of search engine log files, a study by [Lau, Horvitz 1999] shows some interesting patterns. For their 4,690 queries / requests portion of the Excite log file, they calculated an average query length of 2.30 words. This value supports the findings of Jansen et al. and other figures from Excite shown below. [Lau, Horvitz 1999] used an ontology of information goals to categorize the queries and detected different average query lengths for different information goals. They found, for example, a mean of more than three words per query for their category "Education", and a mean of fewer than two words for "Places" or "Recreation and Sports".

All the figures here presented regarding the average number of terms per query only represent glimpses of a largely unexamined field. As has been mentioned, the general impression is that there are far fewer terms in the queries than in traditional IR, and that the number of terms per query when searching for information in the Web is around two. Two findings listed by [Spink, Xu 2000] are very interesting in this context:

- The number of terms per query increases over time (at least for Excite)
- The number of terms per query differs between countries (at least for Excite[48])

---

[48] As mentioned in [24], the data of the Excite study by Jansen et al. was collected between 00:00:00 and 00:49:19. It would be interesting to know on which time zone this information from [He, Göker 2000] is based. The time of day may have influenced the composition of the user population from different continents or the topics searched, and may therefore have influenced the findings of the study.

The mean length of Excite queries between mid 1996 and mid 1999 was 2.4. In 1996 it was 1.5 for US, UK, and European users[49], in 1999 it was 2.6 for US/UK users and 1.9 for European users [Spink, Xu 2000]. If we consider now that the data [Hölscher 1998] used came from Fireball, and also that this is a European (German) search engine, the difference to the other two studies makes a lot more sense. Figure 8 brings the research into average terms per query into a frame. Please note that it is only clear whether null queries are included in the calculation of the average length or not in two of the numbers shown.

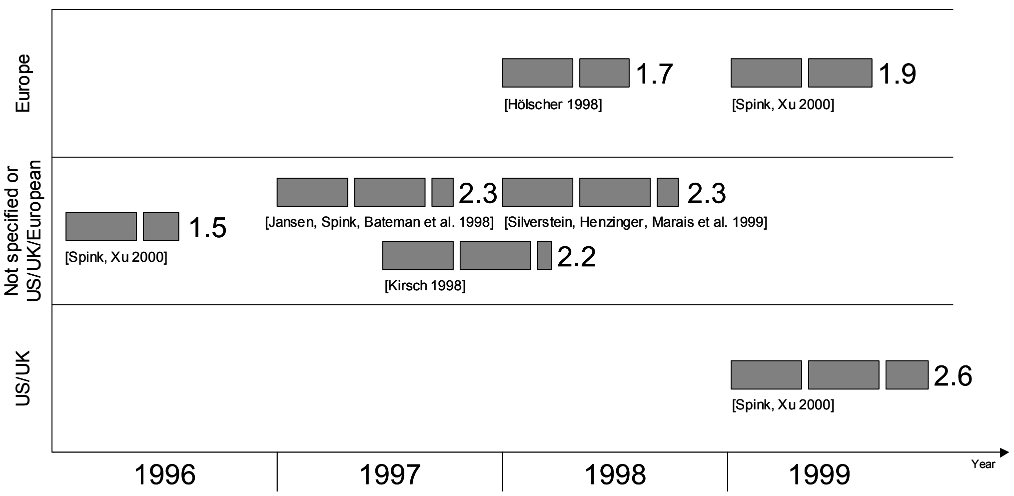


Figure 8: Average number of terms per query over time and area[50]

The growth of the query length over time is an interesting phenomenon. Two possible explanations are, that the overall growth of the Web makes it necessary to have more specific queries to find the requested information via higher precision and lower relative recall, or that the users of the Web get more mature in specifying queries over time, or both. Both could also explain the difference between Europe and US/UK. The number of non-English Web pages seems to be still smaller than the number of English Web pages[51], and the large-scale usage of the Web started in Europe later than in the US. The difference between US/UK and Europe could also be caused by structural differences in the languages[52]. All this is speculation at the current point of time. In the literature there seems to be so far no investigation or validated explanation of this trend.

When observing the complexity of the queries, there is a clear picture: most of the queries are simple. The majority of the queries do not contain Boolean operators or modifiers like “+” or “NEAR”. [Hölscher, Strube 2000] report at least 72% of the queries as being without any modifiers, [Röttgers 1999] reports 79.34% of all queries from the simple interface without any modifiers,

[49] The differentiation between UK and European users is particularly noteworthy. The UK is definitely a part of Europe, despite the fact that people from Great Britain often make a distinction between Europe and Great Britain. In the context of the cited study the term “European” will be used in the same way as in the cited study where “European” means “people from Europe without UK”.

[50] Year of the data not listed in [Kirsch 1998]. Assumed here to be 1997/8. For [Jansen, Spink, Bateman et al. 1998] the calculated 2.33 is taken without null queries.

[51] [Moore, Murray 2000] report 84.7% US pages and 15.37% international pages (for pages with fewer than 200 KB, limited by their crawling technique). Probably most of the US pages will be in English, and some of the internationals too.

[52] [Hölscher 2000] speculates that the difference in the query length between the Excite and the Fireball study is caused by the fact that in German more concepts are expressed as compound words (Komposita), where in English two words must be used.

[Jansen, Spink, Bateman et al. 1998] at least[53] 81%[54] of all queries and [Silverstein, Henzinger, Marais et al. 1999] 80% of all distinct queries. Figures for the usage of a phrase indicator like quotation marks are given by [Jansen, Spink, Bateman et al. 1998], [Hölscher, Strube 2000] and [Röttgers 1999]. In the Excite study it was used in 6.4% of all queries / requests, or by 5.6% of all users, and in the 1988 Fireball study it was used in 8.6% of all queries / requests. When focusing on users, instead of queries for the modifiers too [Jansen, Spink, Bateman et al. 1998] show that at least 86%[55] of the users did not use any modifiers. Besides the pure usage statistics [Jansen, Spink, Bateman et al. 1998] documented high error rates for queries with modifiers. For the most-used Boolean modifier "AND", they list an incorrect usage in 32%[56] of the cases or for 50% of the users. When studying the detailed analysis, it becomes clear that at least some of the cases of incorrect usage have to do with a specialty of Excite, which requires that Boolean operators are all in uppercase letters. From the appearances of AND they found, 36% were mistakes if considered as Boolean operators because they had not all been written in uppercase letters. The picture will be similar for the other Boolean operators. Their figures regarding incorrect use of the '-' (minus) modifier are particularly interesting. In their Table 4 they list incorrect usage in 95% of the cases without discussing this fact[57]. They just provide the example that minus is often used in phrases like "pre-teen" and that 38% of the users make incorrect use of the minus operator. Taking the figures from their Tables 3 and 4, there were only 88 queries with correct usage of the minus operator, but 146 users who correctly used the minus operator[58]. I won't criticize the Excite study as a whole. Not without cause, it is one of the most cited studies in the field of Web search engines, and a very important contribution to this research area. The authors state: "*We took great care in derivation of counts, but because of the 'messiness' of data there still may be errors – we estimate at less than 1%.*" [Jansen, Spink, Bateman et al. 1998]. Sometimes it seems to be more than 1%. As we will see later, for the publications about the results of the INSYDER project we also had some problems with stable results from number crunching through a series of publications.

Here again it is interesting to see a trend over time reported by [Spink, Xu 2000]. From 1997 to 1999 the use of operators (AND, OR, NOT, +, -) increased for Excite users from 22% of the queries to 28%, or using the method listed above, simple queries decreased from 78% to 72%.

---

[53] From the data the authors present it is not possible to calculate the intersection of the queries containing any operators. For some sub cases the authors deliver figures which are exactly the numbers when just summing up the single values (4,776 +/- queries in [Jansen, Spink, Bateman et al. 1998], or 5,323 Boolean queries in [Jansen, Spink, Bateman et al. 1998a]). This means none of the queries could have included more than one type of modifier, which seems to be very unlikely. Nevertheless, the same method was used here to calculate the percentage of queries without operators. The real number of queries without operators would have been higher if there had been any queries that contained more than one type of operator. The 72% of the Fireball study was calculated here using the same method.

[54] Again, looking in different publications from the same authors about the same study of the same dataset, different figures can be found. [Jansen, Spink, Bateman et al. 1998] and [Jansen, Spink, Saracevic 2000] list 4,094 queries with "AND", [Jansen, Spink, Bateman et al. 1998a] and [Jansen, Spink, Saracevic 1998] 4,798. The other figures are for "OR" 177 / 132, "AND NOT" 105 / 120, and the usage of parentheses 273 / 273.

[55] To calculate this value, the same method was used as described in [53].

[56] [Jansen, Spink, Saracevic 1998] report 26.30% instead of the 32%

[57] The 95% can be found in [Jansen, Spink, Bateman et al. 1998] and [Jansen, Spink, Saracevic 2000]. None of the documents comments on the fact. In [Jansen, Spink, Saracevic 1998] 97.42% incorrect minus-queries are reported in Table 4. In this document the fact that there is a high percentage of mistakes when using the '-' modifier is also mentioned in the text.

[58] 1,766 queries – 1,678 queries incorrect = 88. 508 users – 363 users incorrect = 146. Exactly the same figures can be found in [Jansen, Spink, Saracevic 2000]. [Jansen, Spink, Saracevic 1998] list 2,573 queries – 2,495 queries incorrect = 78 for the same dataset. There are no figures for the users.

How many results do people have a look at? All three search engines used in the studies presented the results of a query at the time of the studies in pages with 10 hits each. The log data used in the studies only permitted examining the number of hits presented to the user[59], not the number of hits really viewed by the user. Nevertheless, it is interesting to see that the majority of the users do not go beyond the first page of results with at maximum 10 hits. It is difficult to compare the number of pages really viewed between the different studies. [Silverstein, Henzinger, Marais et al. 1999] report an average of 1.39 viewed result screens per query[60] for their study and compare this to an average number of 2.21 screens per query from [Jansen, Spink, Bateman et al. 1998]. But [Jansen, Spink, Bateman et al. 1998] do not report the average number of screens per query, but the average number of screens per user. ("*The mean number of pages examined per user was 2.21.*"[61, 62]). Taking the values from Jansen et al. as screens per query instead of screens per user also happened in other publications, like for example [Amento, Hill, Terveen et al. 1999] "*… showed that 86% of all users looked at no more than 30 pages returned in response to their query [6]*[63]." Remembering that [Jansen, Spink, Bateman et al. 1998] reported an average of 1.6 queries per session (without the identical queries, which may be mainly requests for further result pages), we could hypothesize that their reported average page view number compared to the [Silverstein, Henzinger, Marais et al. 1999] value is too high for this factor. If we conduct an intellectual experiment and divide the 2.21 screens per session by the 1.6 queries per session, we get a theoretical value of 1.38 screens per query for the Excite study. The corresponding AltaVista value was 1.39. On the other hand, when taking the 51,474 queries that correspond to the same number of pages, and the 18,133 users which correspond to the same number of sessions reported by [Jansen, Spink, Bateman et al. 1998], every user must have seen an average number of 2.84 screens and not 2.21. Omitting all the null queries, even with the knowledge that they also include the "more-like-this" requests, we get an average of 2.70. If we do a comparable intellectual experiment with the 285,474,117 sessions and 843,445,731 non-empty requests reported by [Silverstein, Henzinger, Marais et al. 1999] we get an average of 2.95 screens per session, or 3.48 when taking all 993,208,159 requests into account. Stopping speculations and returning to the reported facts, it seems to be clear that in most cases people do not go beyond the first page of ten results of their query or session. [Jansen, Spink, Bateman et al. 1998] report 58% for the session. From the numbers delivered by [Hölscher 1998a],

---

[59] To be fully correct, the log data only contained the number of pages requested by the user, not the number of hits really presented. For example for searches with less than 10 results the researchers could only see that the user requested just the first result page, not the real number of hits presented.

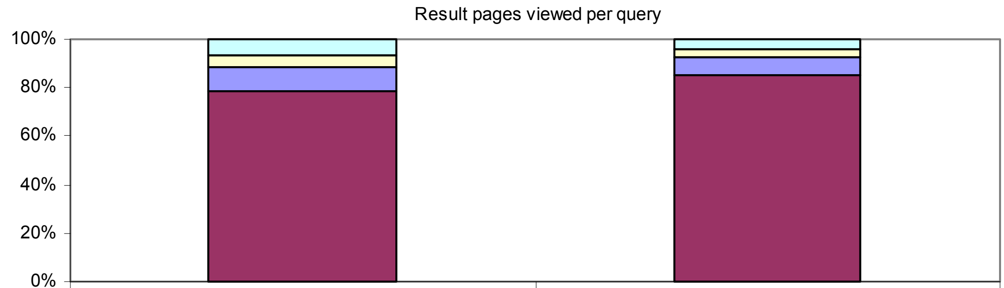
[60] The Title of their Table 7 is "*Statistics concerning the characteristics of result screen requests in sessions.*", but the numbers they report are titled "*screens per query*".

[61] In [Jansen, Spink, Saracevic 2000] we can find ("*The mean number of pages examined per user was 2.35.*") instead of 2.21, but when taking the figures from Table 7 of [Jansen, Spink, Bateman et al. 1998] and the identical Table 4 of [Jansen, Spink, Saracevic 2000] an average of 2.2185 (should be 2.2 if rounded) can be calculated for both cases if the number of cases listed is taken, or 2.146 (should be 2.1 if rounded) if the number of all users is taken. Interestingly, the tables only contain data for 18,101 users, whereas in both papers the number of users is specified as 18,113. If the figures are correct it may be that the remaining 12 users only entered null queries with no result pages. Another puzzle is that the tables just contain 40,157 queries. Even when assuming that all or some of the 2,584 null queries are not included, the sum is a maximum of 42,741 queries. Where are the remaining roundabout 8,700 queries?

[62] The discussion of the values from [Jansen, Pooch 2000] for the Excite study is skipped here, because their quality seem to be even worse than the original figures.

[63] [6] = "*Jansen, B. J., Spink, A., Bateman, J., and Saracevic, T. Searchers, the Subjects They Search, and Sufficiency: A Study of a Large Sample of EXCITE Searches, submitted to WebNet'98*". Apparently [Amento, Hill, Terveen et al. 1999] used a preliminary version of the [Jansen, Spink, Bateman et al. 1998a] WebNet'98-paper here. The final version of the paper did not contain any detailed information about the number of viewed pages.

78.4% can be calculated[64] for the query. [Silverstein, Henzinger, Marais et al. 1999] report 63.7% for the session[65] and 85.2% for the query. Detailed data on query level, shown in Figure 9, is only available from [Hölscher 1998a], and [Silverstein, Henzinger, Marais et al. 1999].



	Fireball [Hölscher 1998a]	AltaVista [Silverstein, Henzinger, Marais et al. 1999]
more than 3 pages	664,951	4.3%
3 pages viewed	411,862	3.0%
2 pages viewed	1,000,694	7.5%
1 page viewed	7,543,840	85.2%

Figure 9: Result pages viewed per query according to [Hölscher 1998a][66], and [Silverstein, Henzinger, Marais et al. 1999]

One of the differences between classical IR and Web searching is that a near miss in a Web search can nevertheless lead to the requested information. This aspect is taken into account in Web search models. Empirical evaluations studying users instead of analyzing log files of search engines, like [Hölscher 2000] or [Körber 2000], show that browsing episodes are not only part of the models, but really occur quite often in reality. The number of screens viewed from the search engine is an important parameter when talking about the search engine part of an information-seeking episode, but it is important to remember that this – in contrast to classical IR studies – is not correlated with the number of pages or documents the user really viewed.

When studying the topics people look for when searching the Web by analyzing the most commonly occurring keywords in [Jansen, Spink, Bateman et al. 1998], [Silverstein, Henzinger, Marais et al. 1999] or [Sullivan 2000a], the impression is that the topics come from all conceivable areas, they definitely contain sexual topics, and they seem to be influenced by trends. Or in the words of [Jansen, Spink, Bateman et al. 1998]: "*There is a lot of searching about sex on the Web, but all together it represents only a small proportion of all searches. ... A great many other subjects are searched, and the diversity of subjects searched is very high.*" What they do not comment

[64] [Hölscher 1998a] contains a table listing that 59.1% (9,621,347) of the 16 Million queries are requests for a first result screen, 12.85% (2,077,507) are requests for a second result screen, 6.66% (1,076,813) for a third, 4.11% (664,951) for a fourth, and so on. Summing up the request from the table, there are 0.5% missing of the reported 16,252,902 queries. Assuming that in most cases people will have a look at the result pages in a sequence beginning with the first screen, then the second screen, then the third screen, and so on, it can be calculated that having 9,621,347 requests for a first screen, and 2,077,507 requests for a second screen, there must have been 7,543,840 queries where only one screen had been viewed. Like other engines, Fireball allows one to follow hyperlinks to later or earlier result screens more than one step away from the current one directly. Assuming a sequence jumping around in the result pages with skipping single screens is ignored. [Jansen, Pooch 2000] list in their Table 1, for [Hölscher 1998] "*Number of Relevant Documents Viewed in a Session*", "*10 or less: 59.51% (9,621,347)*". Hölscher did not try to reconstruct sessions from the log file [personal mail communication with Christoph Hölscher 2000-01-15]. Probably [Jansen, Pooch 2000] misinterpreted here the table from [Hölscher 1998a], and took the percentage of requests for a first screen from all requests, as percentage of sessions with just one result screen.

[65] [Jansen, Pooch 2000] cite [Silverstein, Henzinger, Marais et al. 1999] with 85.2% for the session, but 85.2% is the figure for the query, for the session it is 63.7%.

[66] Numbers for [Hölscher 1998a] are calculated as described in [64]. Percentages are: one screen per query 78.4%, two screens 10.4%, three screens 4.3%, more than three screens 6.9%.

on is the dynamic changing process of the top topics over time, but that is clear when discussing a dataset which covers only a portion of a single day. In addition to the influence of trends, it doesn't take much research to realize that the topics people are looking for will at least additionally be dependent on the country where they are living.

2.3.2. User group differences

When considering that there are differences in Web usage or Web search behavior depending on the characteristics of a user population, it will be interesting to know what the characteristics of the users who are behind the results presented in the last chapter are. One factor influencing the behavior we have already seen is the location where people live – or is it their Internet maturity, or their mother language?

[Spink, Bateman, Jansen 1998] performed a theoretically appealing study trying to find out more about the users of the Excite search engine by doing an interactive survey in April 1997, one month after the log data for [Jansen, Spink, Bateman et al. 1998] had been collected. The results from the 357[67] users who responded are to a certain degree interesting[68], especially the fact that single search sessions appear often to have been part of a longer search process (at least valid for the participants of the survey). In general, the sample seems to be too small and the participants are from a too specific group of users who took the time to answer the questionnaire, to get a valid impression about the Excite user population, their characteristics, their goals, or their behavior. A much broader picture about the user population, not for a specific search engine, but for the Web itself, comes from the series of the Georgia Institute of Technology's Graphics, Visualization, and Usability Center WWW User Surveys [GVU 1994 – 1998], or from surveys carried out by various consulting companies. One important trend shown very clearly by [Pitkow, Kehoe 1996] is the continuous change over time of the characteristics of the people using the Internet. Therefore in the case of a log file analysis from search engines, all data of the characteristics of a user population must be drawn contemporary to the time frame of the log file. These criteria are fulfilled by [Spink, Bateman, Jansen 1998], but their work is heavily influenced by the problems described by [Pitkow, Kehoe 1996] for all WWW-based surveys: self-selection and sampling. In general, when studying search behavior and user characteristics in the Web there is always a problem: The broader the statistical basis of data about the search behavior, the more difficulty one has in getting detailed information about the characteristics of the user population and vice versa. The excursus about the user population that is responsible for the trends reported in the last chapter will be stopped here. The lesson learned is that drawing conclusions from the results should be done with care when trying to transfer them to a special user population like people from small and medium size enterprises looking for business information in the Web. We will now turn to the question of whether there is any material available which focuses on such a special population or if there are any more differences known between users groups other than the already reported differences between UK/US and European users respectively the differences between the users of the different

[67] In the abstract [Spink, Bateman, Jansen 1998] write "*Three hundred and fifty-seven (357) EXCITE users responded ...*", and in the result section "*Only 316 of the 480 returned survey forms contained usable data.*" The maximum number of reported answers for one of the 18 questions was 301 answers.

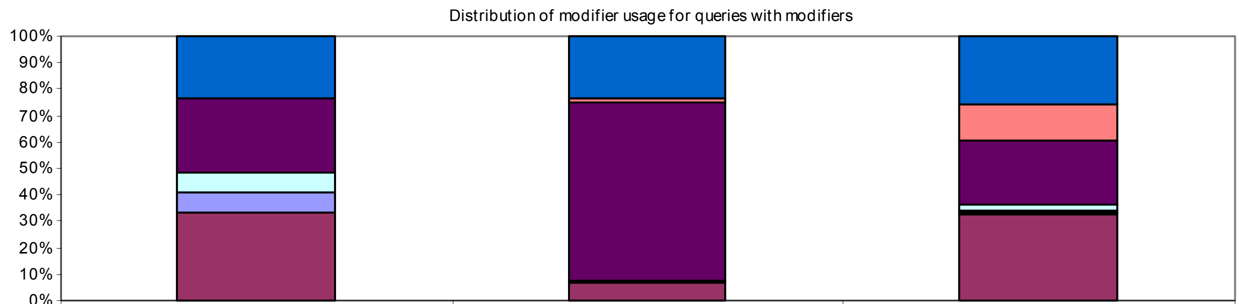
[68] Some other results seem to be rather banal, like for example "*Interestingly, the largest group of respondents were searching EXCITE from home ...*" [Spink, Bateman, Jansen 1998]. Indeed very surprising for a dataset collected in a five-day period from Friday to Tuesday, with the heaviest usage of the survey form on Saturday, that 36% of the respondents used their computer at home.

search engines.

At this stage, four studies should be mentioned. The investigation of [Meyer, Sit, Spaulding et al. 1997] about age group and training differences in World Wide Web navigation, the studies of [Hölscher, Strube 2000] and [Körber 2000] about differences in Web search behavior between internet experts and newbies, and the exploration by [Wang, Hawk, Tenopir 2000] into the influence of search experience, affective states, and cognitive style on Web search process and success.

In an experiment involving 20 participants without significant WWW experience, [Meyer, Sit, Spaulding et al. 1997] detected that the "older" participants (ages 64 to 81) took more steps to reach a target in a set of 19 locally stored Web pages, than the "younger" ones (ages 19 to 36). The average of the nine tasks for older adults was 9.7 steps, for younger adults 6.4. Unfortunately there is no statistical validation of their results included[69]. The training effects were also interesting, showing that the 11 users (7 old / 4 young) who got a "hands-on" navigation tutorial had an average of 7.8 steps compared to an average of 9.3 steps for the other 9 users (6 old / 3 young) who just got a "hands-off" description of navigation methods.

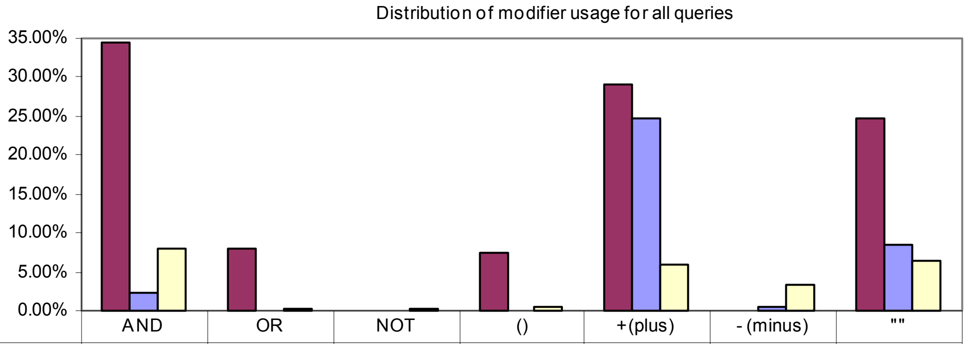
In a first step, [Hölscher, Strube 2000] performed interviews with 12 established internet experts and from this developed a process model of the information seeking process in the Web. Their model is comparable to the ones introduced in Chapter 2.2. What is really interesting is the fact that in a second step, the experts had to perform a number of real-world information tasks using their own choice of strategy and search engine. Analyzing these information seeking episodes, the authors calculated transition probabilities between the steps of the model. It was found that in 47% of the cases, using a search engine led to a browsing episode of varying length or that the experts often switched back and forth between browsing and querying. Also interesting was the fact that the average query length was 3.64 words, instead of the considerably shorter averages found in the large scope Web-searching studies. Another difference was the usage of the different types of modifiers in the queries shown in Figure 10 as a relative distribution for all queries using modifiers, and in Figure 11 as a percentage of all queries.



	12 Experts [Hölscher, Strube 2000] / [Hölscher 2000]	1998 Fireball [Hölscher, Strube 2000] / [Hölscher 2000]	Excite [Jansen, Spink, Bateman et al. 1998]
""	40	1,401,738	3,282
- (minus)	0	77,531	1,766
+ (plus)	47	4,034,312	3,010
()	12	15,738	273
NOT	0	10,372	105
OR	13	13,817	177
AND	56	390,272	4,094

Figure 10: Distribution of modifier usage for queries with modifiers 12 Experts / Fireball / Excite

[69] This is a particular problem because the two independent variables are not counterbalanced. This can be seen from the age differentiation: 53.8% of the older got the "hands-on" training, compared to 57.1% of the younger, and in the training differentiation 63.8% of the "hands-on" group were older, compared to 66.9% of the "hands-off.



	AND	OR	NOT	()	+(plus)	- (minus)	""
12 Experts [Hölscher 2000]	34.57%	8.02%	0.00%	7.41%	29.01%	0.00%	24.69%
1998 Fireball [Hölscher 2000]	2.40%	0.09%	0.06%	0.10%	24.82%	0.48%	8.62%
Excite [Jansen, Spink, Bateman et al. 1998]	7.95%	0.34%	0.20%	0.53%	5.85%	3.43%	6.38%

Figure 11: Distribution of modifier usage for all queries 12 Experts / Fireball / Excite[70]

In a second experiment, [Hölscher, Strube 2000] looked for the potential influence of Web expertise and domain knowledge on Web search behavior. 24 participants in 2x2 matrix experiment had to solve a set of five information-search problems from an economic domain. Among the findings are the following points:

- Double-novices (low Web expertise and low domain knowledge) had the highest proportion of query reformulations, chose the smallest number of target documents for closer examination, and viewed the highest proportion of irrelevant documents.
- Double-experts were overall most successful in their search behavior
- Double-experts showed the lowest percentage of backward oriented behavior like using the back button or returning to previous search engine results.
- In some cases double-experts followed a strategy of directly accessing Web sites related to economics. No other group displayed this behavior.
- Domain-experts spent significantly less time with domain-specific documents, than domain-novices.
- Web-experts used modifiers significantly more often (87% vs. 47%) and made far less formatting errors (1.9% vs. 19.6%) than Web-novices
- The average query length of the Web-experts was only marginally longer than that of the Web novices (2.61 vs. 2.32), but surprisingly the average query length of domain-experts was significantly shorter than that of domain-novices (1.97 vs. 2.96)

[Hölscher, Strube 2000] differentiated between technical Web expertise and domain-specific background knowledge. "*Participants which could rely on both types of expertise were overall most successful in their search behaviour.*" [Hölscher, Strube 2000]. Web expertise alone did not help to get higher effectiveness rates. On the other hand the authors report, discussing their first experiment, that there are differences in the way Web experts search the Web compared to the average user. For an overview covering studies dealing with the influence of expertise on retrieval success see [Hölscher 2000]. His summary is that an information-seeking process is positively influenced by all of the three expertise types defined by [Marchionini 1997]: domain knowledge, general information-seeking expertise, and system expertise. The differences between search experts and novices are clearer for measures which focus on the search process [Hölscher 2000].

Körber's study [Körber 2000] has already been mentioned. He also provides a number of other measures and findings like modifier usage, and query modification. Despite the fact that no statistical validation is provided, some of the results are worth presenting. His findings concerning the average number of terms per query are listed on page 35. [Körber 2000] supports the findings of other studies, i.e. that people often do not go beyond the first page of results presented. For his experiment with 18 users he used the German version of AltaVista, also presenting 10 hits per page. As demonstrated above, one has to be careful when comparing the measures used. [Jansen, Spink, Bateman et al. 1998] and [Silverstein, Henzinger, Marais et al. 1999] delivered the percentage of users looking only at the first screen in a session. [Silverstein, Henzinger, Marais et al. 1999] delivered the percentage of queries where only one screen is requested as well. This value can also be calculated for [Hölscher 1998a]. [Körber 2000] delivers the percentage of additional screens, beyond the first one, from all result screens[71]. The trend is the same. In the two-task-experiment, the experts had 16.3% and 29.6%, the novices 10% and 11% of additional screens from all result screens. For [Hölscher 1998a] a value of 40.5%[72] for the 1998 Fireball study can be calculated. For the 1999 Fireball study by [Röttgers 1999], the result is 37%. Assuming all identical queries as requests for viewing subsequent pages, [Jansen, Spink, Bateman et al. 1998] report 43% of additional screens. For [Silverstein, Henzinger, Marais et al. 1999] a value of 27% from all requests and 31.8 % from all non-empty requests can be calculated.

Figure 12 shows a comparison. There are more cases in the controlled experiments where subjects just look at the first result screen than with the user population responsible for the results of the large scale studies based on the analysis of logfiles. Despite the fact that the samples from [Hölscher 2000] and [Körber 2000] are small it is also an interesting trend that in controlled experiments, the users seem to use more keywords than in the large scope Web-searching studies.

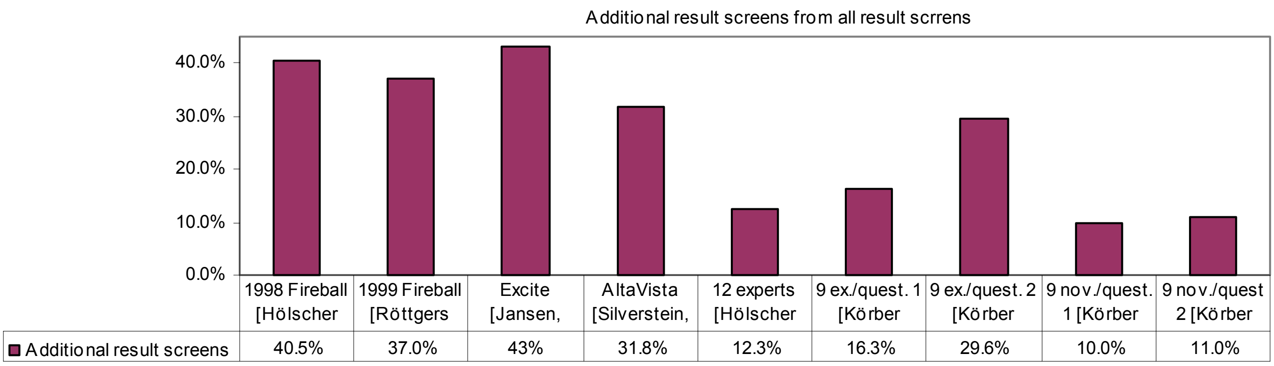


Figure 12: Percentage of additional result screens from all result screens

[Wang, Hawk, Tenopir 2000] developed a model of Web searching containing the three main components: user, interface, and Web space. Using this model, they performed a study with 24 graduate students in a library and information science program. The users had to perform two tasks in an unguided Web search. 14 participants were entry-level, and 10 advanced-level students or graduates. The authors did not find a significant relationship between search time and computer and search experience (IR and Web). In terms of effectiveness, the advanced-level participants performed slightly better than the entry-level ones (advanced/entry: correct answers 80% / 64%,

---

70 Percentages calculated by assuming a basis of 51,474 queries

71 "[…] *Anteil späterer Ergebnisseitenaufrufen an den Gesamtsuchseiten* […]" [Körber 2000]

72 All requests (100%) minus requests for the first result screen (59.51%)

incorrect answers 10% / 25%, no answer found 10% / 11%). But the authors summarize this part of their results as follows "*The advanced-level students who had completed the masters level information science core curriculum did not perform significantly better (more incorrect answers and less search time) than the entry-level students who just started the graduate program in information science.*" [73] Measuring the post-search confidence level regarding the correctness of the answers, 77.7% of the users who submitted incorrect answers expressed high or moderately high confidence on a four-point scale. In addition to the facts concerning efficiency, effectiveness, and confidence, the authors also present first results concerning cognitive style and affective states.

Another study assessing differences between four Web experts and four novices performing six search tasks, has been undertaken by [Weber, Groner 1999]. The experts' results were only slightly better than those of the novices. The main differences found occurred in terms of which strategies had been used.

[Lazonder, Biemans, Wopereis 2000] performed an experiment using 25 students with different levels of Web expertise, two different task types, and three levels of difficulty each. The users with higher Web expertise performed significantly better when time, performance success, efficiency, and effectiveness for site location tasks were evaluated. For tasks where the users had to locate information inside a Web site, both groups performed equally well. The authors had expected these results. They cite a number of studies from hypertext research that found little to no difference for browsing tasks between novice and expert users.

The above listed investigations, as well as a number of findings cited in earlier chapters, show that in a number of cases, but not always, there are differences in Web search strategies, behavior and success between different user groups. This fact will be taken into account later, when turning to the study which forms the basis of this thesis.

2.4. Summary of the chapter about Information Seeking

There are a number of different high-level models available which structure the information-seeking process into goals, tasks or strategies. Conclusions from the high level approaches were that: classical search is just one of the possible ways to fulfill an information need; goals and strategies are not static but may change during an information-seeking episode; not only the final result set is important but a number of items contributing to fulfilling the information need may also arise along the way; and strategies may depend on user experience. On a global level, in the remainder of this thesis a model from [Shneiderman 1998] will be used that differentiates the task actions: specific fact-finding, extended fact-finding, open-ended browsing, and exploration of availability.

When concentrating on searching as a possible strategy to fulfill an information need, the iterative nature of the search process has to be emphasized. Using the four-phase framework of information seeking developed by [Shneiderman, Byrd, Croft 1997] / [Shneiderman 1998] when talking about the visualization of search results, the most important phase is the "review of results". The preceding steps from the popup of an information need to the articulation of the problem are summarized as "formulation". This includes the selection of search sources and the transformation of the information need into a query which is understood by the system. The "action" as the next logical

[73] Must be „more correct answers"

step where the search is launched plays only a small role in the user-centered approach of this thesis. The "refinement" after the review of results is discussed in more depth in other papers.

Low-level tasks, goals, and interface actions were included to complete the overview, but will only play a subordinate role in what follows, because the discussion will take place in the area of high-level goals, tasks, and strategies or functions, phases, and steps of searching. An exception is a later chapter concerning visualization techniques.

When turning from theoretical constructs to real-world information regarding how people search the Web, some general trends found in large scale Web-studies are:

- The average search session contains roughly two queries
- The average length of a query is around two keywords, with an increasing tendency
- The majority of the queries do not contain Boolean operators or modifiers like "+" or "NEAR".
- In the majority of cases, people do not go beyond the first page of results
- Topics people are looking for come from all conceivable areas, they contain sexual topics and they seem to be influenced by trends

Controlled experiments sometimes show results differing from the large scale Web-studies based on the analysis of logfiles from search engines. Concentrating on differences between user groups, in a number of cases, but not always, there are differences in Web search strategies, behavior and success between different user groups.

3. Information Visualization

3.1. The ideas behind Information Visualization

As mentioned earlier in Chapter 1.2 the human perceptual system is highly adjusted to processing visual coded information very effectively. To exploit this, data visualization had been carried out for centuries [Tufte 1983]. In the last few decades such visualization using computers developed as an independent technical discipline within the area of Human Computer Interaction (HCI). A number of other disciplines are also contributing to the effective usage of visualization. Among them are Experimental or Cognitive Psychology and Human Factors Engineering. Especially since the 1980s, the concepts from data visualization have been transferred to many areas of application. The most important catch words are "Scientific Visualization" in the 80s and "Information Visualization" in the 90s. Scientific Visualization involves the use of visualization and animation for large data collections (e.g. concentration of ozone in the atmosphere) to exploit the human perceptual system and stimulate cognitive recognition of patterns in data [Robertson, Mackinlay, Card 1991]. The idea of Information Visualization is to transfer these methods to other forms of applications and data. Where Scientific Visualization had its main focus on physical phenomena, visualization is now used for diverse, often abstract types of information from large heterogeneous data sources (e.g. representations of large document sets). (Cf. [Robertson, Mackinlay, Card 1991], [Wise, Thomas, Pennock et al. 1995], [Card, Mackinlay, Shneiderman 1999])

"Data visualization" in general has two main facets: data presentation and data exploration. The focal point of consideration for data presentation is the communication of already known facts by suitable representational forms. The keyword for data exploration is to ease recognition and to support uncovering of unknown thematic connections by suitable visualizations, or to use a modern buzzword "Visual Data Mining". The transitions can be regarded as flowing. The realization gain, which is to be achieved with the help of the representation, is common to both facets. In the case of presentation, communication stands in the foreground, in the case of exploration it is the discovery. When concentrating on the visualization of search results, both aspects could theoretically be interesting. In the context of the search process itself however, where the user has an information need and the goal is to find relevant information, exploration is definitely the more important facet.

Systems combining the functionality of retrieval systems with the possibilities of information visualization systems are called *visual information seeking systems*. A large part of today's Web search results are documents. The following discussion about visualization will focus mainly on visual information-seeking systems for documents. Other data like multimedia or other use cases of visualization will only be mentioned briefly. An important aspect of visual information seeking systems is the possibility to visualize a great variety of document characteristics which they give, allowing the user to choose the most appropriate ones for his task.

The Information Visualization literature offers many ideas with regard how to visualizing data could help users to reach their goals. There are a considerable number of guidelines concerning when to use which visualization. Some of the findings are based on experiments and investigations. Despite the fact that the tradition of evaluations is quite long, there are many more ideas and theoretical thoughts about the value of visualization ideas than really evaluated results. Recently this trend has changed slightly, because information visualization systems and features are evaluated in more and more cases. The number of empirical studies is rapidly increasing [Chen, Yu

2000]. A number of factors influence the success of visualization for certain data in certain situations and for certain users, but for years it has been known that there is no "best" solution [Washburne 1927]. In the subsequent chapters, the field of Information Visualization will be dealt with in the following steps. First the reference model for visualization by [Card, Mackinlay, Shneiderman 1999] will be presented, to show the basic steps necessary to create successful visualizations. Then an overview covering the state-of-the-art of Information Visualization will be provided structured in metaphors, techniques, components, and systems. The chapter will focus on the visualization of abstract data. The special case of multiple coordinated views will be addressed in a separate sub-chapter. The chapter will close with a discussion of empirical evaluations of visualization ideas and a compilation of crucial factors for the usefulness of visualizations.

3.2. The reference model for visualization

In an article about user modeling for adaptive visualization systems in the context of Scientific Visualization [Domik, Gutkauf 1994] write: "*In order to generate the most meaningful picture for a specific instance, a careful mapping process from 'numbers to pictures' is necessary.*" The same careful mapping is necessary when mapping data to pictures, as with visual information seeking systems or Information Visualization in general. [Card, Mackinlay, Shneiderman 1999] developed a reference model for visualization describing this mapping process. They understand visualizations as adjustable mappings from data to visual form. The model is shown in Figure 13 and includes data transformations from the input in the form of raw data to data tables, visual mappings from data tables to visual structures, and view transformations from visual structures to the final views. The terms are explained in Table 13. The whole process is triggered by the users' task and manipulated by the human interaction.

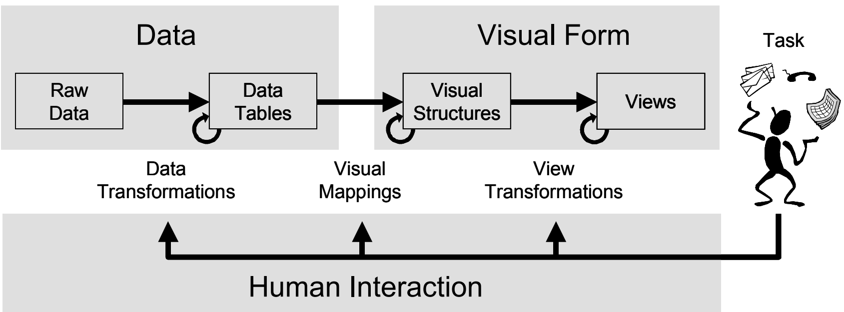


Figure 13: Slightly modified reference model for visualization adapted from [Card, Mackinlay, Shneiderman 1999] Figure 1.23

Term	Explanation
Raw data	Idiosyncratic formats
Data tables	Relations (cases by variables) + metadata
Visual structures	Spatial substrates + marks + graphical properties
Views	Graphical parameters (position, scaling, clipping, …)

Table 13: Explanation of terms from the reference model for visualization according to [Card, Mackinlay, Shneiderman 1999] Figure 1.23

The raw data as an input for a visualization can take a lot of forms, like for example a spreadsheet or data collected by a crawler from the World Wide Web. The raw data is usually transformed into one or more relations with variables and cases, because relations are structured and therefore easier to map to visual forms. [Card, Mackinlay, Shneiderman 1999] emphasize the distinction between different types of variables like the three basic types: Nominal (only = or ≠ to other values), Ordinal (obey a < relation) and Quantitative (can do arithmetic on them). Additionally, they make a distinction between important subtypes like spatial or geographical quantitative data, or quantitative or ordinal time. These differentiations are vital because they determine the type of axis to be used in a visual structure, or because the subtypes as important properties of the real world are normally associated with special visual conventions. The data transformation from raw data into data tables can lead to a loss or gain in information. It can range from a simple reduction of variables or cases, through statistical computations, to construction of derived values or derived structure. In Chapter 4.2.2, data transformations from raw data to data tables done in the INSYDER system will be shown.

The next step in the model, the visual mapping from data tables to visual structures, is one of the most critical ones in the whole process of visualization. "*Good mappings are difficult, ...*" [Card, Mackinlay, Shneiderman 1999]. A good mapping must preserve the data, it must be expressive, and it must be effective. Examples of what can be done wrong can be found in [Tufte 1983] or [Card, Mackinlay, Shneiderman 1999]. The route to the visualizations finally used in the INSYDER system was also not free from errors. Rules, guidelines, or examples to follow can be found in a large number of publications. It will blast this thesis, even when trying to discuss the most important ones. It is recommended that interested readers should have a look in publications like [Bertin 1977], [Bertin 1982], [Tufte 1983], [Mackinlay 1986], or [Card, Mackinlay, Shneiderman 1999]. When discussing the decisions made for the INSYDER system, a number of rules or guidelines will be mentioned which directly influenced the process.

The last step in the reference model for visualization is view transformations from visual structures to views. View transformations allow the user to get more information from a visualization than would be possible from a static presentation. The three most common view transformations listed by [Card, Mackinlay, Shneiderman 1999] are: location probes, viewpoint controls, and distortions. Location probes reveal additional data table information by using location in a visual structure. Viewpoint controls change the point of view by zooming, panning, or clipping. Overview + detail [Shneiderman 1996] is also a viewpoint control technique. By using distortion, overview + detail are combined in a single view with focus + context.

The human interaction that is also part of the model can work on all transformation and mapping steps described above. An example for a human interaction influencing the transformation from raw data to data tables is a selection of cases or variables, and an example for influencing the mapping from data tables to visual structures is a change of the diagram-type in a spreadsheet program. For influencing the transformation from visual structures to views, a good example would be a zooming operation in a diagram displayed.

The information visualization data state reference model of [Chi 2000] is very similar to the reference model for visualization by [Card, Mackinlay, Shneiderman 1999]. Chi presents a detailed analysis of a large number of visualization techniques using his version of the model. The similarities are not surprising because the data state model by [Chi, Riedl 1998] which is the basis for the new taxonomy from Chi was influenced by Card.

3.3. State of the Art: Visualization Ideas, Metaphors, Techniques, Components and Systems

The aim of this chapter is to give an impression of the great variety of ideas that have already been developed to map data tables on visual structures. When designing the INSYDER system, a scan of the available literature showed that it may be scientifically honorable to develop new ideas of how to visualize search results from the World Wide Web, but that there are already a great number of ideas available. Some of them already used for the visualization of Web search results, some are used for other IR-related systems, or others come from different application areas that could be potentially useful. Some of them have been evaluated, others of them not. Some of them proved useful, others of them not. The printouts of the figures found in the literature filled the walls of the researchers' office, and the question was, who was to structure the heap of visualization ideas?

[Shneiderman 1996] solved the problem by proposing a data type by task taxonomy (TTT) of information visualizations. The tasks are the ones shown in Table 6 on page 27. The data types are listed in Table 14. Shneiderman used the TTT in [Shneiderman 1998] to structure his overview of visualization ideas and systems. In [North 1997] and the On-line Library of Information Visualization Environments [OLIVE 1997], which also used the TTT, the data types were expanded and include an additional eighth type "workspace".

Data type	Examples
1-D Linear	Textual documents, program source code, alphabetical lists of names.
2-D Map	Planar or map data include geographic maps, floor plans, newspaper layouts.
3-D World	Real-world objects such as molecules, the human body, buildings
Temporal	Timelines used in medical records, project management, historical presentations. Special form of 1-D Linear.
Multi-Dimensional	Relational- and statistical-database contents.
Tree	Hierarchies and tree structures, with each item having a link to one parent item (except root)
Network	Network structures with items linked to an arbitrary number of other items

Table 14: Data types of the TTT data type by task taxonomy from [Shneiderman 1996], [Shneiderman 1998]

In other publications discussing a large number of visualization possibilities, the different techniques are grouped by a number of principles. [Card, Mackinlay, Shneiderman 1999] divide their overview into the chapters Space, Interaction, Focus + Context, Data Mapping: Document Visualization, "Infosphere, Workspace, Tools, Objects", Using Vision to Think. [Chi 2000] organized his discussion of visualization techniques into the following groups: "*Some example Scientific Visualizations*", "*Geographical-based Info Visualization*", "*2D*", "*Multi-dimensional Plots*", "*Information Landscapes and Spaces*", "*Trees*", "*Network*", "*Text*", "*Web Visualization*", and "*Visualization Spreadsheets*". In a general examination of the visualization of search results in document retrieval systems, [Zamir 1998] used a classification shown in Figure 14. The classification focuses on post-retrieval document visualization techniques.

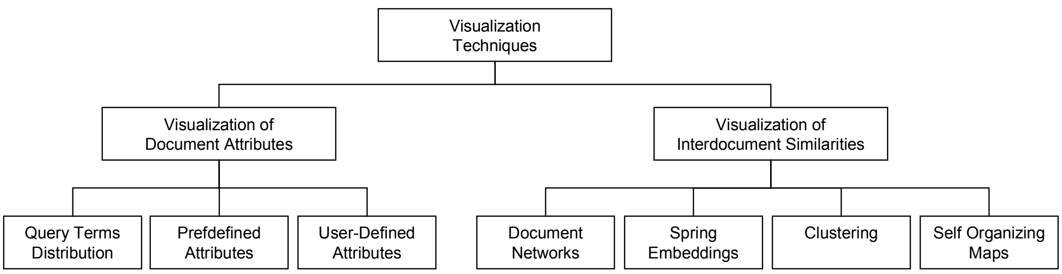


Figure 14: Classification of post-retrieval document visualization techniques according to [Zamir 1998] Fig. 1

The lesson learned from these examples: it is difficult to structure overviews about visualization ideas in a one-dimensional system or a hierarchy. Application domain focused classifications seem to be easier. The classification system used in this thesis to give an overview covering visualization ideas will, like others, also be multidimensional and contains the following dimensions: Metaphors, Techniques, Components, and Systems. Multiple Coordinated Views will be discussed separately in a later chapter. The dimensions are shown in Figure 15. “System” here means a complete software system for visualization, or a system which incorporates a visualization part. Components are parts of a system using their own visual structures. Techniques can be the basis of or work inside a component or between components. Metaphors can stand behind a component or behind a complete system.

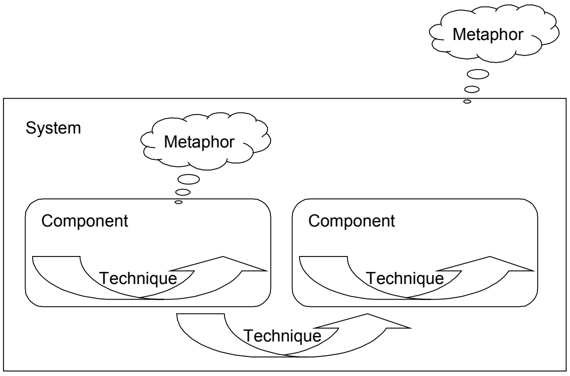


Figure 15: Dimensions of the classification scheme for visualizations

The dimension of the components will be structured in a way comparable with a merger of a simplified and slightly extended visualization classification by [Zamir 1998] and the four-phase framework of information seeking by [Shneiderman, Byrd, Croft 1997] / [Shneiderman 1998]. It will not only contain visualizations used in (Web-) document retrieval systems, but also visualizations that come from other systems, but may be suitable for such a use case.

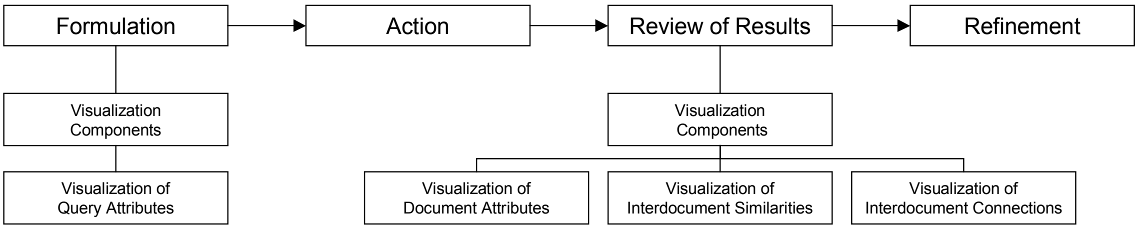


Figure 16: Classification of visualization components

A number of systems, components, techniques, or metaphors will be mentioned in more than one of the following chapters. It is intrinsic to the nature of the approach to use a multidimensional classification for discussing the state-of-the-art of Information Visualization for search results. It is even more the case because it is not always really possible to distinguish between systems, components, and subcomponents. Organizing the overview just along one dimension will lead to problems too. Not all of the items described have characteristics in all dimensions. Taking the metaphor dimension, not all systems or components make use of metaphors. On the contrary, there are a number of authors explicitly not using metaphors. On the other hand, a number of systems use multiple metaphors. In terms of the techniques dimension, most of the systems or components use more than one technique and will therefore be mentioned more than one time. However, with the systems dimension, not all of the components described in the literature are part of a system. Some are just isolated ideas. On the other hand many of the systems use different components for different use cases. To use the component dimension is probably the best approach for structuring the overview. In addition to its relatively generic level, it can be sub-structured by usage scenarios based on the four-phase framework, like done above. Therefore the chapter about components and usage will serve as the backbone of the overview. The other chapters about metaphors, techniques, systems, and multiple coordinated views provide other views on the state-of-the-art. The goal of these additional chapters is to broaden the horizon and emphasize the fact that there are additional possibilities for structuring the scene. Furthermore, they serve as focus points for aspects which would otherwise be underrepresented by just being spread over the components and usage chapter. The discussion will start with metaphors, because they are a good lead into the scene. Intrinsic to the nature of metaphors is that in most cases, the ideas behind them are easily understandable without in-depth explanation, examples, or figures. The chapter about techniques sharpens the eye for the following discussion of the components and their usage. Additionally this view of the scene is connected tightly to the framework of visualization. After the detailed discussions of components and their usage, the chapter about systems serves as a sort of reference for the items discussed in the chapters before it. The last chapter about multiple coordinated views prepares the field for the introduction of the INSYDER system and its evaluation described in Chapter 0.

3.3.1. Metaphors

When using metaphors in software system design, a central goal is often to control the complexity of the user interface by exploiting specific prior knowledge that users have of other domains [Carroll, Mack, Kellogg 1988]. Taking the reference model for visualization, as in Figure 17, the starting point of a metaphor is the visual mapping from data tables to visual structures.

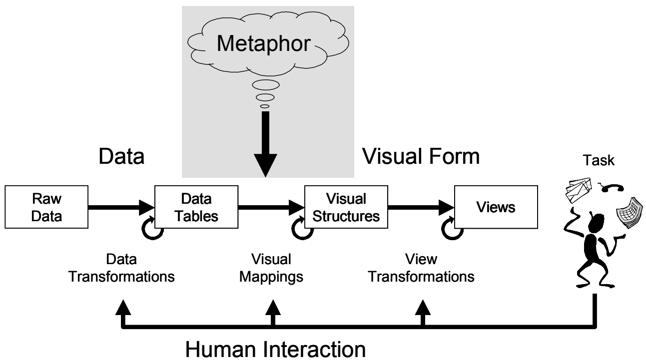


Figure 17: Starting point of a metaphor in the reference model of visualization

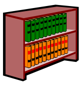
The choice of a metaphor will heavily influence or be restricted by this mapping process. The visual mapping of a data structure to a visual structure is the core of the reference model for visualization [Card, Mackinlay, Shneiderman 1999]. This core will be illuminated in this chapter by introducing a number of metaphors already used for the visualization of queries, search results, or browsing.

A lot of literature is to be found regarding the pros und cons of metaphors. Papers like [Carroll, Mack, Kellogg 1988] give a lot of examples and discuss metaphors from different angles of view. This includes aspects of composite metaphors, possible benefits of metaphor mismatches, the fact that is likely that people generate metaphoric comparisons on their own - whether or not explicit metaphors are designed for a user interface. They discuss operational, structural, and pragmatic analysis of metaphors and show the necessity for the user not only to have metaphors but a coherent and complete mental model of a software system. [Stubblefield 1998] gives a very vivid illustration the role of metaphors not only for the users of a system but also for the developers. For an interesting discussion about the disadvantages of metaphors, also including aspects of design, see [Bederson, Hollan 1994], [Bederson, Hollan, Perlin et al. 1996]. They propose using physics-based design strategies instead of metaphors.

During the design phase of the INSYDER system and its visualizations, one part of the work was to investigate which metaphors had been used for other systems with comparable functionality. The usage of certain metaphors would also be a candidate for structuring an overview covering visualization ideas. But metaphors can stand behind a component or behind a whole system. Their match can be more or less complete. Composite metaphors could be used. Metaphors are sometimes easily comparable, on the other hand a certain metaphor can be used for completely different target domains or tasks. During the design of the INSYDER system, the results from searching metaphors served more as a pool of ideas, than for the classification of visualizations in general. In the INSYDER system itself metaphors are used in a number of ways, like for example presenting predefined or stored searches, watches and news in the form of a file-browser, or using visualizations with similarities to business-graphics, after discussing the target user group of the system and their possible pre-experiences. This chapter about metaphors will be restricted to a brief overview of metaphors used in systems with visualizations of queries, browsing or search results. Metaphors found include: Book, Bookshelf, Newspaper, City, Landscape, Rooms, Building, Tower, Guided Tour, Lens, Butterfly, Pile, Universe / Galaxy / Starfield, Magnet, Sculpture, Television, Wall, Aquarium, and flowing Water.


A **book** metaphor has been used in several systems. Examples of these include SuperBook (showing one document as one book), BOOK HOUSE (showing the metadata of a book as a book), the WebBook (showing groups of Web pages as books), and the libViewer (showing n documents or n Web pages as n books). SuperBook / MiteyBook [Egan, Remde, Gomez et al. 1989] is more a hypertext browsing-system than a retrieval system, but implemented a number of good ideas (not really visualizations) using a book metaphor. An ASCII-text with heading markers or in a standard text markup language is preprocessed and displayed in book format with table of content, word lookup, and text display. A number of features are available, such as string search, and highlighting of query terms in text. [Pejtersen 1989] used an image of an "open book" in the implementation of the BOOK HOUSE system to show descriptions of retrieved documents one at a time. The BOOK HOUSE itself was an electronic DOS/GEM replica of a real library with a library building, rooms, or people. Retrieval was depicted using icons symbolizing the different dimensions of the classification system. A globe represented the geographic setting of the book, a clock the time dimension, or a theatre mask the emotional experience provided by a book. Icons are also used for a number of other functions in the BOOK HOUSE. The WebBook [Card, Robertson, York 1996] allows users to group related Web pages into a higher aggregated entity, and to manipulate them as a unit. WebBooks themselves are used in an information Workspace called Web Forager. The whole system is implemented in the framework of the Information Visualizer system [Robertson, Card, Mackinlay 1993]. The WebBook preloads a collection of Web pages and shows them in a 3D simulation of a real book. A number of HTML-properties are adapted to the usage of the pages in a collection. Links, for example, are color-coded depending on if they point to pages inside or outside the virtual book. The WebBook supports a number of features associated with real-world books, like for example ruffling or insertation of bookmarks. Other features have no counterparts in the real world, like the possibility to explode the book out so that all pages are available simultaneously and can be viewed using a fisheye-technique called Document Lens

[Robertson, Mackinlay 1993]. Animation plays an important role in the implementation. In the libViewer applet [Rauber, Bina 1999], which is part of the SOMLib project, search results from a retrieval system are shown as 3D-books by mapping metadata of documents to attributes of real-world books. Examples can be seen on page 87 and page 102. The applet has also been used to show Web search results [Rauber, Bina 2000]. [Card, Robertson, York 1996] list a number of other systems that also use the book metaphor.


Among other systems the already introduced Web Forager system and libViewer applet both use in addition to the book metaphor also a **bookshelf** metaphor. The Web Forager allows the user to place the WebBooks on a virtual bookshelf as a tertiary storage area, in addition to an immediate storage area to work on and a virtual desk as intermediate storage [Robertson, Card, Mackinlay 1993]. The libViewer uses a virtual bookshelf to display the book-representations of documents in an ordered or grouped way. In a simple mode the “books” are ordered in the bookshelf by a dimension of the available metadata, like size or relevance. In an advanced mode, the authors use an unsupervised neural network in the form of a self-organizing map [Kohonen 1998] to cluster documents dealing with similar topics. Every single cluster is then displayed as a single shelf in the bookshelf labeled by using a so-called LabelSOM technique. [Baeza-Yates 1996] also proposes the usage of a bookshelf metaphor in a way comparable to the “simple” mode of the libViewer. He calls it “library” or “bookpile” depending on the orientation. Document attributes like relevance, size, or age can be mapped by the user to graphical properties like position, color, width, or height. The “library”-idea has like the libViewer, been implemented in a Java-Applet [Alonso, Baeza-Yates 1998]. There the library-view is also called horizontal bookpile.


In the VOIR (Visualization Of Information Retrieval) system [Golovchinsky 1997][74] uses a **newspaper** metaphor for the visualization of search results, respectively the navigation in a query-mediated hypertext. Newspaper metaphors are quite frequent in the Web. Examples are electronic newspapers or personalized electronic newspapers[75]. The special point of the VOIR system is the usage of a newspaper metaphor for the visualization of texts that have in general nothing to do with news as content. The idea is to use the metaphor of a newspaper for organizing loosely related units of internally coherent text, retrieved by a number of different mechanisms. Visual cues from newspapers, like space used to display a certain document, are applied, for example, to mirror the relevance of the text in a current situation[76].

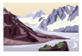

[Dieberger 1994] proposed the usage of a **city** metaphor as a conceptual spatial user interface metaphor for large information spaces. [Dieberger, Frank 1998] contains an overview covering other use cases of the city metaphor. In their Information City approach the authors describe an ontology of spaces and connections to be used when talking about systems of spatial metaphors and how they interrelate. The ontology includes containers, landmarks, and paths in form of districts, sub-districts, buildings, rooms, doors, taxis, subways, and others.

---

[74] See also: [Golovchinsky 1997a], [Golovchinsky 1997b], and [Golovchinsky, Chignell 1997]

[75] “The Kraktatoa Chronicle” [Kamba, Bharat, Albers 1995] seems to the first one using besides the news as content also a newspaper-like layout.

[76] The usage of space to reflect relevance seems to be used not fully consistent. The author describes that every page displayed shows a fixed number of eight articles. More than eight retrieved articles are displayed on subsequent pages. Therefore the usage of space can only reflect the relative relevance of an article in a group of eight, but not the overall relevance. The ninth article in an overall relevance-ranking list will get more space than the eighth one.

 A **landscape** metaphor has been used in a number of systems including the Harmony Hyper-G / Hyper View browser, ThemeScapes in the SPRIRE system, and Landscapes in Vineta or Bead. [Andrews 1995] describes the Harmony VRweb 3D scene viewer with handcrafted three-dimensional landscapes (e.g. a plan of the city center of Graz containing hyperlinks to sightseeing information), or automatically created three-dimensional landscapes depending on user navigation steps or searches in the hypertext environment. Providing an additional 2D-map overview helps to keep orientation in the three-dimensional landscape. ThemeScapes [Wise, Thomas, Pennock et al. 1995] was one of the views developed in the MVAB (Multidimensional Visualization and Advanced Browsing project) / SPIRE (Spatial Paradigm for Information Retrieval and Exploration) project. ThemeScapes are abstract, three-dimensional landscapes of information constructed by automatically analyzing the thematic content expressed in the documents of a collection. The second visualization in SPIRE is the Galaxies view. The visualizations of the German prototype Vineta were described in an earlier paper [Krohn 1995] as spheres in 3D space. Later Vineta also used a landscape and a galaxy view[77] [Elzer, Krohn 1997]. Whereas the automatically constructed landscapes in the Harmony VRweb 3D scene viewer looked like pedestals and boxes connected by wires[78], ThemeScapes provoke the impression of mountains or natural terrain. An example of a ThemeScape can be found on page 99. [Chalmers 1993] also used a technique to present high dimensional data in low dimensional space in the Bead system. The system calculates similarities between pairs of documents. In the visualization the documents are spread over a landscape like trees or little pyramids. Documents with keyword-matches are displayed in another color. In later versions the landscape looked more like cubes and wires [Chalmers 1995] and had additional colored districts [Chalmers 1996]. There seem to be also more labels. As opposed to the Harmony browser where wires symbolize links between documents, the wires in Bead seem to visualize other connections. [Bekavac 1999] used a landscape to symbolize the geographical frame of an electronic mall in the VR-emb[79] prototype. The navigation in the electronic mall itself was done inside a tower (See below). In front of the tower some road signs allowed navigation to cities and institutions in the geographical area of the electronic mall. The landscape around the tower also included cars and a helicopter in front of the tower for navigation to other malls or places (not implemented).

 [Henderson, Card 1986] used the **rooms** metaphor for a technique that virtually enlarged the available screen space by allowing the user to organize, save, and recall window positions and other features as working sets for later reuse. Their Rooms system included a lot of additional ideas and metaphors like an overview to switch between rooms, “pockets” for carrying windows to every room, or “baggage” to carry windows to another room. They also list a number of previous usages of the room metaphor. Their usage was purely desktop organization. The logic of the original 2D-version was later in the 3D/Rooms of the Information Visualizer extended to a 3D-version, whilst keeping the original controls like doors for “walking”[80] from one room to another and add-

---

[77] *„Um die Brauchbarkeit und Akzeptanz verschiedener Darstellungsformen besser testen zu können, wurden zwei Modelle realisiert: „Die ‚Galaxie' (Fig.6 und 7) und die ‚Landschaft' (Fig. 5).*“ [Elzer, Krohn 1997]

[78] Comparable to the pedestals, boxes, wires-look of the FSN (pronounced fusion) 3D File System Navigator for IRIX developed by [Tesler, Strasnick 1992]. The FSN has also a similar 2D-map overview window.

[79] Virtual Reality – electronic mall bodensee (Lake Constance, Germany – Switzerland – Austria)

[80] Walking is an additional metaphor used by [Robertson, Card, Mackinlay 1993]

ing additional functions like zooming [Robertson, Card, Mackinlay 1993]. In the Information Visualizer, the idea of rooms was then combined with techniques for browsing or searching. The already-mentioned BOOK HOUSE [Pejtersen 1989] used rooms to structure the functions of the retrieval engine in areas like search functions for children's books, search functions for adults books, or mixed. The first steps of the retrieval process are structured as a route from room to room. The input of the query itself and the display of the retrieval results used other metaphors. The Information City ontology of [Dieberger, Frank 1998] also included rooms.

 A **building** metaphor is also often used in visualizations of search results or for browsing, mostly in conjunction with other metaphors. In [Dieberger, Frank 1998], buildings are part of the Information City ontology, and contain rooms, doors, or windows. In the Information Visualizer [Robertson, Card, Mackinlay 1993] a spatial structure of a building is used as a structural browser for people – but seems to be independent from the also used 3D/Rooms. In the BOOK HOUSE [Pejtersen 1989] the building is the overall metaphor of the system and the framework for the integrated rooms.

 A **tower plus elevator** metaphor, as a special form of building, is used by [Bekavac 1999] in the VR-emb prototype mentioned above. After entering the tower, the user found himself in an elevator with the possibility to navigate inside the mall by using the elevator controls. Participants of the electronic mall are virtually located on different floors of the tower.

 [Guinan, Smeaton 1992] combined the **guided tour** metaphor from [Hammond, Allinson 1987] with information retrieval techniques to create dynamically guided tours in direct response to a query of a user. The system was restricted to a single hypertext about databases with special link categories indicating different types of relations. After a three month test with 125 users, the authors claim that their solution overcomes three problems associated with using hypertext: getting lost, finding information, and logical sequence of nodes. The main difference to most other guided tours is the dynamic creation of the guided tour in response to a query.

 As mentioned above, the Information Visualizer also included a component called Document Lens [Robertson, Mackinlay 1993], using a **lens** metaphor for a graphical fisheye-view of the pages of a book. Figure 41 on page 77 shows the principle of this component. The lens metaphor was also used for the see-through tools by [Bier, Stone, Fishkin et al. 1994], including magic lenses as movable filters for the change of the view of items under the lens [Stone, Fishkin, Bier 1994], or to formulate database queries [Fishkin, Stone 1995]. Figure 27 on page 68 shows an example of movable filters. The Table Lens [Rao, Card 1994] as a tool for viewing tables or results lists in tabular form, also uses the lens metaphor. While all these components use the lens metaphor, there are some differences between them. The Table Lens uses the metaphor in a more abstract form. There is not really a "lens" identifiable. See Figure 95 on page 110 for an example. Some rows of the table can be viewed with a presentation of more details. Whereas the see-through-tool examples provided by [Bier, Stone, Fishkin et al. 1994], [Stone, Fishkin, Bier 1994], [Fishkin, Stone 1995] only influence the items under the lens, the Document Lens distorts all pages of the document displayed. A lens metaphor is also used by [Resnick, Iacovou, Sucak et al. 1994] for the GroupLens system. GroupLens is a tool for collaborative filtering of postings from

newsgroups. The system is based on user ratings, and the comparison of ratings and profiles. The term "lens" for the filtering mechanism is really used metaphorically.

A **butterfly** metaphor is used by [Mackinlay, Rao, Card 1995] in the Butterfly part of the Information Visualizer project. The system, targeted in general to solve a Fast User Interface / Slow Multiple Repository problem, is used to support asynchronous querying of three DIALOG databases: the Science Citation Index, the Social-Science Citation Index, and the IEEE Inspec database. The Butterfly visualization shows references of an article as "veins" of a stylized left wing of a butterfly, and the article's citers located in the citation databases as veins of the right wing.

A **pile** metaphor is used by a number of systems to visualize search results. [Rose, Mander, Oren et al. 1993] used the metaphor "a pile of documents" presented in [Mander, Salomon, Wong 1992] for a prototype implementation of a tool to support casual organization of information on a Macintosh. Besides possibilities for manual organization of documents in piles[81], the system also included mechanisms for automatic filing and indexing of documents. They used a variant of the popular tf*idf algorithm to rank documents and additional mechanisms for extracting terms to describe documents and piles. The prototype supported functions like flipping step by step through the documents, ordering, or automatic subpiling of piles. The icons of the documents and piles used the well-known icon-style of the Macintosh. [Brown, Shillner 1995] introduced DeckScape as a Web browser based on a "deck" metaphor. Web documents are represented as stapled simple rectangles containing the titles of the documents. The system supports mechanisms like inserting documents into a deck when returning to a previous seen document, and following a new link from there. Other features include "Expand One Level" as a command, which follows all links of a particular page, and returns all resulting pages in a new deck. As mentioned above, [Baeza-Yates 1996] / [Alonso, Baeza-Yates 1998] also called their "library" view "horizontal bookpile" when oriented horizontally, or just "bookpile" when oriented vertically. The Butterfly [Mackinlay, Rao, Card 1995] part of the Information Visualizer project also uses a pile metaphor in the form of a stylized pile below the butterfly to stack articles the user has selected.

A **galaxy** or **starfield** or **universe** metaphor has been used in a number of systems including Galaxies in the SPIRE system, and Vineta. [Wise, Thomas, Pennock et al. 1995] described the Galaxies used in SPIRE as 2D scatterplots of 'docupoints' appearing in the way that stars do in the night sky. They show cluster and document interrelatedness by reducing a high dimensional representation into two dimensions. Clusters are annotated with key terms. The more similar two documents or clusters are, the nearer to each other they appear in the visualization. The component is enriched by additional features like a "temporal-slicer" to divide the document collection into temporal units. The galaxies in Vineta [Elzer, Krohn 1997] were implemented in 3D. The usage of the metaphor here is more abstract than in the SPIRE Galaxies. The main concepts are the same.

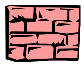
A **magnet** metaphor is used by [Morse, Lewis 1997] in the WebVIBE to symbolize the reference points / Points of Interest (POIs) attracting documents in a virtual 2D-Document space.

---

[81] [Robertson, Czerwinski, Larson et al. 1998] also implemented a prototype where they allowed users to organize documents in piles. The usage there was for bookmarks. They did not talk from a "pile" metaphor, but used instead the term "Data Mountain", because the users had a virtual mountain with a planar surface in form of a plane tilted at 65 degrees to put down and organize the document thumbnails.

 In the Information Visualizer [Robertson, Card, Mackinlay 1993] also used a **sculpture** metaphor for a visualization called Data Sculpture, visualizing in the example 65,000 sampling points from a data set like a sculpture in a museum. The visualization is a 3-D surface plot offering the possibility to fly around the object. The visualization shown in their Figure 6 has more similarities to a landscape than a sculpture. Interestingly, in the system overview displayed in their Figure 1 the room with the Data Sculpture is labeled DataMap.

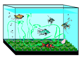
 Influenced by the FRIEND21[82] project [Nonogaki, Ueda 1991], a **television** metaphor has been used in the WebStage prototype by [Yamaguchi, Hosomi, Miyashita 1997]. The aim of the system is to reduce user operations necessary to access the Web by presenting Web page information in a style comparable to television programs. This includes media transformations in a form where, for example, titles and captions are displayed on the screen using a large font, whereas other text strings are spoken by a text-to-speech-synthesizer. Images are also presented on the screen. The presentation can be accompanied by background music or sound effects chosen by the system to create an appropriate atmosphere for certain information types[83]. Retrieval or selection of Web pages to be displayed is also implemented in a TV-like style by organizing, for example, URLs by time slots over the day and automatically starting a currently scheduled presentation when starting the system. Clusters of URLs to be displayed on a channel-panel can be retrieved by using other Web search engines or directory services.

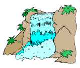
 The **wall** metaphor is used in the form of the “Perspective Wall” in the Information Visualizer environment by [Mackinlay, Robertson, Card 1991] to solve two principle problems of visualizations of large amounts of linear structured data: the large amount of information that must be displayed, and the difficulty of accommodating the extreme aspect ration of a linear structure on the screen [Robertson, Card, Mackinlay 1993]. A detailed and a contextual view are integrated in one visualization. In the implementation, the horizontal dimension of the wall is used for time, and the vertical is used to visualize layering in an information space. Examples are visualizations of files with the modification date in the horizontal axis and the file type in the vertical axis. The Perspective Wall is a variant of the one-dimensional Bifocal Display introduced by [Spence, Apperley 1982]. The Bifocal display does not use the wall metaphor, and has a constant demagnification rate for the regions out of focus, whereas the Perspective Wall has an increasing rate for demagnification. On page 109, figures of both techniques are shown. [Mackinlay, Robertson, Card 1991] use a number of other metaphors to explain the functionality of the Perspective Wall, namely sheets in a player piano to explain navigation on the wall, and a sheet of rubber to explain changes of the ratio between detailed and contextual information. The metaphor of a “rubber sheet” is also used by other authors to explain the functionality of their system. Examples are [Jog, Shneiderman 1995] for the Filmfinder (“rubber mat”, “rubber carpet”) or [Bederson, Hollan, Perlin et al. 1996] for Pad++ (“rubber sheet”). [Leung, Apperley 1994] use the “rubber sheet” metaphor to explain distortion-oriented presentation techniques in general, and list some additional papers using it.

---

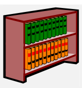
[82] FRIEND21 = Future Personalized Information Environment Development project, initiated in 1988 by the Japanese Ministry of International Trade and Industry

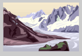
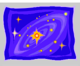
[83] [Bekavac 1999] described also the idea of using background music. In the case of the VR-emb different types of background music should support orientation in an electronic mall.

 [Bryan, Gershman 2000] used an **aquarium** metaphor for the interface of a large online store. The interface supports a mixture between browsing and searching they called "opportunistic exploration". Their motivation had been deficiencies in the usability of current online stores for shopping, they commented as "*This is not shopping; this is information retrieval and order entry.*" The new interface shows products in front of a blue aquarium-like background moving slowly, almost randomly like fish. The selection can be changed by relevance feedback or keyword search. Without user interaction the content is gradually changed automatically to show a diversity of products. Common operations from hypertext browsers like bookmarks, forward or back are also supported. The approach shows some interesting alternatives for difficult definable queries in Web search or classical IR context.

 The last metaphor to be introduced here is **water flowing** through a series of pipes and filters. The concept, also known as Filter/Flow, is used for query formulation by [Shneiderman 1991] / [Young, Shneiderman 1993] to overcome known problems with the formulation of Boolean queries. The filters only let through the appropriate documents, and the pipe layout determines whether the relationship is an "AND" or an "OR". The approach also allows the saving and reusing of combinations of filters and pipes as clusters with a reduced visualization of the cluster for reuse in other queries. Figure 26 on page 68 shows the principle of Filter/Flow.

The variety of metaphors found in the literature and introduced above illustrates the variety of possibilities for visualization of queries, search results, or browsing. It has been shown that a metaphor can stand behind a complete system, behind a component or even behind a part of a component. Table 15 summarizes the overview of metaphors. As we will see later in the chapter about Components, there are also a great number of other visualization possibilities without the explicit use of metaphors. Before discussing these concrete possibilities, in the next chapter there will be an excursus about the more general concept of techniques used in visualizations.

Metaphor		**Literature**	**Component**	**System / Project**
	Book	[Egan, Remde, Gomez et al. 1989]		SuperBook, MiteyBook
		[Pejtersen 1989]		BOOK HOUSE
		[Robertson, Card, Mackinlay 1993], [Card, Robertson, York 1996]	WebBook	Information Visualizer
		[Rauber, Bina 1999], [Rauber, Bina 2000]	libViewer	SOMLib
	Bookshelf	[Robertson, Card, Mackinlay 1993]	Web Forager	Information Visualizer
		[Baeza-Yates 1996], [Alonso, Baeza-Yates 1998]	"library", "horizontal book-pile"	
		[Rauber, Bina 1999], [Rauber, Bina 2000]	libViewer	SOMLib
	Newspaper	[Golovchinsky 1997], [Golovchinsky 1997a], [Golovchinsky 1997b], [Golovchinsky, Chignell 1997]		VOIR
	City	[Dieberger 1994], [Dieberger, Frank 1998]		Information City

Metaphor		Literature	Component	System / Project
	Landscape	[Andrews 1995]	Harmony VRweb 3D scene viewer	Harmony Hyper-G / Hyper View
		[Wise, Thomas, Pennock et al. 1995]	ThemeScapes	MVAB / SPIRE
		[Krohn 1995], [Elzer, Krohn 1997]		Vineta
		[Chalmers 1993], [Chalmers 1995], [Chalmers 1996]		Bead
		[Bekavac 1999]		VR-emb
	Rooms	[Henderson, Card 1986]	Rooms	
		[Robertson, Card, Mackinlay 1993]	3D/Rooms	Information Visualizer
		[Pejtersen 1989]		BOOK HOUSE
		[Dieberger 1994], [Dieberger, Frank 1998]		Information City
	Building	[Pejtersen 1989]		BOOK HOUSE
		[Robertson, Card, Mackinlay 1993]		Information Visualizer
		[Dieberger 1994], [Dieberger, Frank 1998]		Information City
	Tower plus Elevator	[Bekavac 1999]		VR-emb
	Guided Tour	[Hammond, Allinson 1987]		
		[Guinan, Smeaton 1992]		
	Lens	[Robertson, Mackinlay 1993]	Document Lens	Information Visualizer
		[Rao, Card 1994]	Table Lens	
		[Bier, Stone, Fishkin et al. 1994], [Stone, Fishkin, Bier 1994], [Fishkin, Stone 1995]	See-through tools, Magic Lenses, Movable Filters	Toolglass, MagicLens
		[Resnick, Iacovou, Sucak et al. 1994]		GroupLens
	Butterfly	[Mackinlay, Rao, Card 1995]	Butterfly	Information Visualizer (Butterfly)
	Pile	[Mander, Salomon, Wong 1992], [Rose, Mander, Oren et al. 1993]		Macintosh
		[Brown, Shillner 1995]	"deck"	DeckScape
		[Baeza-Yates 1996] / [Alonso, Baeza-Yates 1998]	"bookpile"	
		[Mackinlay, Rao, Card 1995]	"pile"	Information Visualizer (Butterfly)
	Galaxy	[Wise, Thomas, Pennock et al. 1995]	Galaxies	MVAB / SPIRE
		[Elzer, Krohn 1997]		Vineta
	Magnet	[Morse, Lewis 1997]	2D Document Space with Reference Points / Points of Interest (POIs)	WebVIBE
	Sculpture	[Robertson, Card, Mackinlay 1993]	Data Sculpture	Information Visualizer

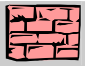
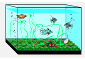
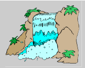
Metaphor		Literature	Component	System / Project
	Television	[Nonogaki, Ueda 1991]		FRIEND21
		[Yamaguchi, Hosomi, Miyashita 1997]		WebStage
	Wall	[Mackinlay, Robertson, Card 1991], [Robertson, Card, Mackinlay 1993]	Perspective Wall	Information Visualizer
	Aquarium	[Bryan, Gershman 2000]		"Online Store"
	Flowing Water	[Shneiderman 1991], [Young, Shneiderman 1993]	Filter/Flow	

Table 15: Metaphors used for the visualization of queries, search results or browsing

3.3.2. Techniques

Besides icons and color highlighting, [Hearst 1999] lists the following main information visualization techniques: brushing and linking, panning and zooming, focus-plus-context, magic lenses, animation, and as an additional combination overview-plus-detail. These different techniques could also be a skeleton to structure an overview covering visualizations. As with the metaphors, the classification will here be used to give an impression of the different techniques which have already been implemented in visualization components for queries, search results, or browsing. [Card, Mackinlay, Shneiderman 1999] use a classification of interaction techniques[84] that has on the one hand some similarities with the information visualization techniques classified by [Hearst 1999]. One the other hand, there are also divergences. Subsequent Hearst's version is used due to its clarity and simplicity.

3.3.2.1. Brushing and linking

The term "brushing and linking" describes a connection between two or more views of the same data. A selection or highlighting of the representation in one view affects the representation in other views as well[85]. A little extension, shown in Figure 18, of the reference model for visualization shown in Figure 13 on page 47 helps to understand this process. The raw data is not only mapped to one view at a time, but to several views.

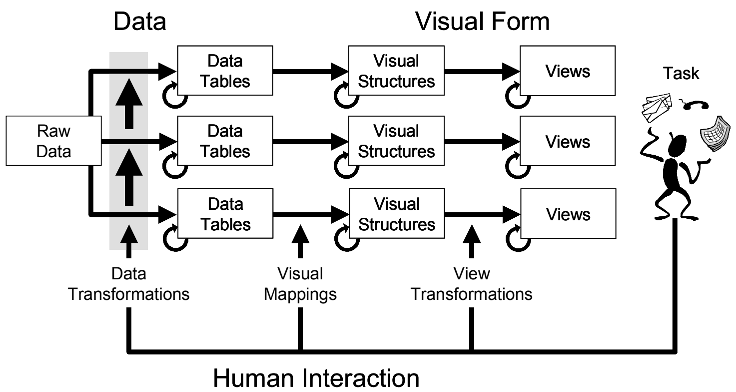


Figure 18: Expanded reference model for visualization: Brushing

Highlighting when brushing can occur in a number of forms. Examples include using a different color, font, background, or symbol, and adding additional labels for highlighted items [Eick, Wills

[84] [Card, Mackinlay, Shneiderman 1999], page 233, Table 3.2

[85] Which is not outside the scope of the original one row model from [Card, Mackinlay, Shneiderman 1999], but seems to be a bit clearer when presented as done in Figure 18 with some parallel rows.

1995]. In the sense of the reference model for visualization, this could be seen as a change of the visual mappings. The example from [Hearst 1999] for brushing and linking, assigning a color to a bar in a histogram, causing titles in a list display to be presented in the same color, seems to support this classification. On the other hand, [Card, Mackinlay, Shneiderman 1999] in their Table 3.2 categorize brushing clearly as a technique which modifies the data transformation. This is emphasized by their example for brushing, "*highlighting a case from the Data Table in one view selects the same case in the other views*". A concept to understand this could be that the visual mapping for highlighted items is already predefined, even when at the beginning of a session no items have been highlighted. When the user selects the items, the visual mapping itself remains unchanged. The selection is mirrored in the data tables and causes a number of so far unselected cases to change from "unselected" to "selected". The changed cases lead to changed visualizations. Examining some classical literature regarding brushing such as [Tweedie, Spence, Williams et al. 1994] or [Eick, Wills 1995], the examples relatively clearly support the classification as "data transformation". Hearst's example can be understood as a two step process: in a first step selecting the items to be highlighted, causing the data transformation to be changed, and then in a second step also changing the visual mapping for highlighted items by assigning the color red to this attribute. Brushing and linking will, in this thesis, be understood as a linked change of data transformations. In Chapter 3.4, linked changes of visual mappings or view transformations will be discussed in the broader context of Multiple Coordinated Views. An example for a system implementing brushing and linking for the visualization of search results is the INQUERY-based 3D-visualization system by [Swan, Allan 1996] / [Allan, Leouski, Swan 1997]. Marking a document as "relevant" in a ranked list or a text viewer by clicking on a green check box[86] also turns the color of the corresponding icon in the three-dimensional Document Map to green. Ranked list and text viewer are linked in the same way. Other visualizations of the system, such as the Concept List and the Concept Map are linked by selection. Another example for a system implementing a brushing technique is the Navigational View Builder [Mukherjea, Foley, Hudson 1995]. The system is able to show multiple hierarchies for hypertexts and Web search results. A node selected by a user in one view is also highlighted in the other views.

3.3.2.2. Panning and zooming

A typical technique influencing the view transformation from visual structure to views is panning and zooming. Changing the viewpoint of the users alters the portion of the displayed part of the visual structures.

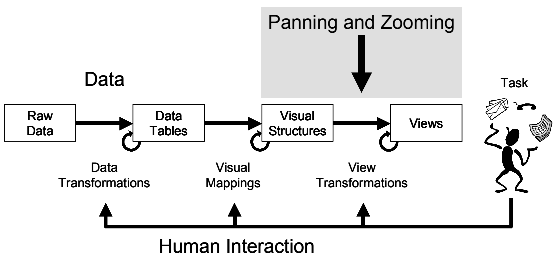


Figure 19: Reference model for visualization: Panning and zooming

[86] [Allan, Leouski, Swan 1997] call it a check box. Looking on Figure 1 and Figure 2 in [Swan, Allan 1996] it is part of a radio-button.

This can be done without changing the zoom factor by moving sideways or by changing the zoom factor. [Hearst 1999] uses the metaphor of a movie camera for explanation: "*scan sideways across a scene (panning) or move in for a closeup or back away to get a wider view (zooming)*". [Card, Mackinlay, Shneiderman 1999] do not use the term "panning and zooming" in their listing of interaction techniques. Their equivalent is "camera movement" on one side and "zoom" on the other side. In contrast to simple panning, camera movement includes the third dimension, when dealing with three-dimensional visualizations. In both papers, zooming includes possible changes of the level of details displayed, when changing the zoom factor. Also an interesting contribution, when talking about zooming, is the "single-axis-at-a-time-zooming", discussed by [Jog, Shneiderman 1995]. Whereas normal zooming can be explained by using a camera metaphor[87], this fails to work when only the scale of one the axes is changed. [Jog, Shneiderman 1995] call this single-axis-at-a-time-zooming, as shown in Figure 20.

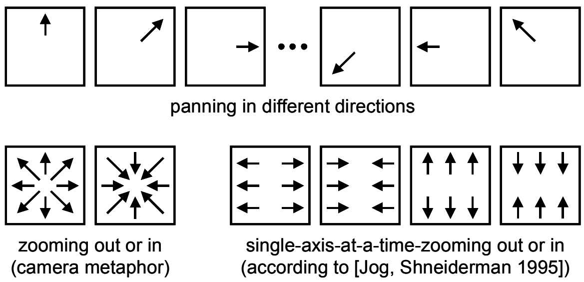


Figure 20: Panning and zooming, including different types of zooming

A classical example for a system implementing panning and zooming for the visualization of browsing and searching is Pad++ [Bederson, Hollan, Perlin et al. 1996]. One of the central characteristics of the system is the fact that scale is added as a first class parameter to all items displayed. In addition to implementing simple panning and zooming, Pad++ goes far beyond this interface technique. Besides other techniques it also offers focus-plus-context views as well as overview plus detail, described later. The explanations of [Bederson, Hollan, Perlin et al. 1996] using space-scale diagrams [Furnas, Bederson 1995] to explain basic concepts of panning and zooming, combinations of panning and zooming, and special problems when animating panning and zooming, are particularly interesting. In general at least simple forms of panning and zooming are today one of the general techniques implemented in a great many of the available visualization systems.

3.3.2.3. Focus-plus-context

An inherent problem of zooming leads to "focus-plus-context" as a solution. The problem is that the higher the zooming factor is, the more details can be shown about particular items or the better the separation between close up items, but less can be perceived from surrounding items or the overall structure. A solution for this problem is to present more details about the items in focus, and less about the context, avoiding completely hiding the context. [Card, Mackinlay, Shneiderman 1999] list as premises for focus plus context the following three points:

- The user needs both overview and detail information simultaneously.
- Information needed in the overview may be different than that needed in detail.
- These two types of information can be combined in a single (dynamic) display

[87] Not to be mixed up with the more complex camera movement metaphor used by [Card, Mackinlay, Shneiderman 1999]

As we will see in Chapter 3.3.2.6 overview plus detail is another method which can be used to cope with the mentioned problem of zooming and the first and the second of the above listed premises, but overview plus detail does not combine both types of information in single display.

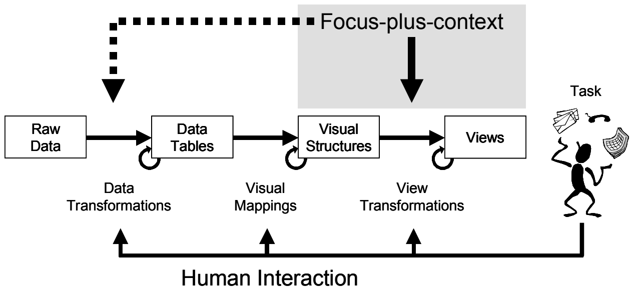


Figure 21: Reference model for visualization: Focus-plus-context

[Hearst 1999] describes a fisheye camera lens as a metaphor for focus-plus-context. The trailblazers for fisheye views were [Furnas 1981] / [Furnas 1986] with his theory about "Degree Of Interest" (DOI) functions, and [Sarkar, Brown 1992] with their extensions for graphical fisheye views. For a good overview of distortion-oriented presentation techniques see [Leung, Apperley 1994]. [Card, Mackinlay, Shneiderman 1999] list the following techniques for selective reduction of information for the contextual area: Filtering, Selective aggregation, Micro-macro readings, Highlighting, and last but not least Distortion. Explanations can be found in Table 16. Interestingly they interpret filtering in focus-plus-context as a data transformation, whereas for zooming, where a sort of filtering can also occur, they categorized the complete technique as working on the view transformation. The interpretation of [Card, Mackinlay, Shneiderman 1999], that focus-plus-context has at least partially to do with data transformations, is indicated in Figure 21 as a dotted line. Actually this should also be valid for panning and zooming in Figure 19, but has been omitted there because of the above-mentioned classification of the authors.

Technique	Explanation
Filtering	Selection of cases in the Data Table
Selective aggregation	Creation of new cases in the Data Table by aggregating other cases
Micro-macro readings	Graphics in which detail cumulates into larger coherent structures[88]
Highlighting	A overall set of items provides a macro environment against the micro reading of individual highlighted items can be interpreted
Distortion	Relative changes in the number of pixels devoted to objects in the space (more pixels for focus objects)

Table 16: Focus plus context: selective reduction of information for the context according to [Card, Mackinlay, Shneiderman 1999]

Examples for systems using focus-plus-context for the visualization of search results or browsing are the Document Lens, the Table Lens, or the Pad++ system. The Document Lens [Robertson, Mackinlay 1993] is a component of the Information Visualizer system. It is a 3D tool for large rectangular presentations of documents or Web page collections, like the WebBook. The pages of a document or a collection are exploded out, so that all pages are available simultaneously and can be viewed using a rectangular lens magnifying the page in focus, and therefore distorting all the other pages. The principle is shown in Figure 41 on page 77. Another component, also using a lens metaphor, is the Table Lens [Rao, Card 1994]. The Table Lens can be used for the viewing of re-

[88] Their example is the illustration of an new born infants sleep/wake cycles from [Winfree 1987], reproduced as Figure 1.8 in [Card, Mackinlay, Shneiderman 1999]

sults lists or other lists in tabular form, and includes functions for magnifying lines or groups of lines whilst keeping the rest of the table viewable in shrunken form. Pad++ [Bederson, Hollan, Perlin et al. 1996] includes, besides the above-mentioned panning and zooming functions, functions for focus-plus-context. In a tree-display of followed hyperlinks and corresponding pages, single pages can for example be magnified, with the rest of the tree still viewable.

3.3.2.4. Magic Lenses

Magic Lenses can be overlapped on items, and change their appearance by causing a transformation applied to the underlying data [Hearst 1999][89]. In general, they are a special form of focus-plus-context technique. As we have seen in the last chapter, the lens metaphor is also used for a number of other components implementing focus-plus-context techniques. Focusing on Magic Lenses [Fishkin, Stone 1995] describe Magic Lens Filters as tools providing mechanisms for visual transformations as well as for semantic transformations of the data. [Card, Mackinlay, Shneiderman 1999] classify Magic Lenses as techniques modifying view transformations, but explain "[…]*, they can apply data or view transformations on the item selected.*" This explanation is taken into account in Figure 22.

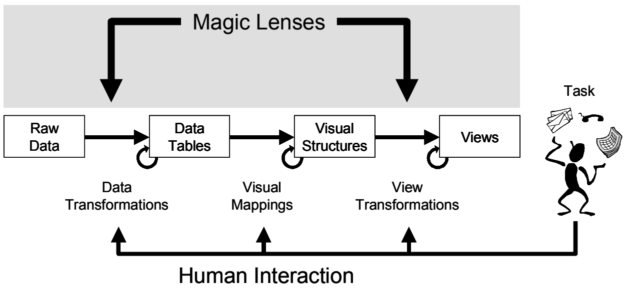


Figure 22: Reference model for visualization: Magic Lenses

[Bier, Stone, Fishkin et al. 1994] provide an (incomplete[90]) taxonomy of see-through tools as a framework for discussions about this technique. The framework uses 14 axes, ranging from trigger type to complexity. See-through tools or click-through tools are sometimes also called "lenses". Chapter 3.3.1 Metaphors includes a short comparison of the usage of the lens metaphor with Document Lens and Table Lens. [Stone, Fishkin, Bier 1994] provide examples of how magic lenses as movable filters can be used to change the view of items under the lens. [Fishkin, Stone 1995] show how they can be used to formulate database queries. The "portals" used in Pad++ [Bederson, Hollan, Perlin et al. 1996] provided, in addition to other functions, lens mechanisms by changing the way objects (e.g. documents) viewed through the portal are presented. Additionally Pad++ used lenses to design interfaces at the level of specific tasks. Examples are number entry lenses changing a generic number entry mechanism into a slider or dial, according to the user preferences. Lenses are also used in Pad++ to change the mapping of data tables to visual structures, for example by changing a presentation of numbers in columns to a scatter plot or bar chart.

[89] [Hearst 1999] mentions especially the usage of lenses in a two-handed way described in a number of publications like [Bier, Stone, Fishkin et al. 1994], or [Stone, Fishkin, Bier 1994]. In the two-handed condition two pointing devices are used simultaneously. The non-dominant hand controls the position of the lens, whereas the dominant hand is used to perform actions through the lens. Most other publications about lenses report solely the usage of lenses with a single pointing device.

[90] The authors themselves attribute their taxonomy as "not complete".

3.3.2.5. Animation

Whereas the other techniques described so far affect data transformations, visual mappings, and / or view transformations, animation does not influence these conversions, but is affected by them.

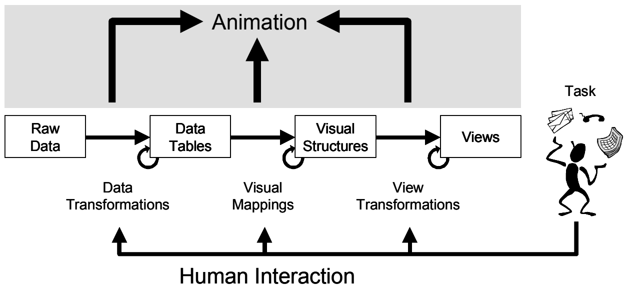


Figure 23: Reference model for visualization: Animation

For a discussion about animation in the larger context of motion and the general usage of motion see [Bartram 1997]. Animation is used more and more in information visualization systems to help users keeping their orientation when transformations or changes of mappings occur. The cognitive load on the user is reduced by providing object constancy and exploiting the human perceptual system [Robertson, Mackinlay, Card 1991], [Robertson, Card, Mackinlay 1993]. Animation is used in a number of already mentioned information-seeking systems like the Information Visualizer, the Navigational View Builder, Pad++, or SPIRE. In the Information Visualizer, animation is used in several ways, like for example animating rotations of Cone Trees to enable users to track substructure relationships without thinking about it [Robertson, Mackinlay, Card 1991]. Smooth animation is also used in the Navigational View Builder for zooming and filtering operations, to allow the user to see changes easily [Mukherjea, Foley, Hudson 1995]. One of the examples for the usage of animation in Pad++ is when clicking on a link the current page is not immediately replaced with a new page. Instead, the user first sees an animation where the new page is added to the tree of pages [Bederson, Hollan, Stewart et al. 1998]. In the SPIRE system of the MVAB project, animation is used in the Galaxies visualization to tie the document spatial patterns with temporal ones [Wise, Thomas, Pennock et al. 1995]. A tool called Temporal Slicer allows the user to partition the document base into temporal units. Moving a “temporal window” makes it than possible to watch the visualization populate itself with documents. In addition to animate changes, [Bryan, Gershman 2000] used movement in their “aquarium” interface for a large, online store to reinforce the absence of structure in the displayed items, because there is no first or last product, and no meaning of the proximity. Conceptually, their approach is interesting not only because of the usage of animation, but also because, for example, data transformations in form of filtering are not only triggered by user actions, but also by the absence of user actions. The displayed content changes gradually over time to show a diversity of products when no operation is performed for a certain period.

3.3.2.6. Overview plus detail

For overview plus detail, two or more levels of linked visualizations with different zoom factors are used. The technique helps users, while looking at a portion of the data in detailed level, keeping an overview of the whole structure. [Card, Mackinlay, Shneiderman 1999] differentiate between time multiplexed overview plus detail displays, and space multiplexed ones. When time multiplexing is used, overview and details are shown one at a time. When space multiplexing is used, overview plus detail are shown both at the same time at different locations on the screen.

Time multiplexed overview plus detail views are conceptually not far away from simple zooming. Overview plus detail is sometimes also called map view concept [Beard, Walker 1990]. [Card, Mackinlay, Shneiderman 1999] report that typical zoom factors range from 5 to 15, and that there is a limit for effective zoom factors of about 3 to 30.

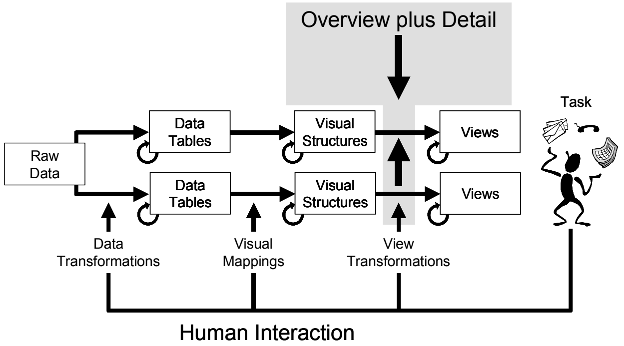


Figure 24: Reference model for visualization: Overview plus detail

Examples for systems using overview plus detail for the visualization of search results or browsing are the already on page 54 mentioned Harmony VRweb 3D scene viewer, or the pre-VIR prototype by [Bekavac 1999]. The Harmony VRweb 3D scene viewer [Andrews 1995] used a 2D-map for navigation in an information landscape. [Bekavac 1999] used in a prototype in preparation of the VIR-project a MapWindow in a horizontal tree view of the graph of the search results to ease navigation through the graph.

3.3.3. Components

It is beyond the scope of this work to give a complete overview of visualization ideas for the visualization of queries, search results or browsing. Nevertheless, in this chapter a number of ideas found in the literature will be presented to show the scope of mappings to visual structures already implemented by other authors. During the development of the INSYDER system, this overview served as a pool of ideas for the selection of candidates to be integrated. As already explained, the overview of ideas manifested in form of components will be structured using the phases of the four-phase framework for information seeking. Not all of the later listed possibilities to support these phases are originally targeted by the authors to sustain searching the Web. Potential candidates of doing this are also included.

Wherever applicable, a standard example for an assumed search about "Visualization of Search Results from the World Wide Web" will be used to explain the visualizations. It is subsequently referred to as the WebViz-example. For visualizations from classical IR and Web search, only this example will be used. When introducing components from other domains, where possible an example of the original usage will be given and in addition, the potential usage in Web searching using the WebViz-Example. When showing principles of visualizations using color palettes, an own color palette is used instead of the different color palettes from the original authors.

For the formulation phase a number of components for the visualization of query attributes will be presented. Despite the fact that sometimes visualizations are used during the action phase, for example in the form of progress bars, there will be no discussion of this phase in more depth. Visualizations are rarely used there, and the character of this phase focuses mainly on the internal processing of the system. From the user's point of view, the results phase is the most interesting phase. Here he gets the suggestions which satisfy his information need. If a long list of URLs is displayed, it would be a good idea to help the user finding the needle in the haystack by applying adequate visualizations. The discussion of components for the result phase will be subdivided into visualizations of document attributes, visualizations of interdocument similarities, and visualizations of interdocument connections. In terms of visualization, the refinement step has elements from the formulation and the result phase. Therefore visualizations for the refinement phase are discussed in the context of the formulation or the result phase.

3.3.3.1. Visualization of queries or query attributes

In the AI-STARS system, [Anick, Brennan, Flynn et al. 1990] used a component called "Query Reformulation Workspace" to visualize Boolean queries automatically derived from natural language queries. The ascertained citation forms are laid out as tiles two dimensional form, representing the Boolean queries with "AND" and "OR" conditions. The system carries out automatic operations on the query, like identification of noisewords or meaningful phrases. The results are also visualized. Figure 25 shows the Boolean query "('copy' AND 'BACKUP saveset' AND 'tape' AND ('v.5.0' OR 'version 5.0'))" automatically derived from natural language query "copying backup savesets from tape under v5.0". The example of [Anick, Brennan, Flynn et al. 1990] is based on a database with technical information for customer support specialists. Using the WebViz-example the query could be "(('visualization' OR 'visualisation') AND 'search' AND 'results' AND ('www' OR 'internet'))" automatically derived from the natural language query "Visualization of Search Results from the World Wide Web". The black tiles represent the query. The white tiles represent citation forms detected, but not automatically selected by the system. By clicking on the tiles the selections can be toggled. Additionally there are number of other functions like changing Boolean operators by moving tiles to other columns or requesting a window with related terms to expand or change the query. The related terms are grouped in phrases containing the term, synonyms, conceptually related terms, and compound terms. The numbers in the lower left corner of the tiles shows the number of postings of each term.

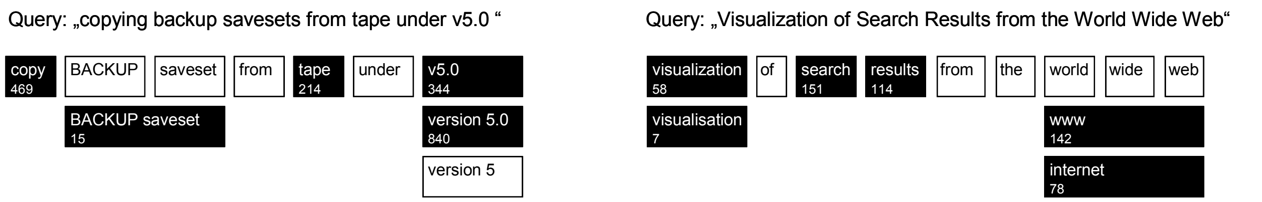


Figure 25: Principle of the Query Reformulation Workspace used in the AI-STARS system by [Anick, Brennan, Flynn et al. 1990]

As described on page 58 discussing the water flow metaphor, [Shneiderman 1991] / [Young, Shneiderman 1993] introduced a component called Filter/Flow to overcome known problems with the formulation of Boolean queries. The filters let through only the appropriate documents and the pipe layout determined if the relationship was an "AND" or an "OR". The left part of Figure 26 shows the simplified example of a complex query according to Figure 5 from [Young, Shneiderman 1993]. The example uses an employee database. The task is to find the accountants or engineers from Georgia who are managed by Elisabeth, or clerks from Georgia who make more than thirty thousand dollars per year. The right part of Figure 26 shows a transfer of the principle to the visualization of Web search results. Assuming an already found result set for the WebViz-example, the task is to filter English or German documents that are mixed linklists from academic servers, or high relevant English or German documents of all types except framesets.

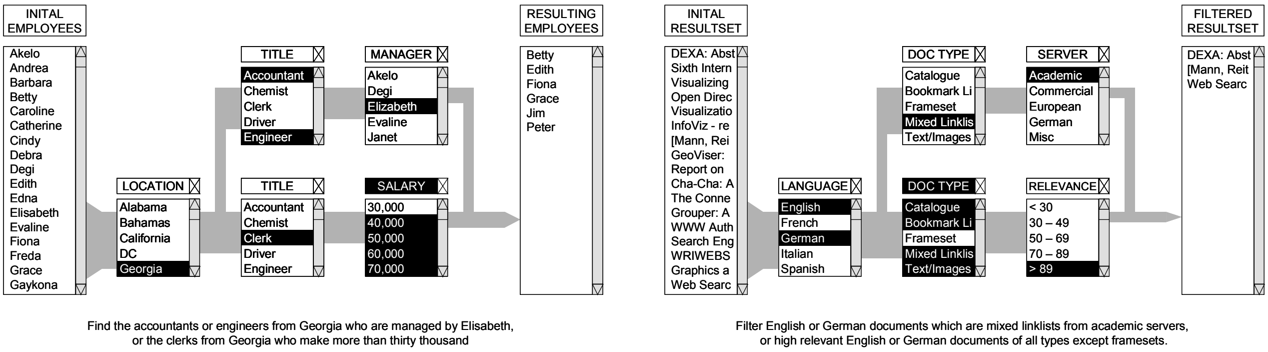


Figure 26: Principle of Filter/Flow component by [Shneiderman 1991], [Young, Shneiderman 1993]

Another visualization of the filter effect of a query has been proposed by [Fishkin, Stone 1995] with the Movable Filters or Magic Lens Filters, already mentioned on page 55. The component allows compound queries to be constructed by overlapping lenses. The left side of Figure 27 shows an example taken from [Fishkin, Stone 1995] where two filters are combined in an "AND"-condition on a map with symbols for cities. One filter allows only cities where taxes are low to pass, the other one only cities with high salaries. Grey rectangles passed the filters. White rectangles do not fulfill the conditions of the query. The example shows cities with high salaries AND low taxes in gray. The threshold values can be changed by a control. Buttons can change the modes of the filters. Besides "AND" and "OR", there is a SELF option to switch to a mode where only the effect of this filter is shown, and a NOP option to switch off the effect of this filter. There are a number of other options described in [Fishkin, Stone 1995], including real-valued filters, a magic lens filter showing cities where no values are available, and a callout lens mechanism. Real-valued filters show not only the presence of a feature, but also it's value by filling the symbol more or less with a color according to the real value. The callout lens allows user to explore "clumbs" where a number of icons are close to each other or overlap. The callout lens displays the items in form of a list, and therefore allows an easy and detailed inspection. The list includes the icons from the scatterplot. These icons are active in the callout lens, and can, for example, be filtered by overlapping additional lenses. The right side of Figure 27 shows a transfer of the general lens principle to the visualization of Web search results. The earlier established result set from the WebViz-example functions as the basis. Every document is represented by a rectangle in a scatterplot. The language of the document is shown on the x-axis. The number of the documents in the language category is shown on the y-axis. Two filters in AND-condition are used: one for the relevance allowing only documents with a high relevance score to pass, and a second one for the document type, allowing only documents without framesets to pass. Therefore, only non-frameset documents with a high relevance are marked gray.

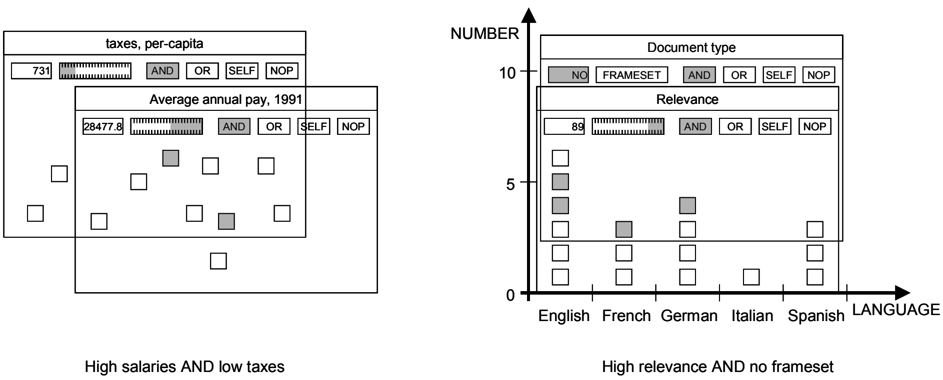


Figure 27: Principle of Movable Filters / Magic Lens Filters by [Fishkin, Stone 1995]

Venn diagrams have been used in a number of cases to represent Boolean queries. One recent example is the usage in the TeSS prototype by [Hertzum, Frøkjær 1996]. A good overview of Venn diagrams can be found in [Jones 1998]. Simple Venn Diagrams are capable of dealing with two or a maximum of three keywords. Figure 28 shows the principle of Venn diagrams for a part of the result set for the WebViz-example. Starting in the upper left corner, the blue circle represents 18 documents retrieved by '(visualization OR visualisation) AND (NOT (search OR results))'. The intersection of the two upper circles shows the 8 documents retrieved by '(visualization OR visualisation) AND search AND (NOT results)'. The intersection of the three circles contains the 32 documents which are retrieved by '(visualization OR visualisation) AND search AND results'.

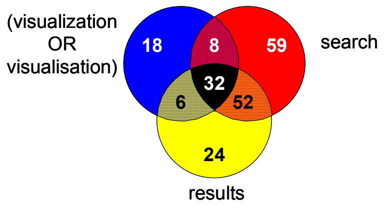


Figure 28: Venn diagram for the concepts (visualization OR visualisation), search, results.

[Jones 1998] integrated Venn diagrams in the VQuery interface in a query workspace to support users in a more flexible way when working with this type of visualization. Figure 29 shows an illustration using the WebViz-example. Six keywords are spread over the workspace. Currently the active query, represented by the gray rectangle, includes three of them. The query is '(visualization AND search) OR results'. Part of the workspace is a text field, where the system presents an English language interpretation of the graphically constructed active query. Besides "AND" and "OR", the systems support also a NOT operator, but complex queries are impossible to construct.

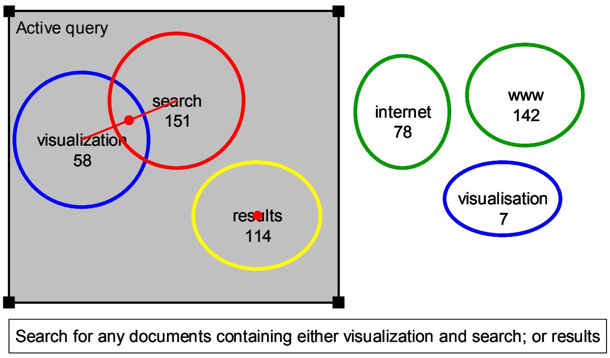


Figure 29: Principle of the Query workspace with Venn Diagrams in the VQuery system by [Jones 1998], [Jones 1998a]

[Spoerri 1993], [Spoerri 1993a] introduced with the InfoCrystal a query-visualization component also derived from Venn diagrams. The InfoCrystal can be used as a visualization tool and as visual query language. Spoerri describes the usage for Boolean or for vectorspace queries, and different modes like simple queries or complex queries using a block building mode. The layout inside an InfoCrystal can be done in rank layout or bull's-eye layout. Information is coded in shape, proximity, rank, orientation, and color or texture. In special cases size, or brightness and saturation coding is used. Figure 30 shows an InfoCrystal for the WebViz-example. It is a simple query in rank layout with color-coding. The number in an icon shows the number of documents satisfying the conditions represented by it. Starting in the upper left corner, the blue circle represents 64 documents retrieved by 'visualization OR visualisation'. The next blue circle shows one document retrieved by '(visualization OR visualisation) AND (NOT (search OR results) OR (www OR internet))'. The rectangle with a blue and a green end stands for 18 documents retrieved by '(visualization OR visualisation) AND (www OR internet) AND (NOT (search OR results)'. The triangle with blue, red, and light green sides stands for 2 documents retrieved by '(visualization OR visualisation) AND search AND results AND (NOT (www OR internet)'. The rhombus in the middle stand for 27 documents retrieved by '(visualization OR visualisation) AND (www OR internet) AND search AND results)'. Icons, which represent two diagonally opposite concepts are represented twice. When specific relevance weights are assigned to concepts, the bull's-eye layout can be used instead of rank layout. In this mode, symbols for relationships with a higher relevance score are placed closer to the center of the InfoCrystal.

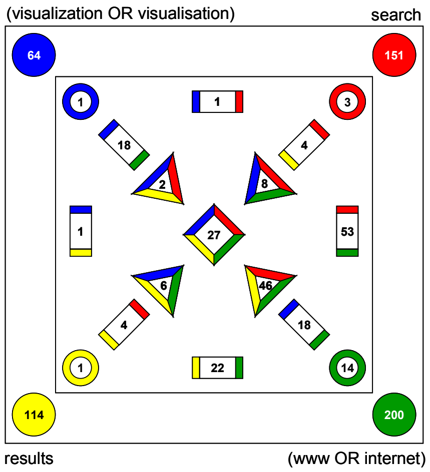


Figure 30: Principle of the InfoCrystal by [Spoerri 1993], [Spoerri 1993a]

[Bürdek, Eibl, Krause 1999], [Eibl 1999] re-implemented the InfoCrystal and had a number of problems when using it. Guided by the speculation that the coding of the InfoCrystal is too manifold and that the presentation is too complex they developed a new visualization. One of their motivations was the need for reorientation when adding additional keywords in the InfoCrystal. The principle of their solution is shown in Figure 31 using the WebViz-example. The visualization has two basic elements: the entry fields on the left side, and the resulting document sets on the right side. Keywords or field-based restrictions can be entered in the entry fields. Fields can be selected by clicking on the "T" [Eibl 1999]. Keywords under a bracket are "ORed". The brackets themselves are "ANDed". So Figure 31 shows the query '(visualization OR visualisation) AND search AND results AND (www OR internet)'. On the right side the resulting document sets are displayed. On the rightmost side is the result of the complete query, combining all four brackets. The other columns show all possible combinations between two or three Brackets in distinct mode. [Eibl 1999] reports that six of the eight users interviewed, preferred distinct mode, in which the first set stands for '(visualization OR visualisation) AND search AND (NOT (results OR (www OR internet)))'. In non-distinct mode where it would have been '(visualization OR visualisation) AND search'. A careful examination of Figure 1 in [Bürdek, Eibl, Krause 1999] and Figures 1c/d in [Eibl 1999] reveals that the distinct mode is used for the columns containing more than one concept, but not for the entry brackets where just one concept is shown. Eibl et al. do not mention this point. In Figure 31 the same mechanism is used like by the authors. To be fully consistent in distinct mode the number of documents in the first row should be 1, 3, 1, and 14 instead of 64, 151, 114, and 200. The InfoCrystal shows both numbers for single concepts, distinct and non-distinct. Besides this minor inconsistency, which could be avoided by showing both numbers, the "Bracket"-visualization has a number of features important to improve usability. [Eibl 1999a] reports for example, that users adopted very fast, instead of interpreting the colors, the feature that when crossing a result set with the mouse the corresponding query-Brackets are dimmed[91]. Like the InfoCrystal, the visualization by [Bürdek, Eibl, Krause 1999], [Eibl 1999] has a number of additional features beyond simple Boolean retrieval. Among them is the support of probabilistic retrieval by using the horizontal position of the Bracket-groups, or the support of vague retrieval.

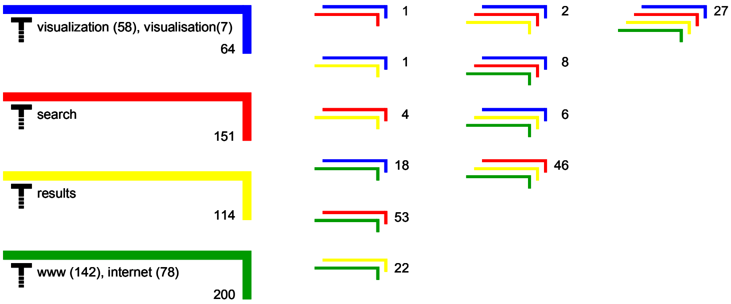


Figure 31: Principle[92] of the "Bracket"-visualization by [Bürdek, Eibl, Krause 1999], [Eibl 1999]

[Berenci, Carpineto, Giannini 1998] created the VIEWER (VIEws of WEb Results) system where a component also shows graphically the distribution of sub queries. The length of a bar indicates the size of the result set of every sub query. Clicking on the bar brings up the subset in list form in a second window. By carefully analyzing Figure 1 of [Berenci, Carpineto, Giannini 1998] it can be found, that in contrast to the InfoCrystal and the "Bracket"-visualization, the subsets are shown in non-distinct mode. Figure 32 shows the principle using the WebViz-example.

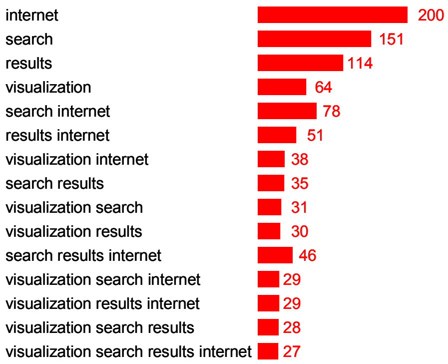


Figure 32: Principle of the Bargraph in the VIEWER system by [Berenci, Carpineto, Giannini 1998]

[Cugini, Laskowski, Piatko 1998] used in the NIRVE system a component called Concept Control to allow the user to map keywords to concepts. Later the component has been named Keyword-Concept Matrix [Cugini, Laskowski, Sebrechts 2000]. Besides the possibility to group keywords into comprehensive concepts, the Keyword-Concept Matrix also allows for each of the resulting concepts to assign a "weight" or importance. Each concept has its own color attribute. Figure 33 shows the principle using the WebViz-example. The keywords "visualization" and "visualisation" are mapped to the concept "visualization". "Www" and "internet" are mapped to "internet". "Search" and "results" are mapped to themselves. The concepts "visualization", "search", and "internet" are important and therefore received a high weight value. "Results" is marked as less important. In later versions of the NIRVE system [Cugini, Laskowski, Sebrechts 2000], the authors used an interactive legend instead of the matrix. The concepts are shown in a row with the

[91] It seems to be more logic to highlight the brackets. Nevertheless it appeared to have the expected effect.

[92] The original uses a black background and is visually optimized by using interface and media design principles not used for the reproduction. In [Eibl 1999] a "T"-symbol is used for field-selection, in [Bürdek, Eibl, Krause 1999] a triangle is used for the same purpose.

corresponding keywords beneath of them. The mapping could by changed by drag and drop. An extra column is reserved for unused keywords. The lower part of Figure 34 shows the principle of the interactive legend.

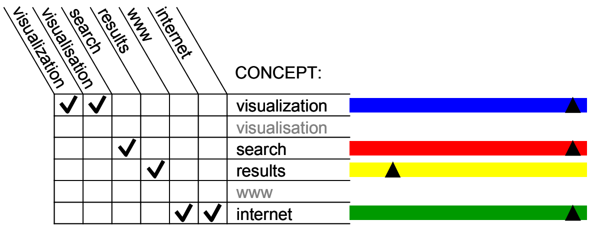


Figure 33: Principle of the Keyword-Concept Matrix or Concept Control used in the NIRVE system by [Cugini, Laskowski, Piatko 1998], [Cugini, Laskowski, Sebrechts 2000].

Starting with visualizations of interdocument similarities and document clusters at a later point of the development of the NIRVE system, a so-called Concept Globe [Cugini, Laskowski, Sebrechts 2000] has been added, showing per default no single documents but only document clusters, the concept distribution and average relevance in the cluster, the number of documents in the cluster, and a number of other features. The primary design version was a 3D globe, but the authors also experimented with 2.5D and 2D versions. The definition of a cluster is guided by previous user experiences and is quite simple: all documents that have the same subset of concepts form a cluster. The clusters are visualized starting at the North Pole of the globe, or the upper end in the 2D version, starting with the cluster containing all keywords. In the next row are the clusters in which one of the concepts is missing, in the next row two concepts are missing and so forth. At the South Pole, or lower end in the 2D version, would be the cluster of documents where all concepts are missing. So the number of concepts defines the „latitude" of an icon representing a cluster. In the 3D version the thickness of the box of a cluster represents the number of documents in the cluster. The height of a rectangle below the cluster icon indicates the same value in the 2D version. Presence or absence of colored bars indicates the presence or absence of concepts. Colored lines between the icons indicate concept differences between clusters. Neglecting the length of the bars, indicating the average relevance of a concept for the documents in the cluster, and some other features not described here, the Concept Globe the presents almost the same information like visualized in the InfoCrystal or the "Bracket"-visualization. Figure 34 shows the 2D principle using the result set from the WebViz-example.

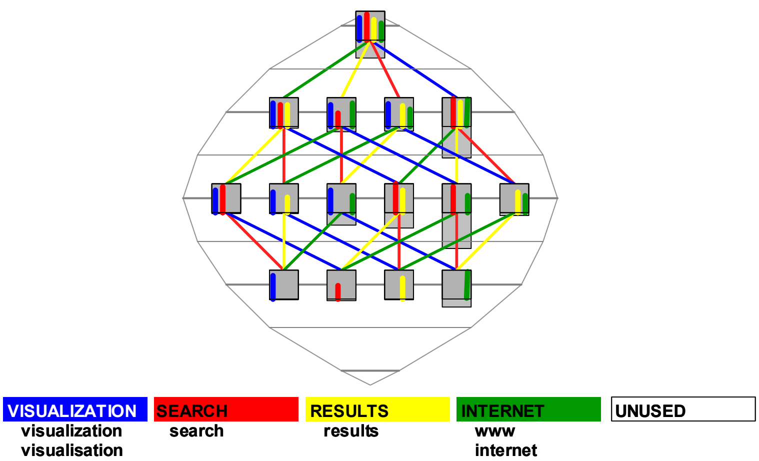


Figure 34: Principle of the 2D Global View used in the NIRVE system by [Cugini, Laskowski, Sebrechts 2000]

In the Query Reformulation Workspace by [Anick, Brennan, Flynn et al. 1990] we have seen some basic functionality to support query expansion. Additionally there has been a feature to look for related terms grouped in phrases containing the term, synonyms, conceptually related terms, and compound terms. There are a number of other components supporting query expansion or refinement by more or less sophisticated visualizations. In general, these components show relations between keywords or concepts, stored in a thesaurus or computed on the fly by analyzing document sets. Examples for supporting the query formulation or expansion can be found in [Fowler, Fowler, Wilson 1991], [Fowler, Wilson, Fowler 1992] where the visualization of a network structure is used, or [Veerasamy, Navathe 1995], where additional items from a thesaurus are listed below the entered keywords. Figure 35 shows the principle used by Fowler et al. for the visualization of a query as a Request Map in the Information Navigator. The system uses statistically-based associative structures, PFNETS, and a spring based network display layout-algorithm not only for the visualization of the query, but in addition with fisheye-techniques also for the visualization of the documents and the concepts found in the document base.

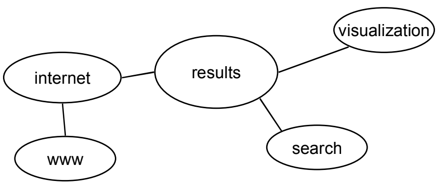


Figure 35: Principle of the Request Map by [Fowler, Fowler, Wilson 1991], [Fowler, Wilson, Fowler 1992]

Graphically simpler is the approach by [Veerasamy, Navathe 1995] / [Veerasamy, Hudson, Navathe 1995] used in the Tkinq system[93]. Additional items from a thesaurus are listed below the entered keywords. Drawing them to a positive or negative feedback box causes them to be included in the query. Positive items are "ORed" with the entered term, negative items are included into the query with a NOT operator. The implementation shown in [Veerasamy, Navathe 1995] / [Veerasamy, Hudson, Navathe 1995] has a number of insufficiencies, like the order of the boxes or the not very intuitive feedback about the actual constructed query. Nevertheless, the idea has some interesting aspects.

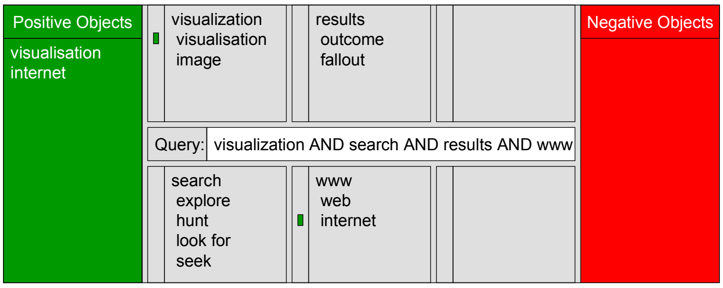


Figure 36: Principle of Positive / Negative Feedback by [Veerasamy, Navathe 1995], [Veerasamy, Hudson, Navathe 1995]

An example for the support of a query expansion in the refinement step, which has some similarities with the formulation phase, is the Cow9 graphical query refinement by [Bourdoncle 1999] used by AltaVista[94]. Cow9 had initially been named LiveTopics [Bourdoncle 1997]. Cow9 shows

[93] The name is not used by [Veerasamy, Navathe 1995], but in their Figure 1 the main window of the system is named "Tkinq". In [Veerasamy, Hudson, Navathe 1995], [Veerasamy 1996], [Veerasamy, Belkin 1996] Tkinq can also be seen in the window title or as label of the system-quit button, but the name is also not mentioned in the text. A number other authors like [McCrickard, Kehoe 1997] use this name to reference the system from Veerasamy et al.

[94] The author used the feature in the years 1997 and 1998. In 2001 it seems to be discontinued.

a map of expandable topics, automatically constructed from the terms contained in the document set and the query. Yellow bars to the right of each word indicate the probable relevance of that word to the query. Words can be marked as included or excluded. In Figure 37 “visualisation” is marked as required, and “draw” is marked as excluded. Initially only used for single terms, the technique has later been expanded for the usage of word groups[95] [Bourdoncle, Bertin 2000].

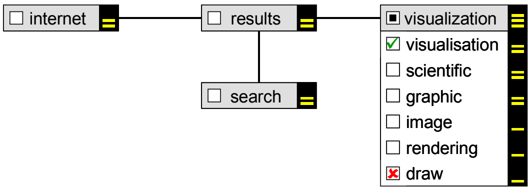


Figure 37: Principle of LiveTopics / Cow9 by [Bourdoncle 1997]

This chapter introduced a number of ideas for the visualization of query attributes. Table 17 gives an overview covering the components discussed. Besides the raw query in itself and its structure we have seen visualizations supporting previews of the result set in form of information about the number of hits and their distribution in the Boolean space. We have seen visualizations supporting the fine-tuning of the query or grouping functions for the mapping from raw data over data tables to the visual structures used for display of the result set. Some visualizations have been introduced supporting query expansion or refinement using a thesaurus or automatically constructed term sets. Especially in this area is a considerable number of other ideas or components not discussed here. The question of visualizations for the formulation or refinement phase in case of the INSYDER system is discussed in more depth in [Mußler, Reiterer, Mann 2000], [Mußler 2002]. This thesis turns now to visualizations supporting the result phase of an information seeking process.

Component	Literature	Used in System
Query Reformulation Workspace	[Anick, Brennan, Flynn et al. 1990]	AI-STARS
Filter/Flow	[Young, Shneiderman 1993]	
Movable Filters / Magic Lens Filters	[Fishkin, Stone 1995]	
Venn Diagrams	[Hertzum, Frøkjær 1996]	TeSS
	[Jones 1998], [Jones 1998a], [Jones, McInnes, Staveley 1999]	VQuery
InfoCrystal	[Spoerri 1993], [Spoerri 1993a]	
“Bracket”	[Bürdek, Eibl, Krause 1999], [Eibl 1999], [Eibl 1999a]	GESINE
Bargraph	[Berenci, Carpineto, Giannini 1998]	VIEWER
Keyword-Concept Matrix / Concept Control	[Cugini, Laskowski, Piatko 1998], [Cugini, Laskowski, Sebrechts 2000]	NIRVE
Concept Globe / Global View	[Cugini, Laskowski, Sebrechts 2000]	NIRVE
Request Map network display	[Fowler, Fowler, Wilson 1991], [Fowler, Wilson, Fowler 1992]	Information Navigator
Positive / Negative Feedback	[Veerasamy, Navathe 1995], [Veerasamy, Hudson, Navathe 1995]	Tkinq
Cow9, LiveTopics	[Bourdoncle 1997], [Bourdoncle 1999], [Bourdoncle, Bertin 2000]	AltaVista

Table 17: Components for the visualization of queries or query attributes

[95] “*groupes nominaux*”

3.3.3.2. Visualization of document attributes

In the last chapter, which focused on the visualization of queries, we have already seen the first ideas of how to present the results of a search. This has been in very condensed form, showing only the number of (possible) results, as well as some ideas how the result set could be structured. Now we move to the visualization of single results. When getting a result set from a query, after zooming or filtering steps, the focus may change from the set level to specific suggestions (commonly called hits) at the document level. The user has to decide if a specific URL is interesting enough to be pursued. Visualizations are needed to get an overview of the structure, the content, or the usefulness for the user of a single document. Approaches range from the Document Lens over Dotplots to TileBars.

A thumbnail view can provide the first glance of a document, or probably as in most cases of the first page of a document. Figure 38 shows some examples. Thumbnail views are used by a number of authors for different purposes. In a number of systems, thumbnail views are used for providing a more ore less sophisticated bookmark functionality. Examples are DeckView, Data Mountain, or TopicShop. In DeckView [Ginsburg, Marks, Shieber 1996] persistent or temporary bookmarks inside a coherent document are one part of the functionality provided using thumbnail views. The other part is a navigation tool, that moves through the pages of a document using a deck of thumbnail views of the pages. In the Data Mountain system [Robertson, Czerwinski, Larson et al. 1998] thumbnails views can be used to organize bookmarks in a spatial layout. In [Czerwinski, Dumais, Robertson et al. 1999] the Data Mountain system is enhanced by the usage of implicit queries automatically highlighting pages related to the currently selected Web page. In TopicShop [Amento, Hill, Terveen et al. 1999] thumbnail views of Web sites index pages are also used as effective visual identifiers for sites the user has already visited and to provide functionality for organizing resources spatially. TopicShop has its own crawling engine collecting information and creating thumbnails views by starting with a list of user provided URLs. Whereas the Data Mountain system uses a 3D layout to place the thumbnails, the TopicShop system uses a 2D layout.

Another usage of thumbnail views is the creation of a graphical browsing history as done in the Graphic History View of the MosaicG [Ayers, Stasko 1995] browser, the Pad++ based PadPrints [Hightower, Ring, Helfman et al. 1998], or the Pad++ Web browser [Bederson, Hollan, Stewart et al. 1998]. The webView system [Cockburn, Greenburg, McKenzie et al. 1999], [Cockburn, Greenberg 1999], [Cockburn, Greenberg 1999a] integrates navigation history and bookmark functionality. In addition to the usage of miniaturized views of pages, the authors experimented with some extra graphical “decorations” showing bookmark status or visitation attributes. The right part of Figure 38 shows three examples using thumbnail views of the German homepage of the University of Konstanz. Figure 38a) shows the principles described in [Cockburn, Greenburg, McKenzie et al. 1999], [Cockburn, Greenberg 1999] and [Cockburn, Greenberg 1999a]. The green dog-ear in the upper left corner shows as an implicit bookmark indicator the number of visitations of this page. The more times the page has been visited, the darker is the dog-ear. The red dog-ear in the lower left corner shows as an explicit bookmark indicator that this page had been bookmarked by the user. In [Cockburn, Greenberg 1999], [Cockburn, Greenberg 1999a] the bar on the right side functions as an additional indicator for the relative time of the first and last access of the page. The lower blue flash stands for the first visit of the page, the upper one for the last. The lower end of the bar is equivalent to the oldest page in the display, the upper end to the current time. The thumbnails themselves are integrated in an add-on-tool for the Netscape Navigator using thumbnails in temporal views, or hub-and-spoke views. In [Kaasten, Greenberg 2000], [Kaasten, Greenberg 2001] the versions b) and c) can be found integrated in a modified Microsoft Internet Explorer. The bar shows the number of visits by its length and darkness. The dog-ear is the explicit bookmark indicator.

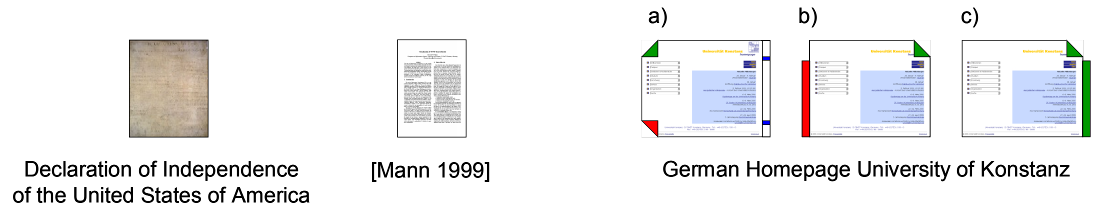


Figure 38: Thumbnail Views

Despite of the potential importance of revisitation aids[96] this usage of thumbnail views is not the main focus of this thesis. Only one of the systems mentioned so far used thumbnails for the visualization of search results. In TopicShop thumbnail views are used as a visual identifier for a group of pages heuristically grouped into a site. Examples for systems where thumbnail views are used to give overviews about keyword distributions in documents are the Document Lens [Robertson, Mackinlay 1993], an unnamed system by [Kaugars 1998], or the J24 system described in [Ogden, Davis, Rice 1998]. [Kaugars 1998] uses four different forms with increasing levels of detail for presenting retrieved articles: closed, thumbnail, semi-open, and fully open. Figure 39 shows a document set of 20 hits from the WebViz-example with 17 documents "closed" and three documents[97] in "thumbnail" view. The screen layout for positioning the documents used by [Kaugars 1998] is somewhat different, but the principle is the same. The length of the bar shows the relevance of the document in both forms "closed" and "thumbnail". In the thumbnail view keyword hits are color-highlighted. The semi-open view can be seen in Figure 42 on page 78. "Fully open" is a normal view of the full text.

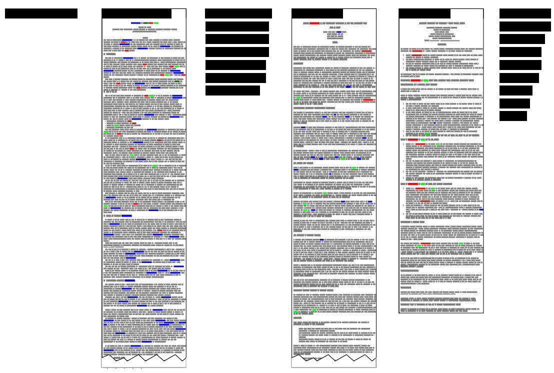


Figure 39: Principle of "closed" and "thumbnail" views with relevance indicator and colored keyword hightlighting by [Kaugars 1998]

The J24 system by [Ogden, Davis, Rice 1998] uses a thumbnail view similar to the one used by [Kaugars 1998], but without the relevance indicator. Figure 40 shows the first 10 hits using the WebViz-example-query with a document set of The Financial Times of London. Besides the thumbnail view, the system has two additional views of the documents, a summaries view and a fulltext view. In contrast to the system by [Kaugars 1998], where every document is only show in

---

[96] According to [Tauscher, Greenberg 1997], [Tauscher, Greenberg 1997a] about 58% of all Web pages visited in a five to six week period are pages the users had already visited before.

[97] The documents are versions of [Mann 1999], [Hearst 1995], and [Veerasamy, Navathe 1995] with one column and no figures. Two documents are only shown partially in the figure. [Kaugars 1998] shows documents in full length.

one view at a time, J24 shows the document in two views simultaneously, and the selected document in all three views. The selected document is marked with a red dot.

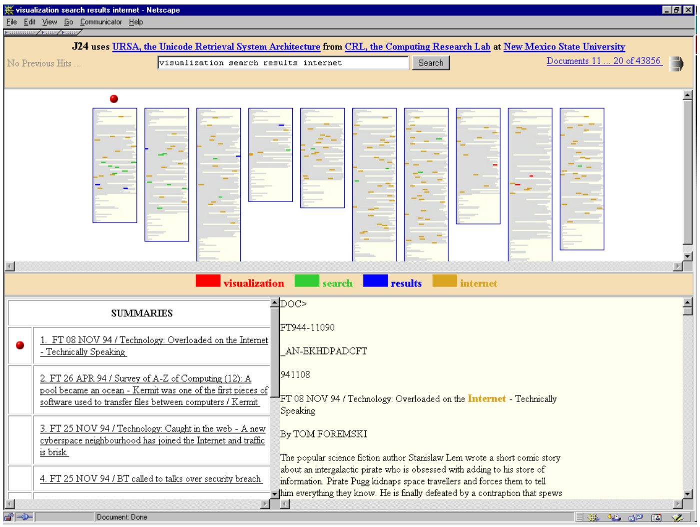


Figure 40: Thumbnail views in the J24 interface by [Ogden, Davis, Rice 1998][98]

The Document Lens developed in the Information Visualizer Project by [Robertson, Mackinlay 1993] has already been mentioned. Using a focus-plus-context technique, the Document Lens is a 3D tool for large rectangular presentations of documents or Web page collections. The pages of a document or a collection are exploded out, so that all pages are available simultaneously and can be viewed using a rectangular lens magnifying the page in focus, and therefore distorting all the other pages. The more the pages are away from the focus, the more they are distorted and "thumbnailed". The principle is shown in Figure 41. The page in the focus is shown in gray. When used for a coherent document, the user gets at least an impression about the number of pages in the document. The lens allows a fast examination of the pages the user is interested in. In addition, [Robertson, Mackinlay 1993] describe that the tool makes it quiet easy to show query term hits (at least for a coherent document). Even when viewing part of the document up close, patterns in the whole document become evident when the query term hits are color highlighted.

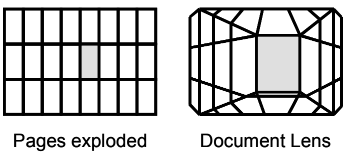


Figure 41: Principle of the Document Lens by [Robertson, Mackinlay 1993].

The system by [Kaugars 1998] uses also a focus-plus-context distortion technique. In the semi-open view paragraphs of the document without keyword hits are displayed in compressed form. [Ogden, Davis, Rice 1998] discuss this multiple fish-eye technique by referencing the PhD-thesis from Kaugars. They mention compressed paragraphs and sentences. Figure 42 shows the principle with compressed sentences using the first paragraphs of a single column version of the document [Mann 1999] and the WebViz-example query.

[98] Figure produced by searching The Financial Times of London http://messene.nmsu.edu/ursa/J24/ [2001-02-11]. The string "Documents 11 .. 20 of 43856" is not the label of this page, but the link to the second page of results.

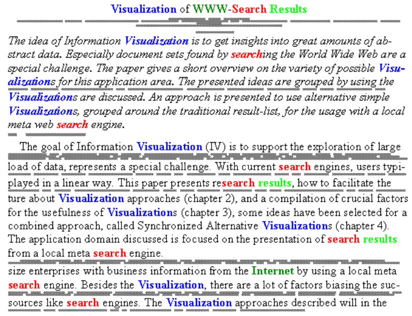


Figure 42: Principle of semi-open documents by [Kaugars 1998], [Ogden, Davis, Rice 1998]

In DigOut4U, a local meta search engine software from Arisem S.A., Paris, France a conceptually similar multiple-focus fisheye technique is used to display document extracts. The component is enhanced by using a slider combined with a "curve of relevance" to control the amount of displayed text. The curve of relevance represents the distribution of the relevant information for the query in the document. The higher the slider is positioned, the more text passages are suppressed in the "Relevant Extracts". In the initial position, the slider is in a high position leading to a very short extract by displaying only the text segments with the highest relevance for the query. The more the slider is lowered, the longer the extract becomes. The suppressed text segments are symbolized by "[..]" independent of their length instead of being distorted like in the semi open documents from [Kaugars 1998], [Ogden, Davis, Rice 1998]. Figure 43 shows an example of the relevance curve plus the relevant extracts using the document [Mann 1999] and the WebViz example query. From left to right three slider positions are shown, displaying more and more text. The relevant extracts do not use keyword highlighting. Using a semantic analysis based on a thesaurus as a knowledge base does the ranking of documents and text segments in DigOut4U. In a number of cases the query terms themselves do not appear in the relevant text segments. The first picture in Figure 43 is a good example for this functionality. The query was "information search results internet", and the highest ranked text segments contain "Visualization of WWW Search Results".

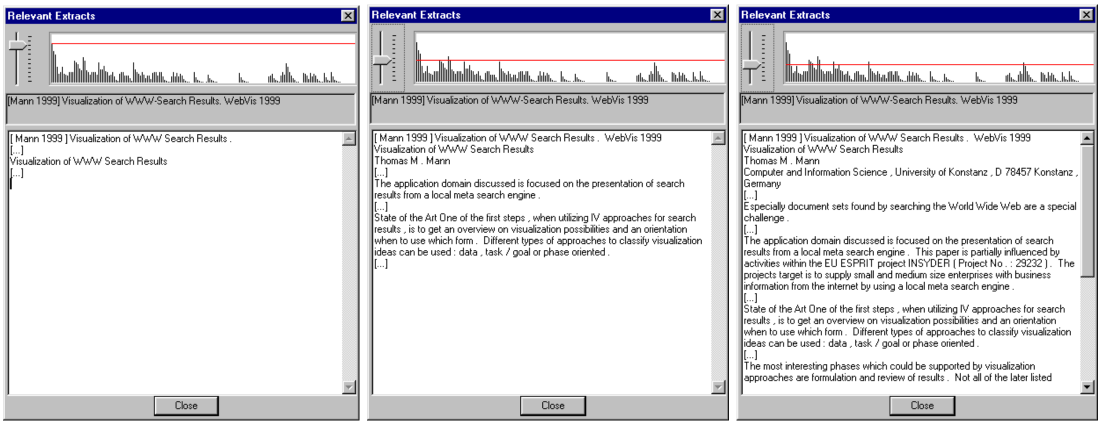


Figure 43: Relevance Curve and Relevant Extracts from DigOut4U of Arisem S.A. Paris[99]

[Eick, Steffen, Sumner 1992] introduced SeeSoft as a tool for visualizing line oriented software statistics. The SeeSoft system includes a number of interactive views and uses techniques like animation, or brushing and linking. Key ideas behind SeeSoft include a reduced representation of text, and coloring by statistics. The dominant view of the code is a bar, where the reduced representation of a line of code is colored according to an associated statistical value. One row of code

---

[99] Figures produced by using DigOut4U Version 1.4, obtained generously for research purposes from Arisem S.A., Paris, France. Free evaluation versions of DigOut4U can be downloaded from http://www.arisem.com [2001-02-11]

is represented by one row of pixels. Examples of dimensions mapped to a color code are date of modification or author. Besides software code, the authors had a number of ideas for other application areas including indexed text, such as legal writings or the Bible. [Eick 1994] has an example of the visualization of the verses of the King James Bible. [Wills 1995] visualizes The Jungle Book by Rudyard Kipling. In the example of the Bible, word usage patterns are visualized. The books of the New Testament are laid out in columns. The verses are represented in rows. Both rows and columns are folded if they are too long to be displayed in one piece. The visualization reveals patterns such as the occurrence of keywords like “angel” or “adultery”. In the example of the Jungle Book, the occurrence of the major characters in the book is displayed. Chapters are mapped to columns and lines to rows.

Figure 44 shows the principle of the SeeSoft bar view using the WebViz example query and the document [Mann 1999][100]. The paragraphs are mapped to columns. The lines are mapped to rows. The figure includes only a visualization of the abstract and the first paragraph of the paper. Additionally, the text of the abstract is shown.

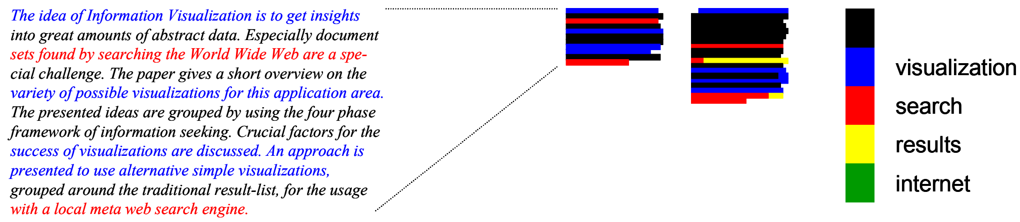


Figure 44: Principle of the SeeSoft bar view according to [Eick 1994], [Wills 1995] [101]

The components discussed so far in this chapter are mainly based on miniaturized or miniaturized and distorted views of documents. In addition, color-highlighting has been used. We now turn to some more abstract representations of documents and their attributes. [Jerding, Stasko 1995], [Jerding, Stasko 1997] introduced with the Information Mural a component capable of visualizing information in an even more condensed way as the SeeSoft bar view. Initially used for the visualization of software program execution patterns, the authors describe a number of other usage scenarios for the Information Mural including the visualization of documents. In the case of text visualization the basic idea is to use grayscale shading and / or color together with anti-aliasing techniques to go beyond the limit of one line of text per one row of pixels. The more information is compressed into a pixel, the darker (or the lighter, depending on the mapping) the pixel.

One of the components of the Envision system is a Matrix of Icons in the Graph View [Nowell, France, Hix et al. 1996]. In Envision, layout semantics are under user control. Different visual mappings from variables to visual structures can be selected by the user. The Graph View itself is modeled after scatterplots and therefore a component more suitable to show interdocument similarities. Nevertheless, some of the mappings in the Matrix of Icons have their own quality for the visualization of document attributes. Figure 45 shows examples of mappings in the Envision system. On the left side the document type is mapped on the icon shape, and relevance is mapped on

[100] Left justification used instead of block justification.

[101] In Figure 6 from [Wills 1995], when two characters appear in the same line, only one seems to be color-coded. The line “*Shere Khan would kill him some day; and Mowgli would laugh and*” is colored only with the color of Mowgli, and not the one from Shere Khan. [Hearst 1999] has in her Figure 10.16 also the SeeSoft Jungle Book example. Magnifying the figure it can be detected that there are a number of lines where two colors appear. Examination of the

size and color. The right side shows another possible mapping, where the relevance is mapped on icon type and color. Other attributes such as author names or publication year can be mapped on the placement along the x-axis and y-axis and will be discussed in Chapter 3.3.3.3 Visualization of interdocument similarities.

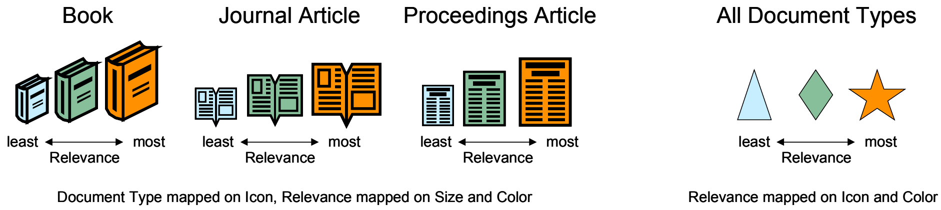


Figure 45: Possible mappings in the Matrix of Icons of the Graph View of the Envision system according to [Nowell, France, Hix et al. 1996]

[Church, Helfman 1993] introduced Dotplots to investigate different sorts of text by visualizing self-similarity of tokens in the text or in the meta-information of text. The goal is to support the discovery of large-scale structures. In a first step the text is split into lines, words, or characters. In a second step a plot is generated, where a dot is placed in every position i, j where the i^{th} input token is the same as the j^{th}. For Dotplots of text the token is usually a word. Besides simple plots, Dotplots have a number of features such as reconstruction, weighting, approximation, and the usage of greyscale or colormaps to visualize results. Dotplots also support overview plus detail with multiple views of a text as plots in two different scales and an additional text window. [Church, Helfman 1993] experimented besides other forms of text like source code with: four Associated Press (AP) news stories about the same topic, the protocols from Canadian parliamentary debates in English and French, and Microsoft manuals in seven languages. Patterns detected are reverse diagonals, broken diagonals, light crosses, checkerboards, reordered diagonals, or density variations. Practical application in case of the four AP-stories was, for example, the support of the detection of rewrites, or in other cases the detection of similarities or dissimilarities between the same text in different languages. The principle of Dotplots is shown in Figure 46. Figure 47 shows an example of Dotplots. Tokens in the examples are characters, not words. Before reading the explanation of the figure, guess which of the three plots is a dada poem.

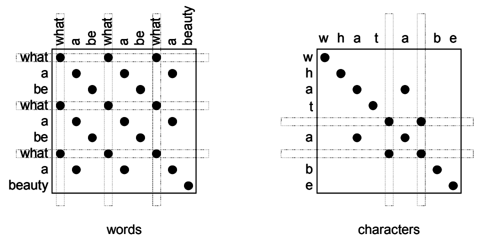


Figure 46: Principle of Dotplots according to [Church, Helfman 1993]

figures was repeated with the same result by using http://www1.bell-labs.com/user/gwills/ntts95/SEE.gif [2001-02-03] and http://www.sims.berkeley.edu/~hearst/irbook/10/seesoft.gif [2001-02-03].

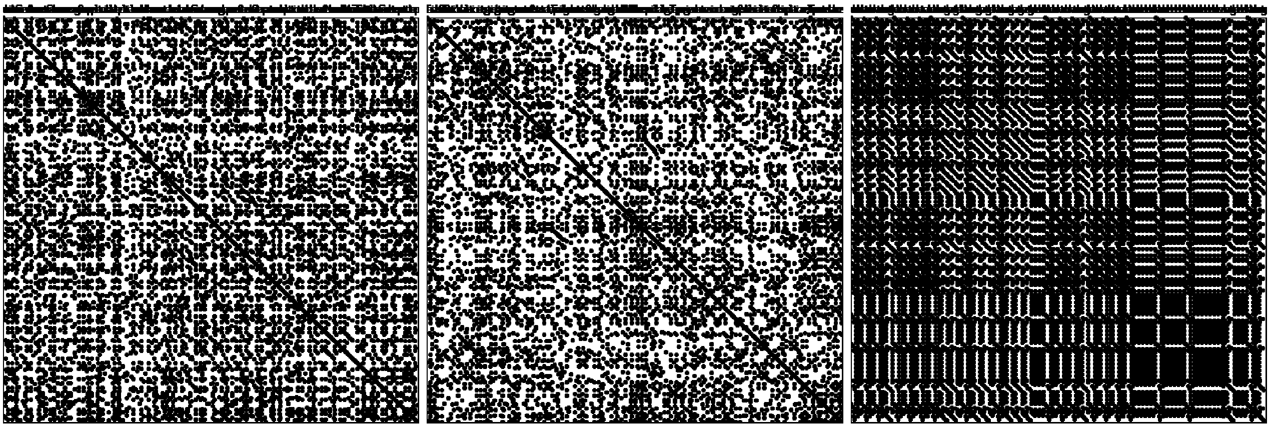


Figure 47: Examples of Dotplots[102]. From left to right: plots of the first 308 characters of the first paragraph of the Declaration of Independence of the United States of America, of the first 308 characters of the abstract from [Mann 1999], and of the 308 characters dada poem "What a be what a be what a beauty" by Kurt Schwitters.

An idea comparable to the Dotplots by [Church, Helfman 1993] has been used by [Gershon, LeVasseur, Winstead et al. 1995] for the visualization of single documents retrieved from the World Wide Web. Instead of the self-similarity principle, word correlations calculated by the proximity of any pair of words are used to produce a "Dotplot"-visualization of a document.

More famous than the already mentioned "Positive / Negative Feedback workspace" by [Veerasamy, Navathe 1995] / [Veerasamy, Hudson, Navathe 1995] is their bar-graphs view of a result set from ranked output systems. The component shows the distribution of query terms for up to the highest ranked 200 or 150[103] documents in a result set. It is used for two main reasons: to gain specific information about individual documents and to gain aggregate information about the query results in general [Veerasamy 1996] / [Veerasamy, Belkin 1996]. For each document a group of vertically stacked bars is used to show the overall relevance of the document for the query, and the contribution of every keyword or concept. A concept can be a single keyword, or a group of keywords being synonyms or other forms of the keyword. Each concept is shown in one row. The taller the bar of a concept, the higher is the contribution of this concept to the retrieval of the document. If a bar for a concept is absent, the concept in the document is absent. Figure 48 shows the principle of bar-graphs using the WebViz-example and a result set of 20 documents. The original examples of Veerasamy et al. show 70 or 150 documents. By examining the bar-graph, it can be detected that the highlighted document #6 has been ranked higher than #7, despite the fact that #6 does not contain the concept "visualization". Document #7 contains all four concepts, including a weak contribution of "visualization". The contribution of the concept "internet" is much weaker than in document #6. As aggregated information about the whole result set it can be seen in the bar-graph, that nearly all of the documents deal with "search" and "results", and many with "visualization". The concept "internet" is not well represented.

---

[102] Figure produced by using http://www.research.att.com/~jon/dotplot/try.html [2001-01-29]. The first "line" in the plot is each time the reproduction of the 308 characters string as a legend.

[103] 200 according to [Veerasamy, Hudson, Navathe 1995], 150 according to [Veerasamy 1996]. There are also some other changes in the interfaces described in the 1995 and the 1996 papers.

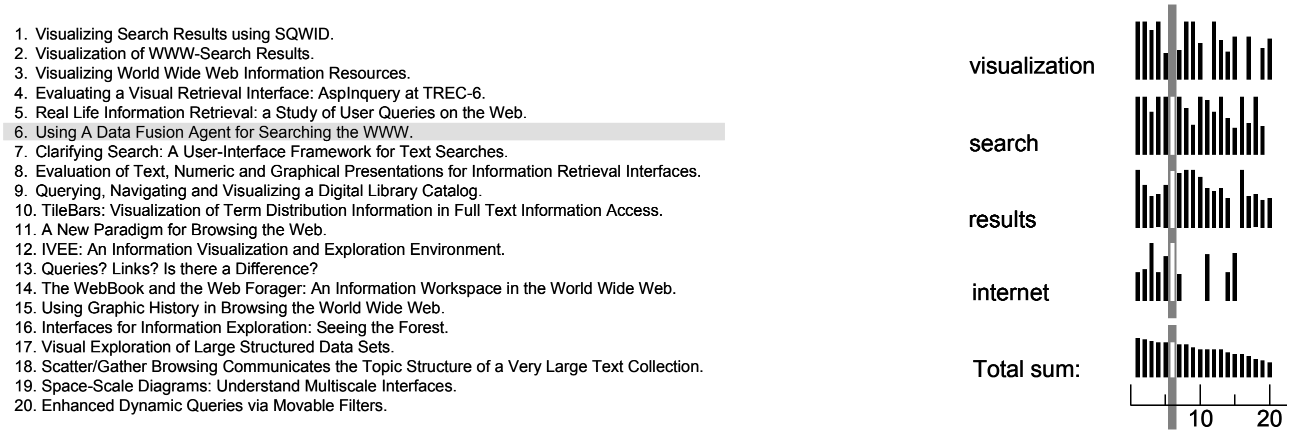


Figure 48: Principle of bar-graphs by [Veerasamy 1996] / [Veerasamy, Belkin 1996]

The stacked histograms used by [Shneiderman, Byrd, Croft 1997] in the WInquery system are a related approach. The authors also proposed a solution to show the contribution of each query term for the overall relevance of the document. Their solution, compared to the approach from Veerasamy et al., focuses more on specific information about individual documents, than aggregated information about the query results in general. The WInquery system is a redesign of the XINQUERY User-Interface done by the authors based on their proposed four-phase framework for search and eight design rules adapted from a previous edition of [Shneiderman 1998][104]. The idea of the stacked histograms has also been influenced by Hearst's tilebars. In a later publication by [Byrd 1999] the component had been named "VQRa", as Visualization of the Query in relation to individual Retrieved documents. The "a" is added because Byrd also describes a solution "b". Figure 49 shows the principle using the same query, the same ranking and the same result set as used above. Looking on the documents #6 and #7 the same results can be observed as described in the explanation of the bar-graph.

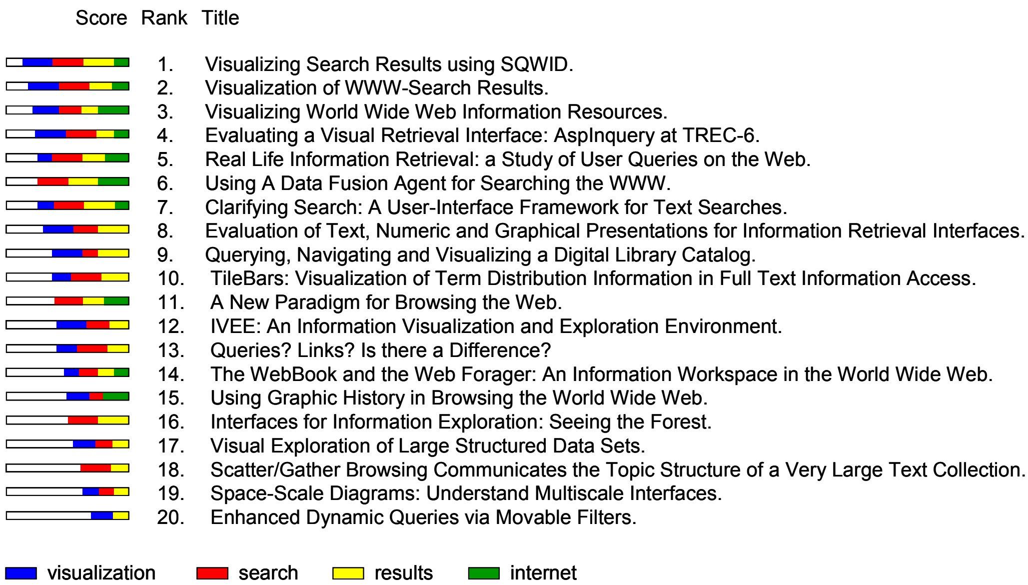


Figure 49: Principle of stacked histograms / VQRa of the WInquery system by [Shneiderman, Byrd, Croft 1997], [Byrd 1999]

[Cugini, Laskowski, Piatko 1998] also try to visualize the relevance of each term in a multiple-term query. They used in the NIRVE system flat Iconic Representations of documents showing the

[104] In fact they took the rules from the second edition 1992. The 1998-version is the third edition.

"concept profile" in form of the relevance of each concept. The Iconic Representation has been used in 3D components of the NIRVE system showing interdocument similarities such as the Document Spiral, the 3-D Axes view, or the Concept Globe [Cugini, Laskowski, Sebrechts 2000]. Being part of a visualization of interdocument similarities the Iconic Representation itself could clearly be used to visualize document attributes. Figure 50 shows the principle using the document set from the WebViz example. Looking again at documents #6 and #7 the same results can be observed as described in the explanation of the bar-graph. In the 3D visualizations of the NIRVE system the icons are on user request additionally decorated with small glyphs for attributes like document length or overall document scores. In later versions there was also an additional glyph indicating user judgments of the document (green = good, red = bad, yellow = undecided) [Cugini, Laskowski, Sebrechts 2000].

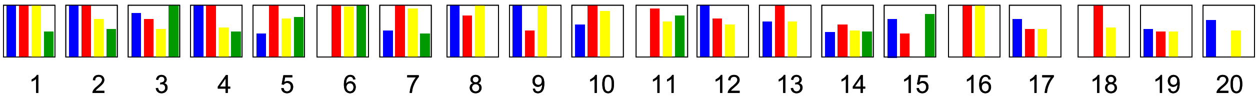


Figure 50: Principle of the Iconic Representation in the 3D Document Space of the NIRVE system by [Cugini, Laskowski, Piatko 1998]

[Grewal, Jackson, Wallis et al. 1999], [Grewal, Burden, Jackson et al. 1999], [Grewal, Jackson, Burden et al. 2000] also try to visualize the relevance of each term in a multiple-term query. In their R-Wheel (Relevance Wheel) component, each term has its own circle segment and color associated. The segment is filled in proportion to the relevance of the term. The number of circle segments corresponds to the number of keywords. Figure 51 shows the R-Wheels using the document set from the WebViz example. Looking again at documents #6 and #7 the same results can be observed as described in the explanation of the bar-graph.

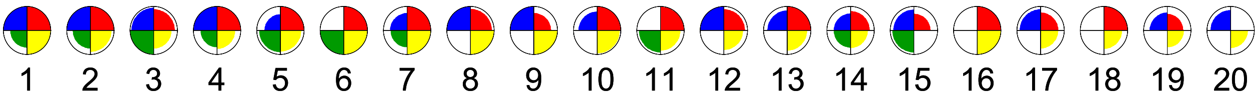


Figure 51: Principle of R-Wheels (Result Wheels) by [Grewal, Burden, Jackson et al. 1999], [Grewal, Jackson, Burden et al. 2000]

Both Veerasamy and Grewal et al. discuss a number of other ideas for the visualization of query term contribution. For different reasons in the end they prefer the solutions shown in Figure 48 and Figure 51. Figure 52 shows the principles of their additional ideas using documents #6 and #7 as examples.

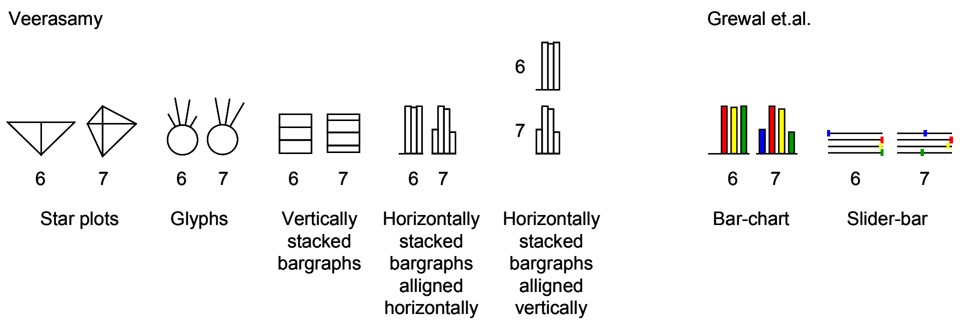


Figure 52: Additional ideas of [Veerasamy 1997] and [Grewal, Jackson, Burden et al. 2000]

In [Grewal, Jackson, Wallis et al. 1999] there is one more idea named "tepee". Basic structure is a transparent pyramid which has as many base sides as keywords displayed. Inside the tepee is a pendulum. The length of the pendulum represents the overall relevance. The pendulum is attracted to the sides of the tepee according to the relevance of the term represented by the side. Additionally the sides are shaded depending on the relevance of the terms. One or two term queries will be difficult to visualize with the tepee. A user test revealed that people had difficulties with this visualization idea.

An interesting usage of bars showing document attributes in form of Retrieval History Histograms is done by [Golovchinsky 1997] in the VOIR system. A histogram is shown above every article in a newspaper layout. The bars in a histogram and their length indicate the relevance values of the article in a series of queries during a browsing session. Additionally a number of link types are color-coded. This reflects mechanisms of the VOIR system to create queries automatically when the users follow hyperlinks or dynamically inserted anchors. An additional feature of the histogram is its usage for navigation. Clicking on a bar brings the user back to the corresponding query. Figure 53 shows the principle. Please note that the number of bars represents the number of queries in which the document has been retrieved, and not the number of keywords or concepts in a single query or the number of documents selected, like in the other bar-graphs introduced in this chapter so far.

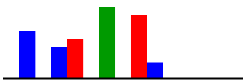


Figure 53: Principle of the Retrieval History Histogram of the VOIR system by [Golovchinsky 1997]

Probably the most famous component for the visualization of document attributes is the TileBars from [Hearst 1995]. A number of ideas listed above had been influenced by the TileBars idea. TileBars visualize the distribution of query terms in a document showing at the same also the query term frequency, query term overlapping within a single passage, and the relative length of the overall document. In the original implementation from [Hearst 1995] the documents had been split into tiles using an algorithm called TextTilling. The tiling algorithm tries to detected subtopic boundaries in a text, and splits it into tiles dealing with one subtopic. In the visualization rectangles with squares inside indicate documents. Each square or tile stands for one TextTile and one keyword or set of keywords. The row of tiles stands for the keyword or keyword set. Every vertical group of tiles represents a TextTile of the document. Besides the original TextTilling algorithm the visualization is also suitable for the usage with other text-tiling mechanisms. The darker a tile is displayed in the visualization, the more relevant is the keyword or keywords set for this TextTile of the document. Because of their natural visual hierarchy varying shades of gray instead of color have been used to show the varying grade of relevance. Nevertheless, a distinct color is attributed to every keyword or set of keywords to ease recognition when using keyword highlighting in the text display part of the TileBars interface. To give the user the possibility to prove his impressions gained from the visualization, the TileBars offer a possibility to jump to a certain part of a document by clicking on the appropriate tile. The TileBars interface has additional features like changing the presentation sequence of the documents according to distribution pattern types, or using distribution constraints of terms in the tiles to filter the result set. Figure 54 shows an example of the TileBars principle using the WebViz-example and the three documents [Mann 1999], [Hearst 1995], and [Veerasamy, Navathe 1995]. The tiling algorithm used for the example is simpler than the original TextTilling used in [Hearst 1995]. In addition a similarity search is used instead of a pure Boolean search. Otherwise [Hearst 1995] and [Veerasamy, Navathe 1995] would not have been included in the result set. The first row of tiles in each of the three documents represents the term set “visualization” OR “visualisation”, the second row “search”, the third row “results” and the fourth row “internet” OR “www”. The terms “internet” OR “www” can only be found at the beginning of the first document. In this document the term “Web” is used frequently but is not part of the term set used here as a query, and therefore is not indicated in the TileBar. In addition, it can be seen that all of the four term sets can be found in the first part of the first document. The second document contains a frequent co-occurrence of the terms “search” and “results”. The term set “visualization” OR “visualization” is not as dominant as in the first document. The last of the three documents is much shorter than the two previous ones.

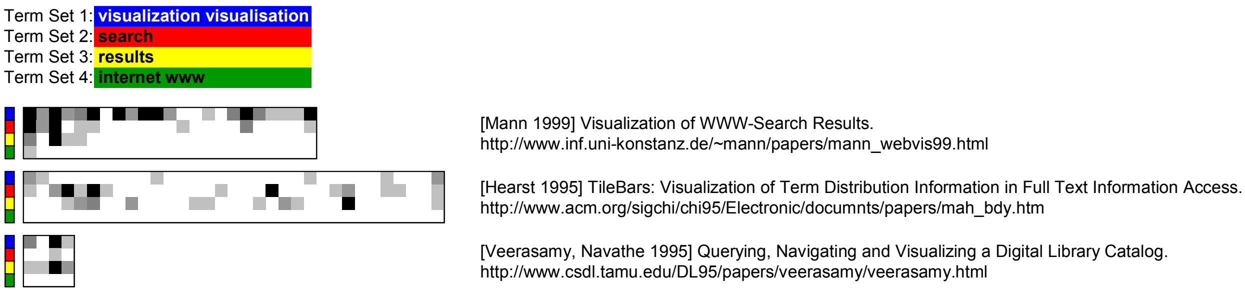


Figure 54: Principle of the TileBars by [Hearst 1995][105]

[Heo, Morse, Willms et al. 1996] modified the TileBar idea in the CASCADE system for the usage with a single document, and coupled it with a scrollbar. Also combined with the scrollbar is a component called Mural. While the TileBar shows the distribution of query terms in the document, the Mural shows the distribution of the hyperlinks. The CASCADE (Computer Augmented Support for Collaborative Authoring and Document Editing) system is a tool to support collaborative authoring of documents. In the CASCADE system, Mural and TileBars are used as intra-document tools to ease navigation through the usage of landmarks. Landmarks in the document are the links and the matches of the query terms. Figure 55 demonstrates the principle of Mural plus TileBars using a HTML-version of [Mann 1999] and the WebViz-query reduced to the maximum number of three allowed term sets in the CASCADE system.

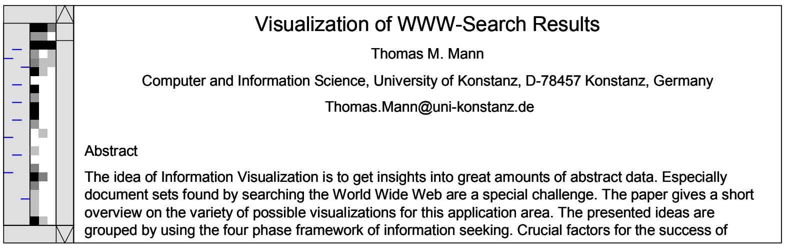


Figure 55: Principle of Mural and TileBars in the CASCADE system by [Spring, Morse, Heo 1996], [Heo, Morse, Willms et al. 1996]

Another modification of the TileBar idea is the content-displaying scrollbar VQRb of the FancyV prototype described by [Byrd 1999]. An ordinary scrollbar is modified and shows every query term hit in the document in the scrollbar pane using 3-by-3 pixel squares colored according to the color associated to the keyword. The slider of the scrollbar is white to ease recognition in the currently displayed portion of the document. Additionally the VQRb is combined with colored query term highlighting in the document itself. Figure 56 shows the principle using the WebViz-example and a one-column version of the document [Mann 1999].

---

105 The figures in [Hearst 1995] show no colors, but [Hearst 1999] Figure 10.15, which can also be retrieved from the XEROX PARC Web server in a colored version, shows the usage of colors. Hearst uses not saturated colors. They look nicer than the color palette used for the examples in this thesis.

Figure 56: Principle of VQRb in the FancyV prototype by [Byrd 1999]

The limitation of approaches with integration of elements in the scrollbar or besides it, are the available pixels of the screen. If the display is 1024x768 pixels, and about 30 to 70 vertical pixels are set aside for window borders and menus, the remaining 700 pixels are sufficient for displaying the equivalent of 233 lines taking the 3-by-3 pixels symbols from Byrd without vertical overlap. The document in Figure 56 has 295 lines. For documents containing more lines than available screen space, techniques such as overlap, distortion, or others must be used.

Like [Heo, Morse, Willms et al. 1996] and [Byrd 1999] [Dieberger, Russell 2001] also restricted the usage of TileBars to a single document. They expanded the TileBars idea from the visualization of query term hits to a more general visualization of “features”. A feature in their sense connected to a tile can be any meta information or item of interest. [Dieberger, Russell 2001] describe possibilities of displaying features such as the severity of spelling mistakes or the presence of phone numbers, email addresses, or URLs. The last example is an integration of the Mural plus TileBars from [Heo, Morse, Willms et al. 1996] into one component. [Dieberger, Russell 2001] use both orientations of TileBars: the horizontal one from [Hearst 1995], and the vertical one from [Heo, Morse, Willms et al. 1996]. Using a focus plus context approach, they introduce Context Lenses as a component for showing different types of focus information in combination with TileBars. Their basic example is the display of parts of the text of the tile, when focusing on a tile. As we will see later the same idea had been implemented before in the TileBar-inspired SegmentView of the INSYDER system, where the text of a tile is displayed as a tooltip when crossing the tile [Mann, Reiterer 2000].

Compared to the more abstract visualizations introduced above, the libViewer component [Rauber, Bina 1999], [Rauber, Bina 2000] of the SOMLib system, mentioned in the metaphors section, is radically real-world oriented. Search results from a retrieval system are shown as 3D-books by mapping metadata of documents to attributes of real-world books. Metadata such as size, language, publisher, number of times referenced, or last time referenced are mapped to attributes like size of the book displayed, color, logo, appearance, or dust. Frequently referenced documents will therefore be shown with crippled, well-thumbed binding or documents not referenced for a long time will be shown with dust on the cover. Figure 57 shows an example with different documents and their attributes.

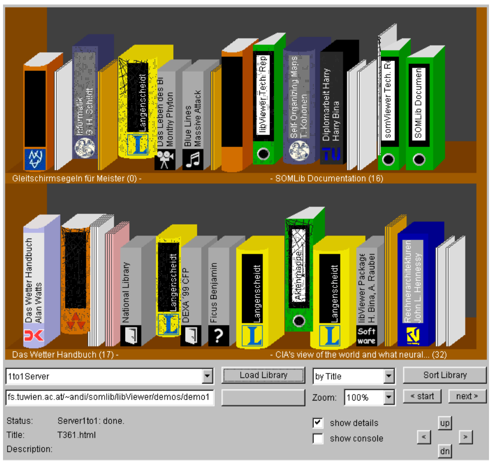


Figure 57: Example of the libViewer from the SOMLib system[106] [Rauber, Bina 1999], [Rauber, Bina 2000]

The last idea introduced in this chapter leads to the next chapter about the visualization of inter-document similarities. [Chase, D'Amore, Gershon et al. 1998] describe an Entity Relation Visualization in the NetMap system, where individual entities found in documents and their relations are encoded with color, shapes, and connecting lines. The main target of Netmap are interdocument relationships, but the authors describe also that the Entity Relation Visualization can be used for the discovery of interentity relationships. Figure 58 shows the principle according to [Chase, D'Amore, Gershon et al. 1998]. Probabely the component could also be used for entities found in a single document.

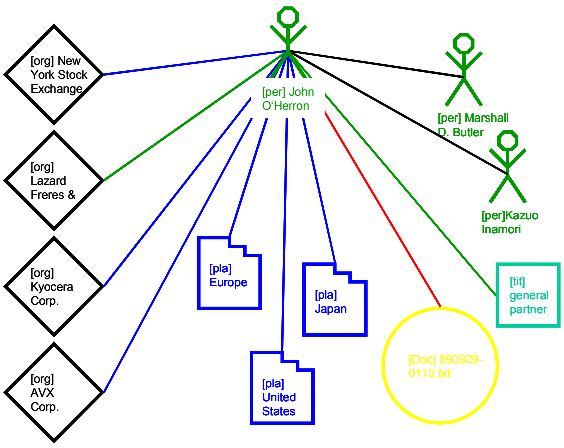


Figure 58: Principle of Entity Relation Visualization by [Chase, D'Amore, Gershon et al. 1998] Figure 3e.

This chapter introduced a number of ideas for the visualization of document attributes. Table 18 gives an overview covering the components discussed. Besides simple miniaturizations of documents or their first pages, with or without fisheye or distortion techniques, we have seen a number of visualizations of document attributes, metadata or query related information. Query related information included overall relevance, relevance per keyword or concept, and query term or concept distribution in the document. Metadata displayed included size of the document, document type, bookmark status, or visitation information. Document attributes included self-similarity patterns, hyperlink information, or the occurrence of query-independent items.

[106] Figure produced using http://student.ifs.tuwien.ac.at/~andi/libViewer/ [2001-03-02]

Component	Literature	Used in System
Thumbnail views	[Ayers, Stasko 1995]	MosaicG
	[Ginsburg, Marks, Shieber 1996]	DeckView
	[Hightower, Ring, Helfman et al. 1998]	PadPrints
	[Bederson, Hollan, Stewart et al. 1998]	Pad++ Web browser
	[Robertson, Czerwinski, Larson et al. 1998], [Czerwinski, Dumais, Robertson et al. 1999]	Data Mountain
	[Kaugars 1998]	
	[Ogden, Davis, Rice 1998]	J24
	[Amento, Hill, Terveen et al. 1999]	TopicShop
	[Cockburn, Greenburg, McKenzie et al. 1999], [Cockburn, Greenberg 1999], [Cockburn, Greenberg 1999a], [Kaasten, Greenberg 2000], [Kaasten, Greenberg 2001]	webView and other unnamed systems
“semi open view”	[Kaugars 1998]	
Relevant Extracts plus Curve of Relevance	http://www.arisem.com [2001-02-11]	DigOut4U
SeeSoft “bar view”	[Eick, Steffen, Sumner 1992], [Eick 1994], [Wills 1995]	SeeSoft
Information Mural	[Jerding, Stasko 1995], [Jerding, Stasko 1997]	
Matrix of Icons	[Nowell, France, Hix et al. 1996]	Envision
Dotplots	[Church, Helfman 1993]	
bar-graph	[Veerasamy, Navathe 1995], [Veerasamy, Hudson, Navathe 1995], [Veerasamy 1996]	Tkinq
Iconic Representation	[Cugini, Piatko, Laskowski 1997], [Cugini, Laskowski, Piatko 1998], [Cugini, Laskowski, Sebrechts 2000]	NIRVE
R-Wheels or Result Wheel	[Grewal, Jackson, Wallis et al. 1999], [Grewal, Burden, Jackson et al. 1999], [Grewal, Jackson, Burden et al. 2000]	
Retrieval History Histogram	[Golovchinsky 1997]	VOIR
TileBars	[Hearst 1995]	
	[Spring, Morse, Heo 1996], [Heo, Morse, Willms et al. 1996]	CASCADE
	[Dieberger, Russell 2001]	
Mural	[Spring, Morse, Heo 1996], [Heo, Morse, Willms et al. 1996]	CASCADE
libViewer	[Rauber, Bina 1999], [Rauber, Bina 2000]	SOMLib
Entity Relation Visualization	[Chase, D’Amore, Gershon et al. 1998]	NetMap

Table 18: Components for the visualization of document attributes

3.3.3.3. Visualization of interdocument similarities

In the last two chapters, we have already seen a number of components capable of providing information about not only the query or a single document, but also first overviews about parts of or the whole result set. Examples are the InfoCrystal from Spoerri or the bar-graphs from Veerasamy et al.. We now turn the focus explicitly to visualizations of document groups and interdocument similarities. On the set level, which means the representation of the whole set of results, it will be interesting to get an overview. Are there any trends, clusters, or hot spots? Do the suggestions made by the system in response to the query seem to satisfying the information needs at all? The last two chapters had only been spotlights on the field of ideas how to map data tables to visual structures. This chapter will be even more incomplete in relationship to a full discussion of all components that can be found in the literature. The number of visualization ideas for document sets is considerably higher than the number of ideas for the two areas discussed so far. Therefore, the discussion about visualization ideas for document sets is separated into two parts, and additionally focuses mainly on 2D-visualizations. Whereas this chapter discusses the visualization of interdocument similarities, the next chapter deals with the visualization of interdocument connections. So far the majority of ideas discussed have been 2D-visualizations. An exception has been the chapter about metaphors where 3D-ideas played an important role. In the area of visualizations of document sets 3D-approaches are found quite frequently. Regardless 3D-visualization ideas have been excluded relatively early in the process of identifying candidates to be included into the INSYDER system. Today's hardware no longer poses a restriction, however, navigation in 3D-space with standard input devices such as keyboards and conventional mice still create a barrier. In addition, a number of authors like [Nielsen 1998] report problems with 3D-approaches. Since the typical technical environment of the target user group are standard PCs and input devices, 3D-approaches were not included in the list of potential components for the INSYDER system. Therefore, the following overview of the variety of approaches for the visualization of document sets will focus on 2D-components.

The decision to classify a visualization component as suitable for the visualization of queries, or document attributes, or interdocument similarities is not always easy, because a number of components can be used for different purposes. Bargraphs are a good example for this. The bar-graphs from Veerasamy et al. reveal information about documents' attributes in relation to the query but also provide an overview of the whole result set or parts of it. The same is true for the stacked histograms / VQRa of the WInquery system. A predessor of them is the Bargraph view of the XINQUERY system [Shneiderman, Byrd, Croft 1997]. One bar represents a document, showing with its length the relevance score value of the document. Figure 59 shows the principle using the 20 document result set of the WebViz-example. The currently displayed document (in example #1) is marked using a different color. Besides showing the score and position of a document in the result set, the overview of the result set and the visualization of interdocument similarities for this two dimensions seems to be dominant, leading to the decision to categorize the XINQUERY bar-graph in this chapter. Dominant function by subjective classification by the author is also used for all the other components discussed.

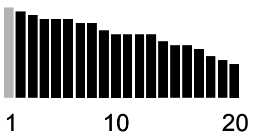


Figure 59: Principle of the Bargraph in the XINQUERY system according to [Shneiderman, Byrd, Croft 1997]

In the FISH (Forager for the Information Super Highway) component of the Starfish system, [Mitchell, Day, Hirschman 1995] use rectangles to represent documents and their relevance. In some configurations these rectangles look like bars. Inspired by the Tree-Map approach from [Shneiderman 1992] attributes of documents from a result set of a multi-source WAIS query are encoded by space, order, and color. Figure 60 shows the principle using the WebViz-example. In the example on the left side the Tree-Map approach is omitted by showing all 20 documents in one group. Relevance is mapped to size, position, and color of the rectangle representing the document. The size of the rectangle is proportional to the relevance of the document. Higher relevant documents use lighter colors and the documents are ordered by relevance. Tree-maps had been designed to represent hierarchies. In the right part of Figure 60 a hierarchy is used by showing the documents grouped by domain of the server from where they have been downloaded. The size of the rectangle representing a domain is determined by the sum of the relevance of the documents in the set coming from this domain. The mapping of the document relevance is done in the same way as on the left side of the figure, except in regards to the hierarchy when ordering the documents and using a variable width of the document representations.

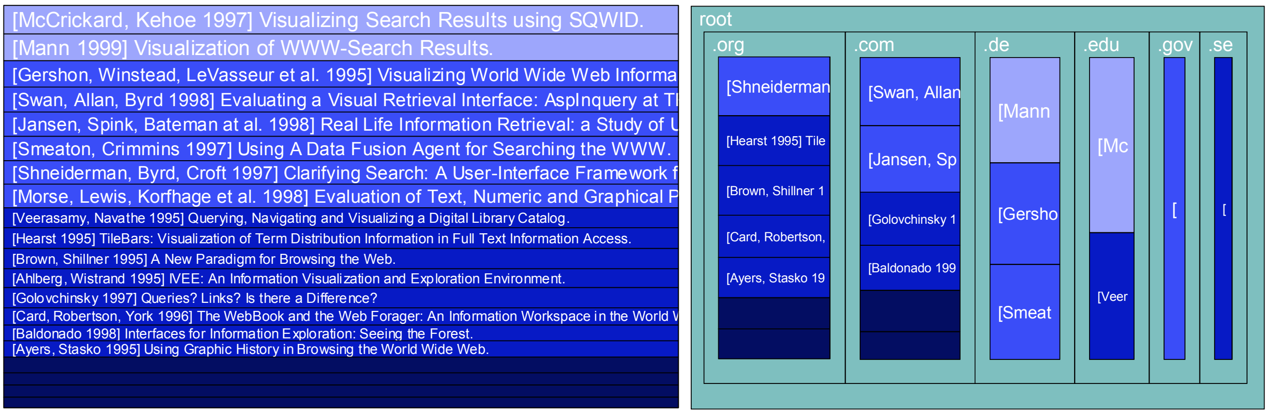


Figure 60: Principle of the FISH component from the Starfish system of [Mitchell, Day, Hirschman 1995]

Treemaps are used in a number of systems to represent multi-step hierarchies. Hierarchies are normally visualizations of interdocument connections, and not of interdocument similarities such as discussed in this chapter. In a number of cases, however, the usage of treemaps is more oriented to the later case. The Information Navigator [Au, Carey, Sewraz et al. 2000], [Carey 2000], [Carey, Kriwaczek, Rüger 2000] is another example for using Treemaps to visualize interdocument similarities from Web search results. The left part of Figure 61 shows the initial view of a document set retrieved using the query from the WebViz-example in a database containing about 550,000 documents from the TREC CDs vol4 and vol5. The hierarchy is determined by clustering the documents using statistical information about the terms in the documents. Instead of the documents themselves only clusters and super clusters are shown. The term list on the right side of the Treemap shows the statistically collected terms from the documents in this cluster in descending order of occurrence. The right side of Figure 61 shows a re-clustered subset of the complete result set after selecting one of clusters.

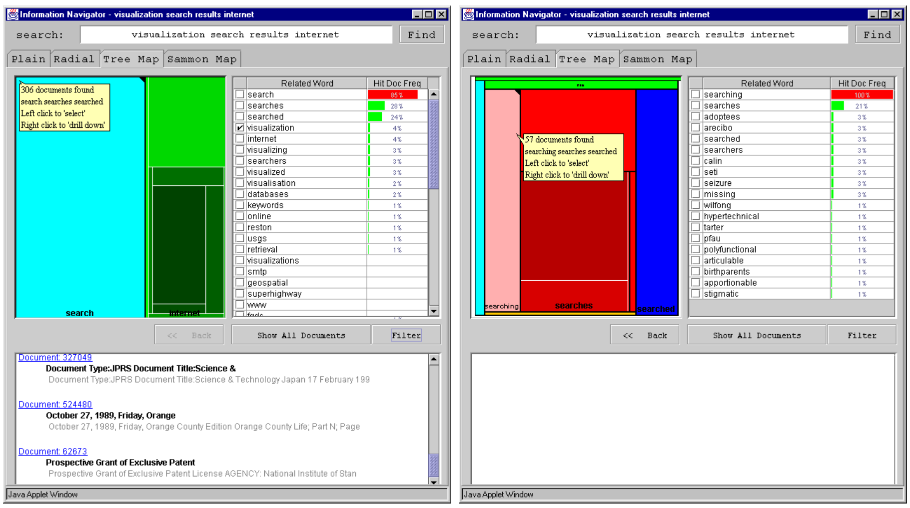


Figure 61: Example of the Treemap View from the Information Navigator [Au, Carey, Sewraz et al. 2000][107]

[Chimera 1992] works with Value Bars. Quantifiable attributes are mapped each to a separate bar next to the scrollbar of a list. The idea is to show an attribute distribution overview for important

107 Figures produced using http://rowan.doc.ic.ac.uk:8000/InfoNavigator/provodnik.html [2001-03-04]

items in different dimensions. List items are represented by differing height regions in different Value Bars. If space is restricted, only the topmost weighted items of a dimension are displayed in the corresponding bar. Figure 62 shows the principle using the 20 document result set of the WebViz-example. The list shows the authors of the documents in alphabetic order. The bar labeled “R” shows the Relevance of the documents, and the bar labeled “Y” shows their youth or document recency. The taller the region in the bar is for a document, the higher its value in this dimension. For a selected highlighted item, its corresponding regions are also highlighted in the Value Bars. In the example, this is done for the third document. Like the scrollbar, the Value Bars have indicators that show which portion of the dataset is visible. In the example, it is 50% of the dataset. The “visibility markers” move synchronized, but change their size according to the different distributions of the attributes.

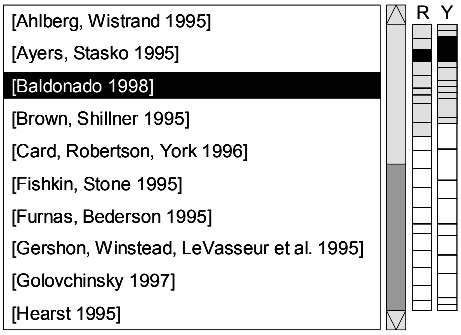


Figure 62: Principle of the Value Bars by [Chimera 1992]

When geographical references are part of the data, a number of systems use map displays to map data tables to visual structures. Examples include the Dynamic Homefinder [Williamson, Shneiderman 1992] shown in Figure 63, or the already mentioned Magic Lens Filters example from [Fishkin, Stone 1995] shown on the left side of Figure 27 on page 68.

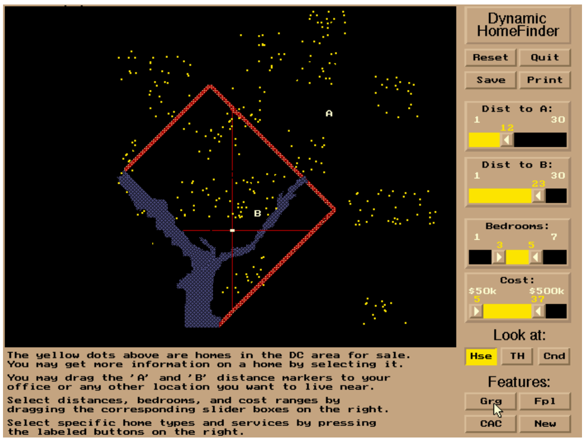


Figure 63: Dynamic Homefinder[108] showing selected houses in the area of Washington D.C.

For the visualization of document sets geographical references may not play an important role, but “map” displays of document sets definitely do. Many systems produce some sort of maps showing interdocument similarities or interdocument connections. Instead of displaying the geographical location, in most cases other mechanisms are used to calculate positions of document representations in virtual 2D, 2.5D or 3D document spaces. The mechanisms used to calculate the low-dimensional positions of documents from n-dimensional data tables are:

[108] Figure produced by using http://www.cs.umd.edu/hcil/pubs/dq-home.zip [2001-02-22]

- Dimensionality reductions by selecting dimensions and mapping their values to axes (X and Y, sometimes also Z)
- Dimensionality reductions by compressing an n-dimensional space to a low dimensional space, while taking into consideration all or nearly all dimensions.
- Dimensionality reduction by using a number of reference points for positioning documents in a low dimensional space.

Force or spring model ideas to calculate positions for the last two possibilities are a science in themselves. Please see for example [Wise 1999] for a discussion.

A famous example for the selection of dimensions and mapping of values to axes to produce a starfield[109] or scatterplot display is the FilmFinder [Ahlberg, Shneiderman 1994], [Jog, Shneiderman 1995] as a logical successor of the Dynamic Homefinder. Instead of using a geographical area with mapping longitude and latitude on X- and Y-axes, values like time or popularity of films are displayed. The starfield idea of the FilmFinder has also been implemented in the systems IVEE [Ahlberg, Wistrand 1995], [Wistrand 1995] and Spotfire Pro [Spotfire 2001]. Figure 64 shows an example using a Spotfire Pro Demo with a film database. The time when the film was created is mapped on the X-axis, the popularity on the Y-axis, and the genre on the color.

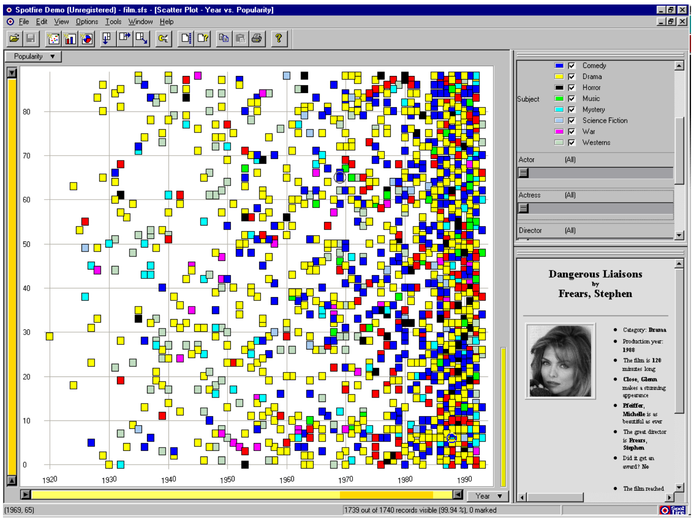


Figure 64: Principle of the FilmFinder demonstrated by using Spotfire Pro 3.3.4 Demo[110]

Figure 65 shows the transfer of the starfield or scatterplot principle to the visualization of search results using the 20 document result set of the WebViz-example. The X-axis shows the relevance of the document for the concept “results” is shown, on the Y-axis the relevance for the concept “visualization”. Additionally the summed up relevance for both concepts is mapped on the size of the icon. Alternatively, the size of the icon could indicate the overall relevance.

[109] [Ahlberg, Wistrand 1995] define starfields as interactive scatterplot with additional features for zooming, panning, details-on-demand, etc.

[110] Download from http://www.spotfire.com/down_00.htm [1999-11-25]

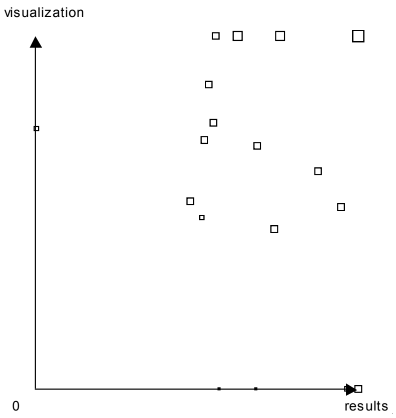


Figure 65: Scatterplot with two axes

The Three-Keyword Axes Display used by [Cugini, Piatko, Laskowski 1997] in the NIRVE system is a 3D-version of the same idea. The NIRVE (National Institute of Standards and Technology Information Retrieval Visualization Engine) system is an advanced visual interface for the PRISE statistical text retrieval system. In [Cugini, Piatko, Laskowski 1997] the authors talk about an "advanced visual interface", in later publications the name NIRVE is used. A number of components also undergo some name changes. The Three-Keyword Axes Display had later been called 3D-Axes [Cugini, Laskowski, Sebrechts 2000]. In the component document icons are positioned in a three-dimensional scatterplot based on keyword strength statistics. The left side of Figure 66 shows an example of the early Three-Keyword Axes Display using the four-keyword query "retir", "commun", "trens", and "develop". The first three keywords are mapped to the axes. If the query contains more than three keywords, the user has the possibility to assign any subset of keywords to each axis. Therefore, a separate keyword window is used with a column of checkboxes for the X-, Y-, and Z-axes shown in the upper left corner. The principle of the keyword window is similar to the principle of the Keyword-Concept Matrix or Concept Control from the same authors shown in Figure 33 on page 72. In the document space of the Three-Keyword Axes Display / 3-D Axes the documents themselves are shown with the Iconic Representation explained on page 83. The length of an extra bar outside the square indicates the overall relevance of a document. The right side of Figure 66 shows an example of the later 3-D Axes variant.

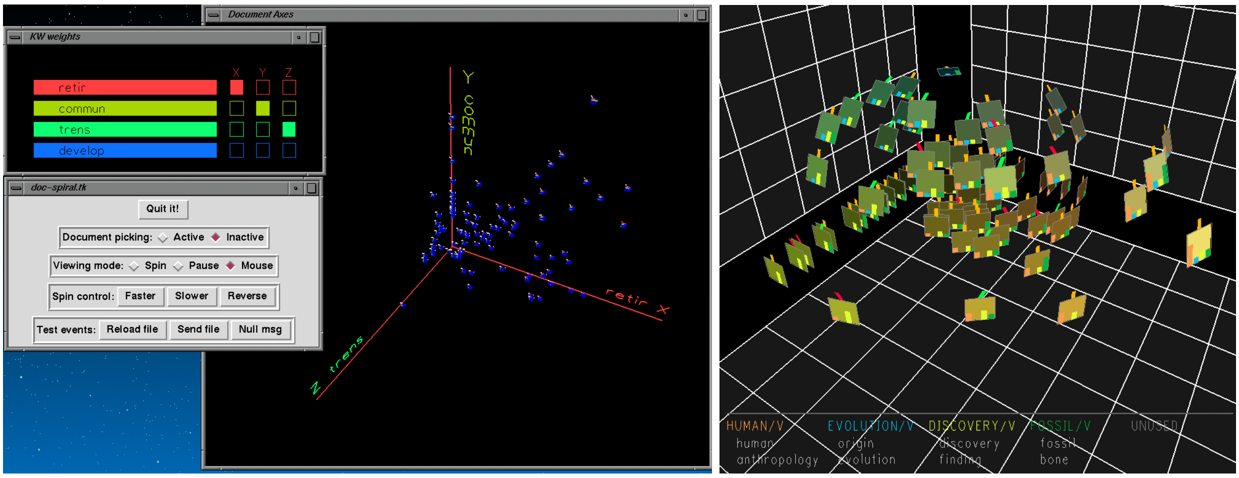


Figure 66: Three-Keyword Axes Display (left figure), 3-D Axes (right figure) from the NIRVE system. Courtesy of NIST John V. Cugini[111]

[111] Download from http://www.itl.nist.gov/iaui/vvrg/cugini/uicd/gallery/axes.gif [2001-02-26], and http://www.itl.nist.gov/ iaui/vvrg/cugini/uicd/gallery/ax3d-detail.gif [2001-02-26]

Besides quantitative variables scatterplots are also used for the visualization of ordinal or nominal data. The Envision system, with its Matrix of Icons [Nowell, France, Hix et al. 1996] is a prominent representative for this type of scatterplots. It has already been mentioned on page 80. Figure 67 shows the principle using the 20 document result set of the WebViz-example. The X-axis shows the year of publication, the Y-axis the conference name. The overall relevance is mapped to color and icon size. The item type is mapped to icon type. The number displayed below the icon is the number of the document in the result set.

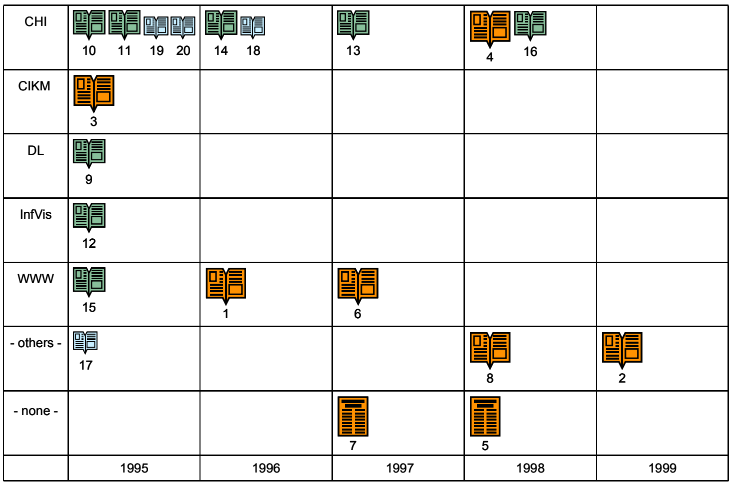


Figure 67: Principle of the Matrix of Icons of the Envision system according to [Nowell, France, Hix et al. 1996]

The usage of scatterplots in the earlier mentioned VOIR system by [Golovchinsky 1997], [Golovchinsky, Chignell 1997] is also interesting. Instead of static document attributes or query-specific dimensions, the so called "global overview" of the VOIR system shows in a scatterplot an overview of the documents retrieved during a complete browsing session. The X-axis shows the average score of a retrieved document, the cumulative score is displayed on the Y-axis. The idea is that frequently retrieved, highly relevant documents can be found in the upper right corner of the scatterplot.

Returning to the initial example of geographical references, the mapping of longitude and latitude has not only been used in the famous Dynamic Homefinder or similar systems, but also for the visualization of Web search results. One of the components in the VisageWeb system [Higgins, Lucas, Senn 1999] shows thumbnails of Web pages from a result set scattered on a world map. Using a Web service host names or IP addresses are mapped to latitude and longitude. The coordinates then are used to display the thumbnails on the map. Figure 68 shows the principle using the 20 document result set of the WebViz-example. The location of the servers from where the documents have been downloaded is not always identical with the location of the authors of the documents. A number of servers host more than one document or their locations are very close to others. Therefore, the number of displayed thumbnails in the example demonstrating the principle is significantly lower, than the number of documents in the result set.

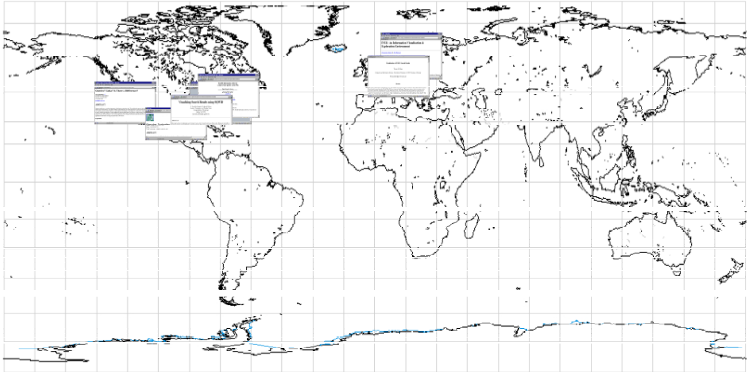


Figure 68: Principle[112] of the map display of the VisageWeb system according to [Higgins, Lucas, Senn 1999]

Besides the geographical scatterplot in the VisageWeb system scatterplots can also be used to display other dimensions of the result set. One example of [Higgins, Lucas, Senn 1999] is to display the rank for documents from one search engine on the X-axis and from another one the Y-axis.

Looking similar like the 2D scatterplots or starfields discussed so far are the Galaxies developed in the MVAB project and used in the SPIRE system [Wise, Thomas, Pennock et al. 1995], [Wise 1999]. Galaxies show cluster and document interrelatedness by reducing a high dimensional vectorspace representation of documents and clusters to a 2D scatterplot of 'docupoints'. The idea is that documents closely related to each other will cluster together in a group, while unrelated documents will be separated by spaces. The first Galaxies visualization in 1994 used in the high-dimensional part a 200,000 units long binary vector. Distances shown in the 2D projection should preserve distance relations between documents in the high dimensional space. Documents are shown as blue-green 'docustars' and cluster centroids are shown in orange on a black background, looking like a night sky. [Wise 1999] draws parallels between the FilmFinder and Galaxies: "*The original Galaxies visualization was essentially a 'starfield' of documents in a type of display seen previously in visualizations like 'Filmfinder' (Ahlberg & Schneiderman, l994) and IVEE (Ahlberg & Wistrand, l995) and now commercialized in a product called 'Spotfire.'*" The type of display is indeed the same. Document representations are positioned in a 2D space. The fundamental difference is that in the case of the FilmFinder two dimensions of an n-dimensional space are selected to be displayed on the X/Y-axes, whereas in the case of the Galaxies an n-dimensional space is reduced to a two-dimensional space by considering all dimensions.

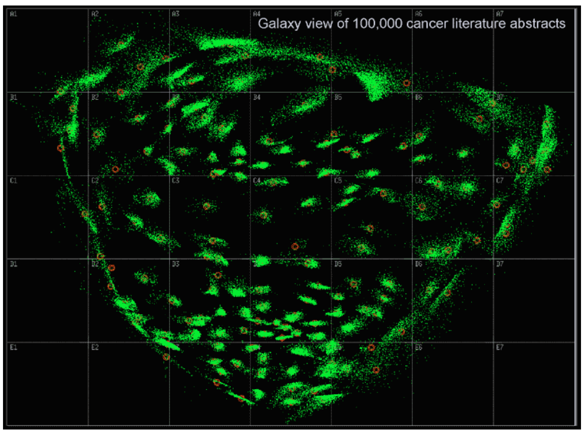


Figure 69: Galaxy view from the SPIRE system. Courtesy of Pacific Northwest National Laboratory[113]

[112] Figure produced by using thumbnail views of the HTML versions of the documents, http://cello.cs.uiuc.edu/cgi-bin/slamm/ip2ll/ [2001-02-24] to find the locations of the servers and http://mapweb.parc.xerox.com/map/ [2001-02-24] to produce the map.

A special form of scatterplot is the Galaxy of News by [Rennison 1994]. A document space, processed by a relationship construction engine, is visualized in different panes and different levels of details. Starting with a scattered keyword overview the system displays depending on user action more and more details like headlines of articles or the body of articles. Galaxy of News relies heavily on interaction and animation.

A third group of components for the visualization of interdocument similarities use reference points to position document representations in virtual document spaces. In the VIBE system [Korfhage 1991] document icons are displayed in a virtual 2D-document-space. Reference points or Points of Interest (POIs) form a coordinate system for positioning document icons. The documents are attracted by the reference points according to the relevance for the individual reference points. Figure 70 shows the principle using the 20 document result set of the WebViz-example. The four concepts displayed as circles are used as reference points. The documents are displayed as squares. The VIBE system originally used rectangles. In the example, the size of the squares is determined by the overall relevance of the document. The explanations are taken logical from [Korfhage 1991]. The two documents with explanations on the dotted line between "visualization and "results" show some of the problems with the positioning of documents in a 2D space between POIs. The position alone makes it sometimes hard to determine which POIs are concerned. Part of the idea is therefore the possibility to add or delete and move POIs around to see which documents are influenced from which reference points. The basic idea to display document icons in space between reference points has also been adapted by a number of other systems including WebVIBE [Morse, Lewis 1997], or the Radial visualization of the Information Navigator [Au, Carey, Sewraz et al. 2000] as 2D-implementations, and VR-VIBE [Benford, Snowdon, Greenhalgh et al. 1995] or the Relevance Sphere of the LyberWorld system [Hemmje 1993a], [Hemmje, Kunkel, Willett 1994] as 3D implementations. WebVIBE[114] is a simplified Java Version of VIBE using a magnet metaphor for the reference points.

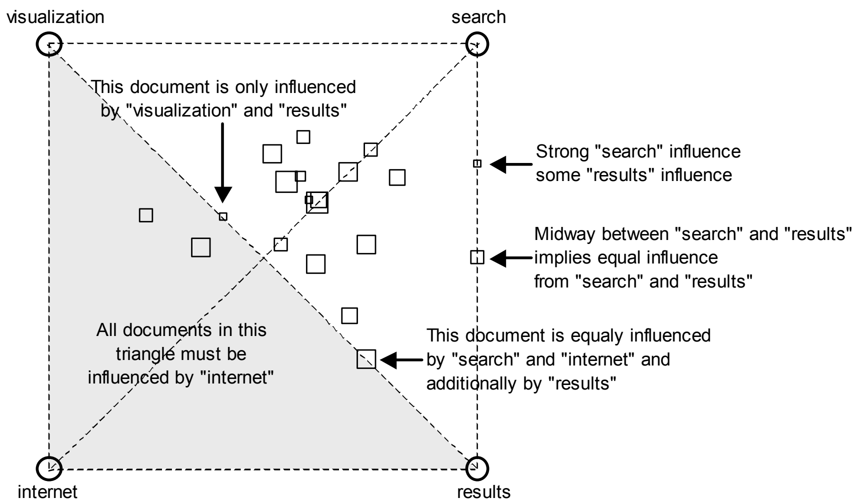


Figure 70: Principle of the reference points - documents display of the VIBE system and explanations according to [Korfhage 1991] page 138

[113] Download from http://showcase.pnl.gov/showcase/medialib.nsf/by+id/APOO-4SH26G?opendocument [2001-02-23]

[114] Web-version available at http://www2.sis.pitt.edu/~webvibe/ [2001-02-18]. Document set display not working during the preparation of this thesis. Tested with two different PCs and four different browsers / browser versions.

[McCrickard, Kehoe 1997] use in their SQWID (Search Query Weighted Information Display) system also a reference point model for the visualization of search results. There are always three reference points displayed. They are displayed as red, green, and blue rectangles. Most of the times, the reference points are not identical with the query terms. They are automatically derived from the most important terms in the titles of the results returned from AltaVista, which is used as the underlying search engine. Which terms are used in the display can be selected by the user from a list, but it must always be three. Per default, documents are shown grouped by site. This can be seen in the left part of Figure 71. Selected nodes or all nodes can be exploded by the user to show individual documents. The later is done in the right side of Figure 71. A tri-colored block is used in both modes to represent each page. The color intensities of the three sub-blocks (red, green, blue) indicate the rating of that particular page for each reference point. When pages are grouped to sites the site representation shows all tri-colored blocks for the pages from this site. In the example the sites “ai.bpa.arizona.edu” and “198.49.220.90” both contain two pages each, all others only one.

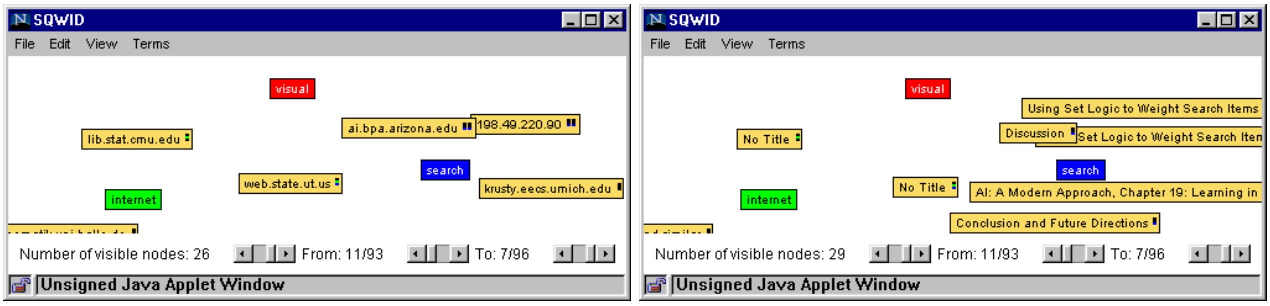


Figure 71: Example of the reference points - hosts/documents display of the SQWID[115] system [McCrickard, Kehoe 1997]

The Radial visualization of the Information Navigator [Au, Carey, Sewraz et al. 2000], [Carey 2000], [Carey, Kriwaczek, Rüger 2000] is another example for a 2D visualization using reference points. The part on the left in Figure 72 shows the initial view of the same document set as used in Figure 61 on page 90 for the Treemap visualization from the same system. The radial visualization also uses statistically collected terms from the result set to differentiate the documents. Per default the twelve highest ranked terms are displayed in the circle. The other two parts of Figure 72 show some of the interactive features of the component. As explained above when working with reference points it is sometimes difficult to find out by which POIs a document is influenced. The Radial visualization deals with this problem by providing a feature that highlights the corresponding reference points, when clicking on a document. The effect is shown in the middle of the figure, where a selected document is influenced by the terms “internet”, “cyberspace”, “compuserve”, and “ctx”. A similar feature highlights all documents influenced by a POI when clicking on the reference point. This can be seen on the right side in Figure 72, where the reference point “search” is selected. Additional features are a tooltip showing the document title when crossing a document representation with the mouse and possibilities to select terms to be used as reference points and to change their positions. The right side in Figure 72 shows an example where, instead of the twelve initial terms, the terms “www”, “internet”, “visualization”, “visualisation” and “search” have been chosen and spatially arranged.

115 Figures produced by using http://www.cc.gatech.edu/grads/m/Scott.McCrickard/sqwid/ [2001-03-16]

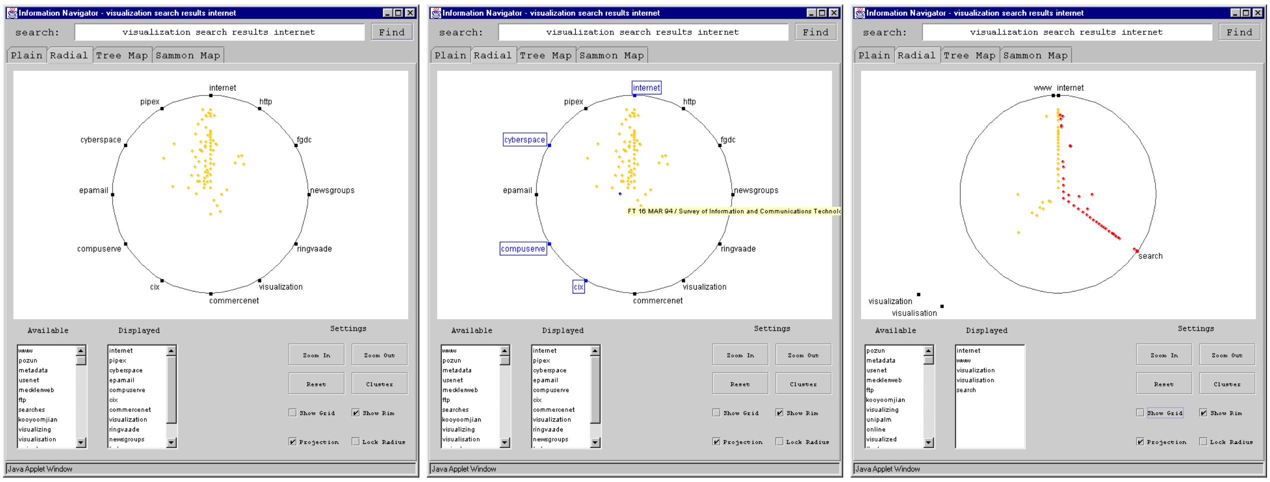


Figure 72: Example of the Radial visualization from the Information Navigator [Au, Carey, Sewraz et al. 2000], [Carey, Kriwaczek, Rüger 2000][116]

The difference between a scatterplot mapping values to axes and the usage of reference points is easy to understand when using the keywords as Points of Interest for which the relevance values have been mapped to the axes in Figure 65 on page 93. Figure 73 shows the result. All documents are positioned on a line between the two POIs instead of being scattered on the 2D pane. Additionally a comparison with Figure 70 between the use of four or two reference points shows the repositioning of the two documents marked on the line.

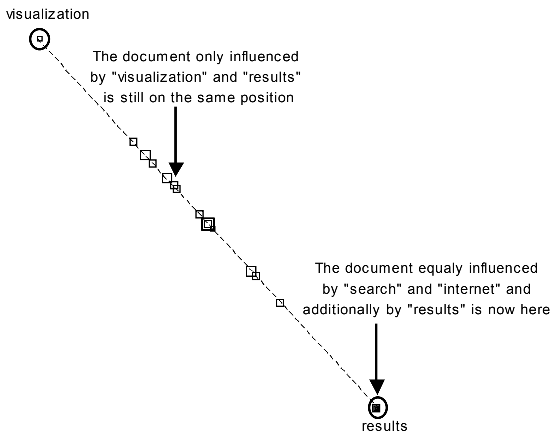


Figure 73: Document space with two reference points

Besides scatterplots populating maps based on real geographic attributes, there exist a number of approaches using the scatterplot plus landscape metaphor to create artificial 2D or 3D maps or landscapes of document spaces. A number of systems using this metaphor have already been mentioned on page 54. Examples of systems using this type of components are Bead [Chalmers, Chitson 1992], Harmony [Andrews 1995], Vineta [Krohn 1995], or SPIRE [Wise, Thomas, Pennock et al. 1995]. The difference between the visualization of interdocument similarities and interdocument connections is very small in this type of components. Early versions of Bead [Chalmers, Chitson 1992], [Chalmers 1993], for example, show only a landscape of documents. Later versions [Chalmers 1995], [Chalmers 1996] show also connections between the documents. The same can be said for a number of other starfield of scatterplot visualizations. The discussion of dual-use components in this thesis focuses on the aspects of interdocument similarities.

[116] Figures produced using http://rowan.doc.ic.ac.uk:8000/InfoNavigator/provodnik.html [2001-03-04]

The ThemeScapes of the SPIRE system [Wise, Thomas, Pennock et al. 1995] are based on the same projection of a document space on a 2D pane than the Galaxies described on page 95 [Wise 1999]. A 3D surface then is constructed by "*successively layering computed contributions of recovered theme terms over underlying document positions.*" [Wise 1999]. The idea is to visualize the primary themes of a document collection and their relative prevalence in the corpus. Dominant themes are shown as mountains. In addition to the 3D landscape a 2D map view of the landscape is provided. ThemeScape[117] has later been renamed ThemeView. Figure 74 shows an example. The fundamental difference between the landscapes of Bead, Harmony or Vineta and SPIRE are that in Bead, Harmony, and Vineta the landscape is constructed by displaying the document representations itself, whereas in the ThemeScape component of the SPIRE system the surface is formed by the terms of the documents.

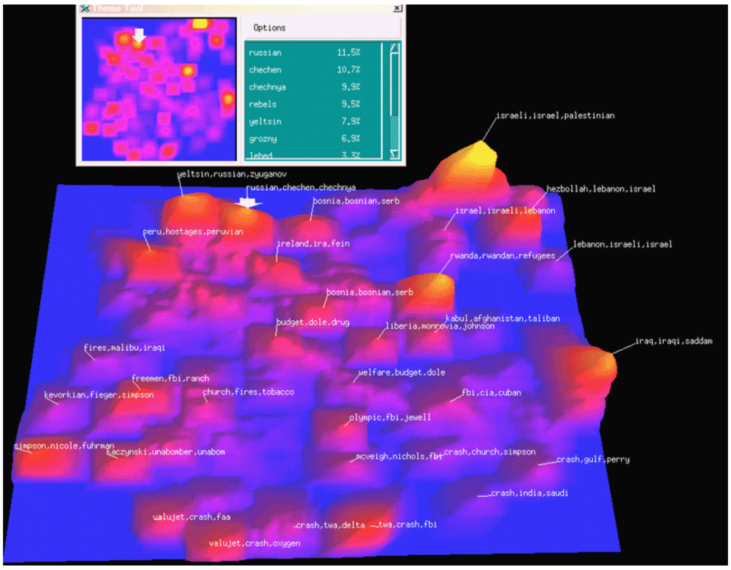


Figure 74: ThemeView from the SPIRE system. Courtesy of Pacific Northwest National Laboratory[118]

A number of authors experimented with neural networks and unsupervised learning algorithms in the form of a Self-Organizing Map (SOM) to produce nonlinear projections from high-dimensional data spaces to a two dimensional grid. Examples dealing with document sets are the Self-Organizing Semantic Map by [Lin, Soergel, Marchionini 1991], the ET-map [Chen, Schuffels, Orwig 1996] and the Adaptive SOM from the University of Arizona, the Visual SiteMap from Lin, the WEBSOM system [Kohonen 1998], or the libViewer of the SOMLib system [Rauber, Bina 1999], [Rauber, Bina 2000]. Taking into account that the principles used to map the raw data to data tables are not very different from each other, the mapping on visual structures leads to quite different views. The principle of a SOM is shown in Figure 75 using an example taken from [Lin, Soergel, Marchionini 1991]. 140 documents about Artificial Intelligence from the LISA database are mapped to a 10x14 grid. The numbers indicate the numbers of documents mapped to this cell. The labels provide hints about the content in the labeled area.

---

[117] The name ThemeScape had been trademarked by Cartia Inc. (formerly ThemeMedia)

[118] Download from http://showcase.pnl.gov/showcase/medialib.nsf/by+id/APOO-4SGVP4?opendocument [2001-03-02]

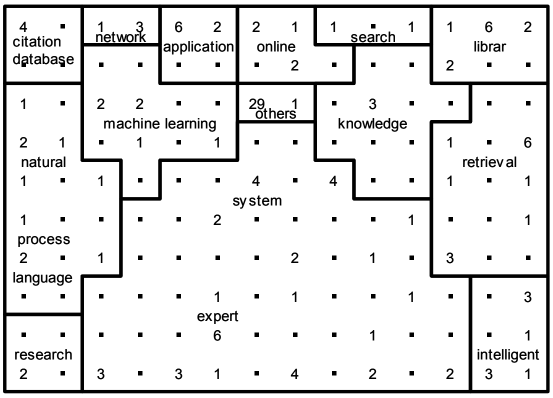


Figure 75: Principle of the self-organizing semantic map according to [Lin, Soergel, Marchionini 1991]

Figure 76 shows examples of SOMs using the 20 document result set of the WebViz-example. The left side shows the documents mapped to a 5x5 grid using the automatically extracted most used noun phrases of the document set. Dominant clusters are “user”, “users”, and “introduction”. Users seem to be important for the authors and scientific papers seem to have often an introduction. The occurrence of keywords is in the system used to produce the maps influenced by a mechanism that automatically groups nouns to phrases like “information visualization” or “query results”. Non-grouped terms like “user” or “introduction” may therefore be more dominant than in a case without using a grouping function. The right side shows a 5x5 map produced by manually selecting a number of noun phrases. The system is very sensitive to the selection of noun phrases. Adding for example the additional term “visualization” when generating the second map causes the system to produce only one single cluster labeled “visualization”. This sensibility may be caused by the small result set of 20 documents.

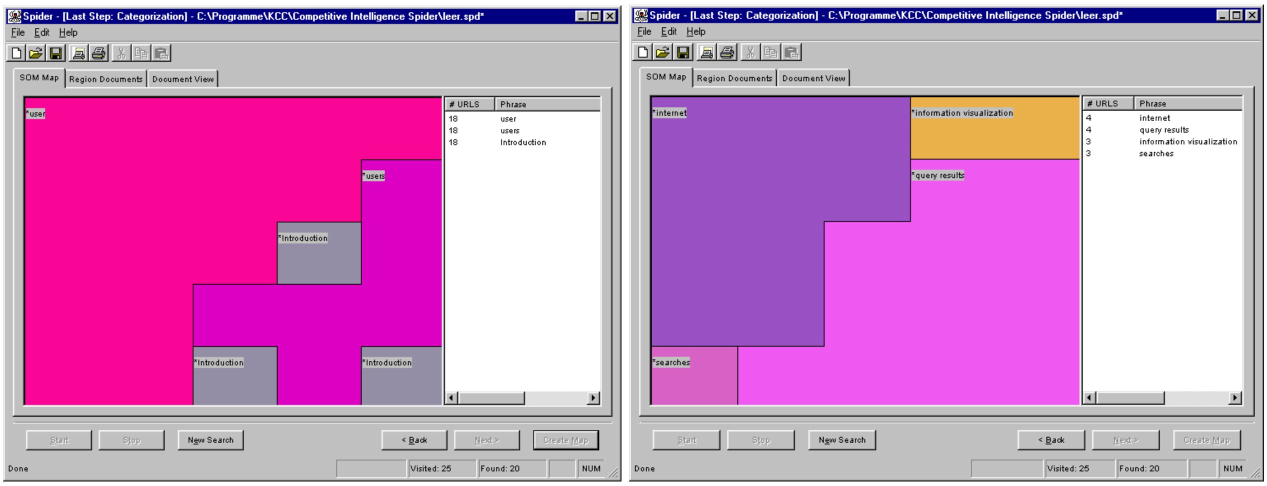


Figure 76: SOMs of the 20 document result set of the WebViz-example[119]

Figure 77 shows an example of a SOM produced by using the WebViz-example query and the Adaptive SOM Web service from the University of Arizona that transfers entered queries to AltaVista and then generates a SOM used to display the returned result set.

[119] Figures produced using a trial version of the Competitive Intelligence Spider (CI Spider) V 1.2.2 from the University of Arizona and Knowledge Computing Corporation. Download from http://ai.bpa.arizona.edu/go/download/cispider/CISpider.html [2001-03-02]. Stop word list was empty when producing the SOM.

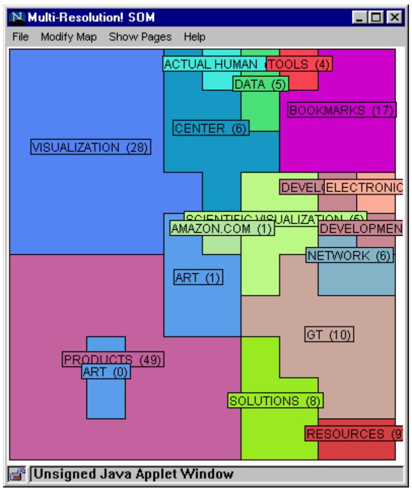


Figure 77: SOM of the AltaVista query "visualization search results internet"[120]

Figure 78 shows a SOM example from the SiteMap system. The figure on the left shows the initial view of the map. In the center, the number of displayed document points is increased by changing a threshold value with a slider. Additionally a group of documents has been selected by the mouse in the right part of the map. The titles are displayed in the "selected items" field. The figure on the right shows the same SOM with a changed threshold value for the number of displayed labels.

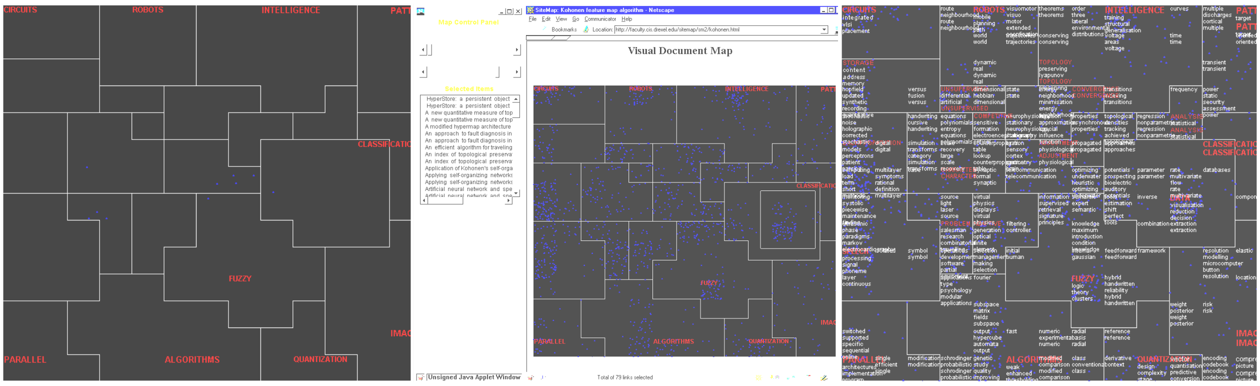


Figure 78: SOM example from the Visual SiteMap system[121]

Figure 79 shows an example from the WEBSOM system using a map containing some 20,000 news bulletins in Finnish from the year 1997. Dark areas indicate major differences between the units and therefore possible cluster borders. The map includes search and zoom features demonstrated in the three parts of the figure. The figure on the left shows the result of a search for the word "bse" in the document collection. The blue circle at the label „creutzfeldt" indicates the area containing documents related to the query. A second search on "bill clinton" is shown in the center. There are number of clusters found above the label "albright". A drill-down by clicking at that area brings up the figure on the right, where sub-cluster labels are shown. One of them is labeled "clinton". A further drill-down by clicking on one of the cluster points would bring up the list of the documents contained in the cluster. The drill-down feature is independent from the search and can be activated in any area of the map.

---

[120] Figure produced using http://ai.bpa.arizona.edu/cgi-bin/dmitri/i/refine.cgi?formname=query&fmt=.&q=visualization+search+results+internet [2001-03-02]

[121] Figures produced using http://faculty.cis.drexel.edu/sitemap/sm2/kohonen.html [2001-03-02]

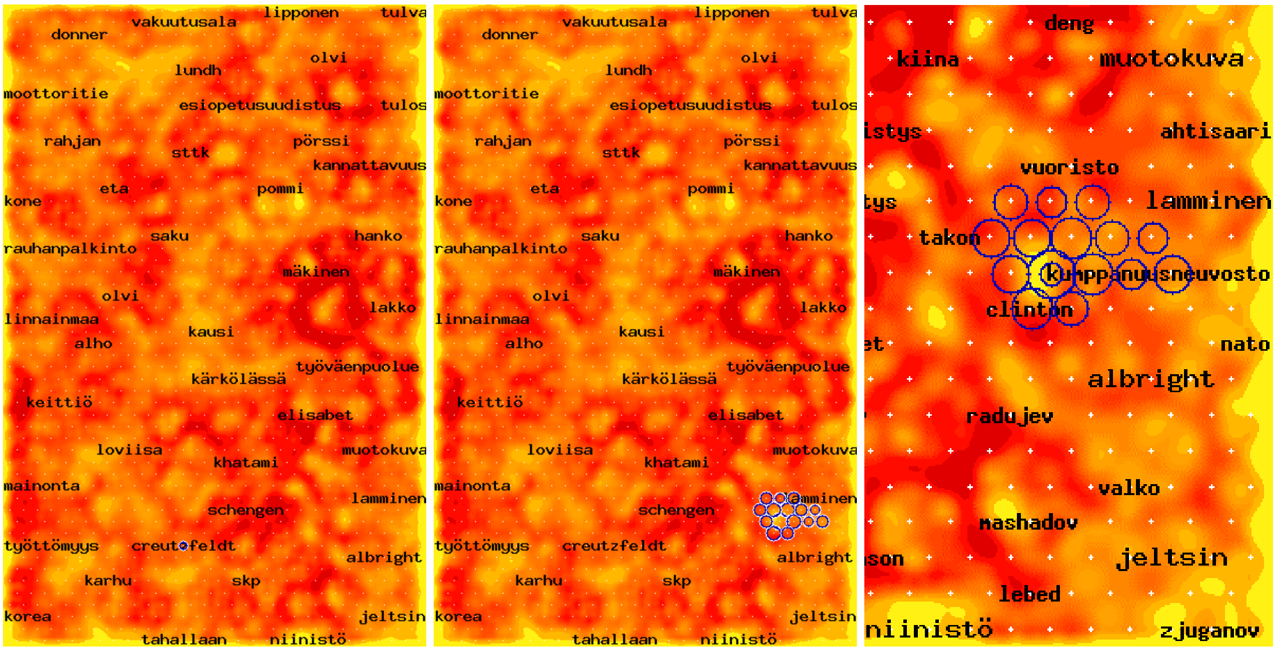


Figure 79: Self Organizing Maps of the WEBSOM system[122] [Kohonen 1998]

Figure 80 shows an example of the libViewer applet from the SOMLib system [Rauber, Bina 1999]. Visualized is a part of a SOM constructed by the SOMLib system from a collection of 420 articles from the TIME Magazine from the 1960's. The SOM uses of 10x15 grid. The left side of Figure 80 shows the 20 lower left cells from the grid. The right side is zoomed in showing the four upper left cells. Zoomed in, documents are shown in more detail using the mapping mechanisms of the libViewer applet described on page 87.

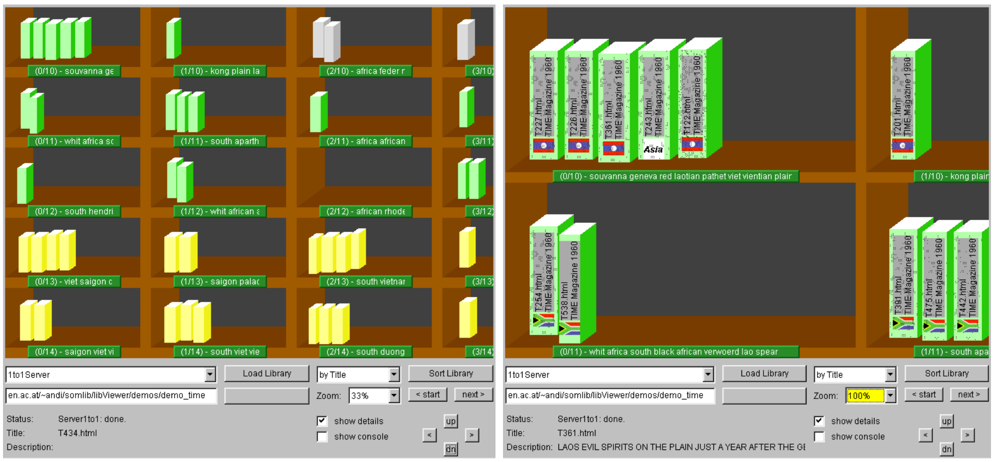


Figure 80: Example of the libViewer from the SOMLib system[123] [Rauber, Bina 1999]

The Sammon view of the Information Navigator [Au, Carey, Sewraz et al. 2000], [Carey 2000], [Carey, Kriwaczek, Rüger 2000] is a further graphical cluster-based representation. It also shows document clusters in a 2D-pane, but uses another mechanism to map points from an n-dimensional space into two dimensions. Instead of the grid layout of the SOM the sammon mapping [Sammon 1969] tries to preserve cluster distances from the n-dimensional space in the two-dimensional space. Figure 81 shows an example using the same document set as used in Figure 61 on page 90 for the Treemap and Figure 72 on page 98 for the Radial visualization of the same system. The left

[122] Figures produced using http://websom.hut.fi/websom/stt/doc/eng/ [2001-03-02]

[123] Figures produced using http://student.ifs.tuwien.ac.at/~andi/libViewer/ [2001-03-02] showing part of the TIME Magazine Article Collection as organized by the SOMLib System.

part of Figure 81 shows the initial view. The cluster labeled “search” in the lower right corner is selected. The term list on the left side of the Sammon view shows the statistically collected terms from the documents in this cluster in descending order of occurrence. The right part of the figure shows a drill-down for this cluster with a sub clustering of its 306 documents. The sub cluster “visualization” is selected. The size and color of the cluster symbols are a nonlinear representation of their cluster size.

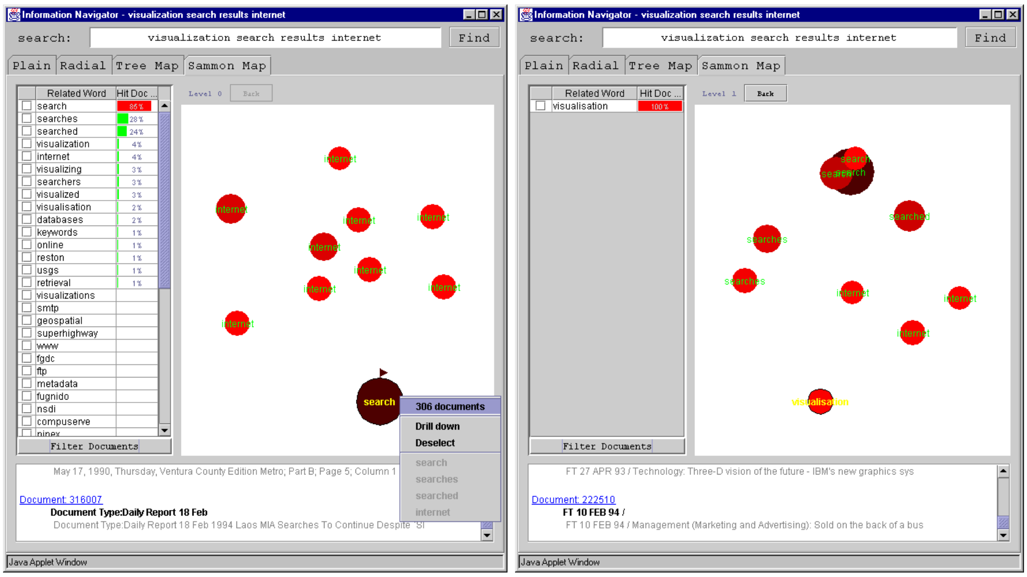


Figure 81: Example of the Sammon View from the Information Navigator [Au, Carey, Sewraz et al. 2000], [Carey, Kriwaczek, Rüger 2000][124]

When talking about clustering the Scatter/Gather paradigm by [Cutting, Karger, Pedersen et al. 1992], [Hearst, Pedersen 1996a] must be mentioned. Scatter/Gather cluster documents into topically coherent groups. Descriptive textual summaries with characterizing topical terms and typical titles are presented to the user. Selecting one ore more clusters invokes a gathering mechanism followed by a new scattering into sub clusters. The work is quite often referenced in the context of information retrieval and sometimes also in the context of information visualization. In a broader sense it may be seen as a possibility for information visualization. [Chen, Yu 2000] excluded it from their meta-analysis of empirical studies of information visualization because Scatter/Gather does not include a “visual-spatial display”.

[Keim, Kriegel 1994] used in the VisDB system pixel-oriented visualization techniques for query specification and data mining in very large databases. The idea is to use every pixel of the display to represent one item of the database. The goal of this technique is to find interesting data spots in scientific, engineering, or environmental databases, but it could probably also be used for the visualization of very large results sets from Web searches. Main principles are the mapping of data items to pixels, sorting according to the relevance for the query, and mapping of the relevance to colors. Figure 82 shows two ideas the authors describe, transferred to Web search results using the 20 document result set of the WebViz-example. Instead of single pixels, squares are used for better visibility. On the left side, every big colored square represents one of the five dimensions: overall relevance and relevance for every one of the four keywords. Inside a big square, the documents are positioned using a spiral mechanism shown in the middle of the figure. The position of a document in the spiral is determined by its overall relevance starting with the most relevant document in the

[124] Figures produced using http://rowan.doc.ic.ac.uk:8000/InfoNavigator/provodnik.html [2001-03-04]

center. A document has always the same position in all dimensions. For a description of general problems with spiral positioning approaches see the end of the explanation of the Document Spiral from Cugini et al. on page 105. On the right side in Figure 82, the grouping arrangement which is another idea by [Keim, Kriegel 1994], is transferred to the visualization of Web search results. Every document is shown with all its dimensions (four keywords plus overall relevance in the example) close together. To position the documents the spiral layout mechanism is used. A comparison of the visualizations of the small result set in Figure 82 with examples taken from [Keim, Kriegel 1994] shown in Figure 83 suggests that these pixel oriented visualization techniques are better suited for trend detections when handling large or very large results sets.

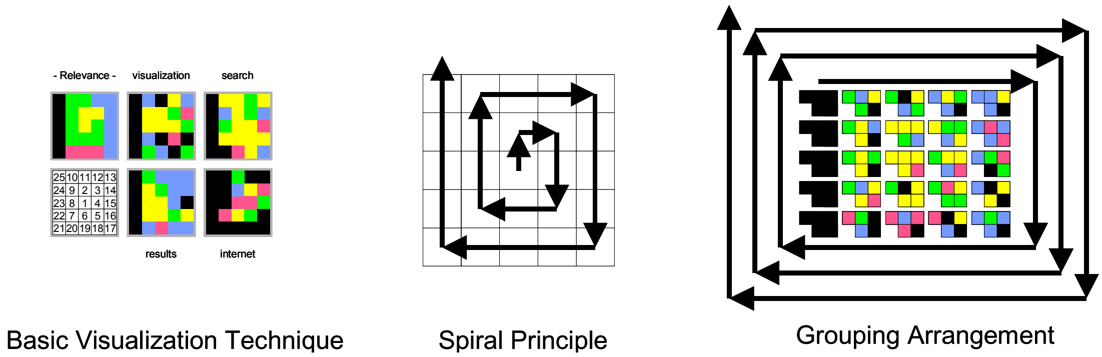


Figure 82: Principles of two of the pixel-oriented visualization techniques used in the VisDB system by [Keim, Kriegel 1994]

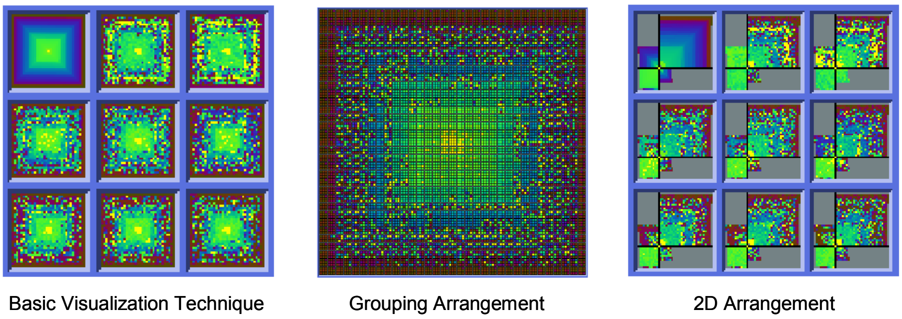


Figure 83: Pixel oriented visualization techniques. Examples taken from [Keim, Kriegel 1994] Figure 6. Eight dimensional data plus overall relevance, 1000 items. Courtesy of Daniel A. Keim

Another pixel-per-value technique for very large data sets is described in [Ankerst, Keim, Kriegel 1996]. In the so-called “Circle Segments”, a circle is divided into a number of segments corresponding to the number of dimensions to be shown. Figure 84 shows the principle by using the four keyword dimensions and the 20 document result set from the WebViz-example. Like the other pixel-per-value techniques described above, Circle Segments seem not to be useful for small result sets.

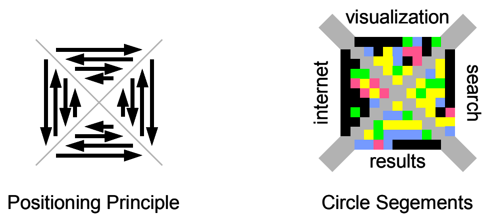


Figure 84: Principle of Circle Segments

[Cugini, Piatko, Laskowski 1997], [Cugini, Laskowski, Piatko 1998], [Cugini, Laskowski, Sebrechts 2000] use in the NIRVE system a number of different spiral, circular, or other designs to visualize interdocument similarities in document spaces. The components are mostly 3D implementations and therefore in principle not suitable for the INSYDER system. Nevertheless, they contain a number of ideas that may work in a 2D implementation. As seen on page 83, the Iconic Representation shows document attributes as part of other visualizations in the NIRVE system. The Three-Keyword Axes Display / 3D-Axes showing interdocument similarities has been mentioned on page 93. Another visualization idea from the authors is the Document Spiral shown in Figure 86. Figure 85 shows the principle in 2D using the 20 document result set of the WebViz-example. The higher the relevance score of a document, the more centrally located is the document on the spiral. The most relevant document is displayed in the center of the spiral. The space between the documents is proportional to their relative document score. Documents with exactly the same score are placed next to each other to avoid overlapping. In the NIRVE system the user can control the density of the overall image. An additional feature allows the elevation of icons above the pane of the spiral by changing the weight of keywords using keyword sliders that can be seen in the upper left corner of Figure 86. The authors state that a major insight provided by the Document Spiral in comparison with a ranked list, is an impression about the distribution of relevance scores in the result set. In [Cugini, Laskowski, Sebrechts 2000] the authors report problems with the usage of a spiral because users tend to assume that spatial closeness of documents in the pane has always a meaning. The truth is that "closeness" is sometimes only caused by the layout algorithm. See for example the documents #8 and #15 "East" of the central document in Figure 85. Document #7 "North" of the central document has a relevance value much closer to #8, but the visualization does suggest lower differences between #8 and #15, than #7 and #8.

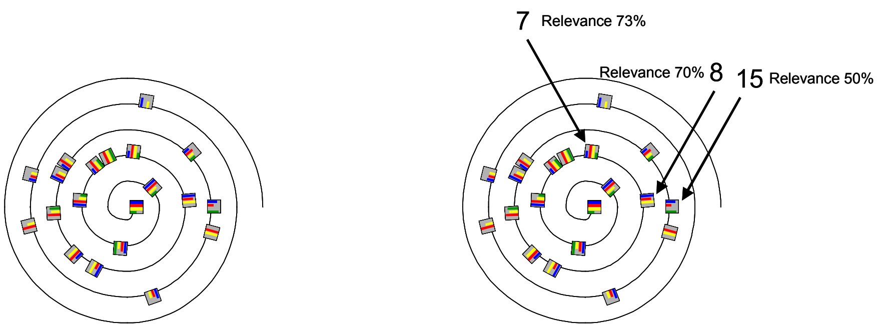


Figure 85: Principle of the Document Spiral in the NIRVE system by [Cugini, Piatko, Laskowski 1997]

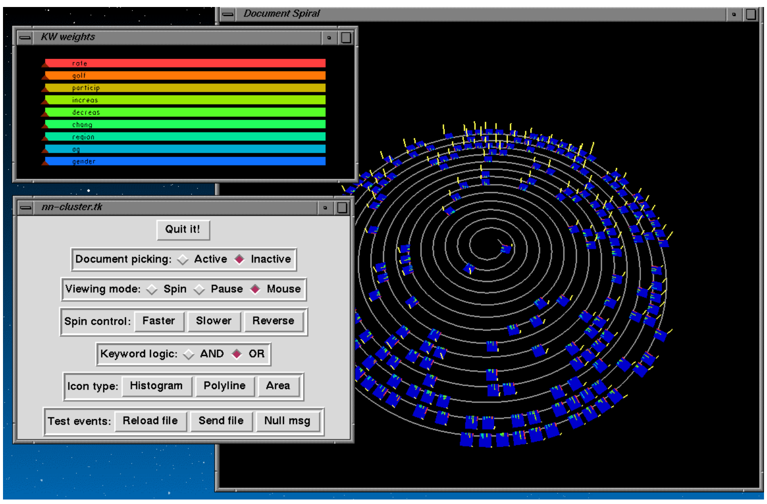


Figure 86: Document Spiral[125] of the NIRVE system. Courtesy of NIST John V. Cugini

[125] Download from http://www.itl.nist.gov/iaui/vvrg/cugini/uicd/gallery/spiral.gif [2001-02-26]

The Nearest Neighbor Circle (NNC) is another component used in the NIRVE system [Cugini, Piatko, Laskowski 1997], [Cugini, Laskowski, Sebrechts 2000]. The idea is to cluster semantically similar documents in the same region. Compared to scatterplots such as used in the SPIRE system, where n-dimensional vectorspaces are compressed to two dimensions, the principle in the case of the NNC used is going one step further by compressing the space to just one dimension. The Iconic Representations of the documents are then positioned upright on a circle like photographic slides in a carousel tray. The space between the documents is proportional to their semantic distance. The left side in Figure 87 shows the principle in 2D using the 20 document result set of the WebViz-example. Clustering in the example is subjective and not computed. In the NIRVE system, as well as for the Document Spiral, the user has the possibility to “elevate” documents by using keyword sliders. The effect can be seen in the example from the NIRVE system in the right part of Figure 87.

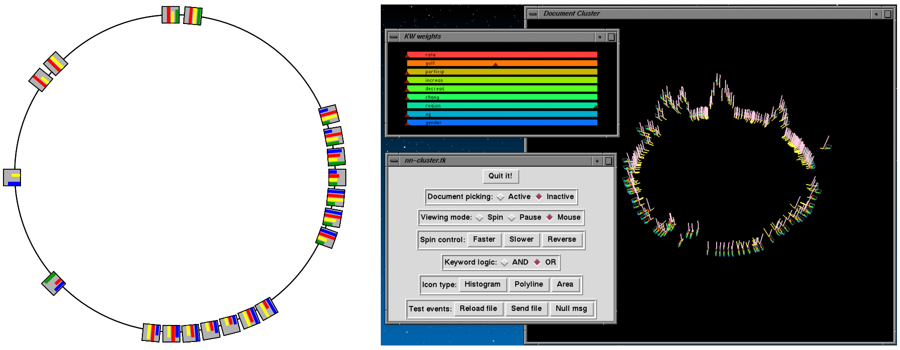


Figure 87: Principle of the Nearest Neighbor Circle (NNC) and example from the NIRVE system[126]. Courtesy of NIST John V. Cugini

The Spoke and Wheel visualization of the NIRVE system [Cugini, Laskowski, Sebrechts 2000], in older publications just named “Document Space” [Cugini, Laskowski, Piatko 1998], is another visualization for document clusters. In contrast to the NNC, the layout is changed, the keyword-concept mapping feature mentioned on page 72 is added and additional representations for the clusters are used. The principle for the cluster representation is the same like for the Iconic Representation of the documents, but with 3D cubes for the concepts showing the average of all documents in the cluster instead of the document specific flat bars. The cluster visualizations are at the outer ends of the spokes [Cugini, Laskowski, Piatko 1998], or at the inner ends [Cugini, Laskowski, Sebrechts 2000]. The left side in Figure 88 shows the principle in 2D using the 20 document result set of the WebViz-example. Clustering in the example is subjective and not computed. Additionally distances between the documents are equal, whereas in the NIRVE system the distance between two documents in a cluster reflects their semantical distance. The right side in Figure 88 shows an example from the NIRVE system.

[126] Download from http://www.itl.nist.gov/iaui/vvrg/cugini/uicd/gallery/nn.gif [2001-02-26]

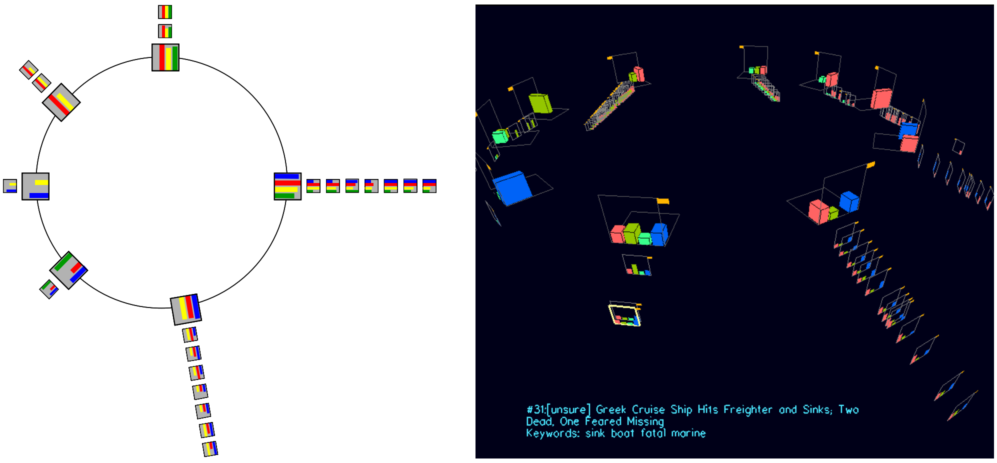


Figure 88: Principle of the Spoke and Wheel and example from the NIRVE system[127]. Courtesy of NIST John V. Cugini

A technique combining a clustering approach and a spiral principle has been used by [Hascoët 1998]. In a first step a set of items is split into clusters. For every cluster a "center item" is identified. The center items are laid out on an ellipse. In a last step the remaining items of each cluster are laid out in a spiral around the cluster center. Figure 89 shows the principle using the 20 document result set of the WebViz-example and the same clusters as used in Figure 87 and Figure 88. Positions on the spiral in the example are random.

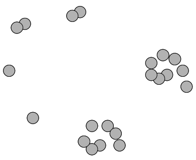


Figure 89: Principle of the spiral display by [Hascoët 1998]

A technique in certain aspects comparable with the bar-graph view of Veerasamy et al. described on page 82 is the Interactive Histograms of the Attribute Explorer [Tweedie, Spence, Williams et al. 1994] and the Influence Explorer [Tweedie, Spence, Dawkes et al. 1996]. Targeted to show relationships within multi-attribute datasets, the component can also be used to visualize inter-document similarities. In theory, the bar-graph of Veerasamy et al. provides similar insights, but whereas in the bar-graph attributes of single documents can be grasped by the user without interaction, in the interactive histograms some interaction is necessary. On the other hand, some trends in the overall result set are easier detectable with the Interactive Histograms. Therefore bar-graphs have been introduced in the last chapter, and Interactive Histograms will be presented here. The idea of Interactive Histograms is to use a group of horizontal and / or vertical one-dimensional histograms to represent a multidimensional dataspace, or a selection of dimensions from a multi-dimensional dataspace. Selections in one or more of the histograms are reflected in the other histograms by a brushing mechanism. Figure 90 and Figure 91 show the principle of Interactive Histograms using the 20 document result set of the WebViz-example. In both examples, 70% to 90% are selected in the "Relevance" histogram. The selection is reflected by black dots / black colored rectangles in the other displayed dimensions: Relevance for the concepts "visualization", "search", "results", and "internet"; Document Size, and Year of publication. [Tweedie, Spence, Williams et al. 1994] describe a number of additional features like the integration of histograms with nominal data, two-color coding when selecting attribute ranges in two histograms, filtering by scale lock-

[127] Download from http://www.itl.nist.gov/iaui/vvrg/cugini/uicd/gallery/docspace-spoke-ship.gif [2001-02-26]

ing, or the examination of individual data items. In Figure 90, the highest ranked document is selected. Color highlighting and a connecting line indicate its corresponding attributes.

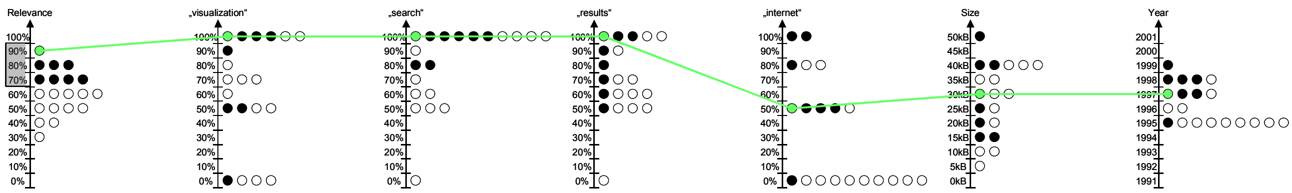


Figure 90: Principle of Interactive Histograms in the Attribute Explorer by [Tweedie, Spence, Williams et al. 1994]

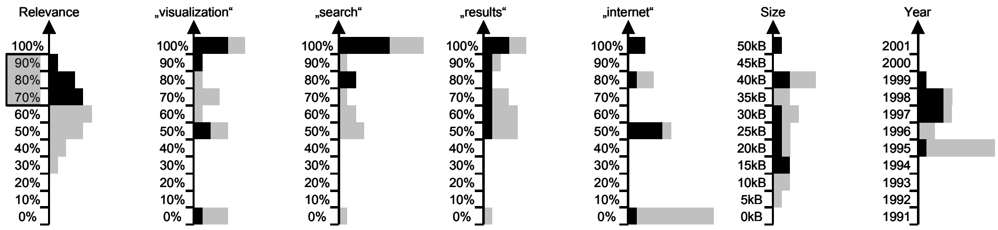


Figure 91: Principle of Interactive Histograms in the Influence Explorer by [Tweedie, Spence, Dawkes et al. 1996]

Showing lines for all documents as it was done for one document in Figure 90 leads to a visualization known as Parallel Coordinates [Inselberg 1985]. Besides Inselberg a lot of authors like [Schmid, Hinterberger 1994] in the VisuLab system, or [Tweedie, Spence, Dawkes et al. 1996] in the Influence Explorer, use Parallel Coordinates for different purposes. For the visualization of document sets Parallel Coordinates have not been used so far. Figure 92 shows the principle using the 20 document result set of the WebViz-example. To a certain degree the small set of 20 documents allows even the direct recognition of document attributes. For larger result set Parallel Coordinates are be focused on the visualization of general trends. An idea of what this will look like provides the last column in Figure 92, which gives the impression that a lot of documents in the result set seem not to deal with “internet”.

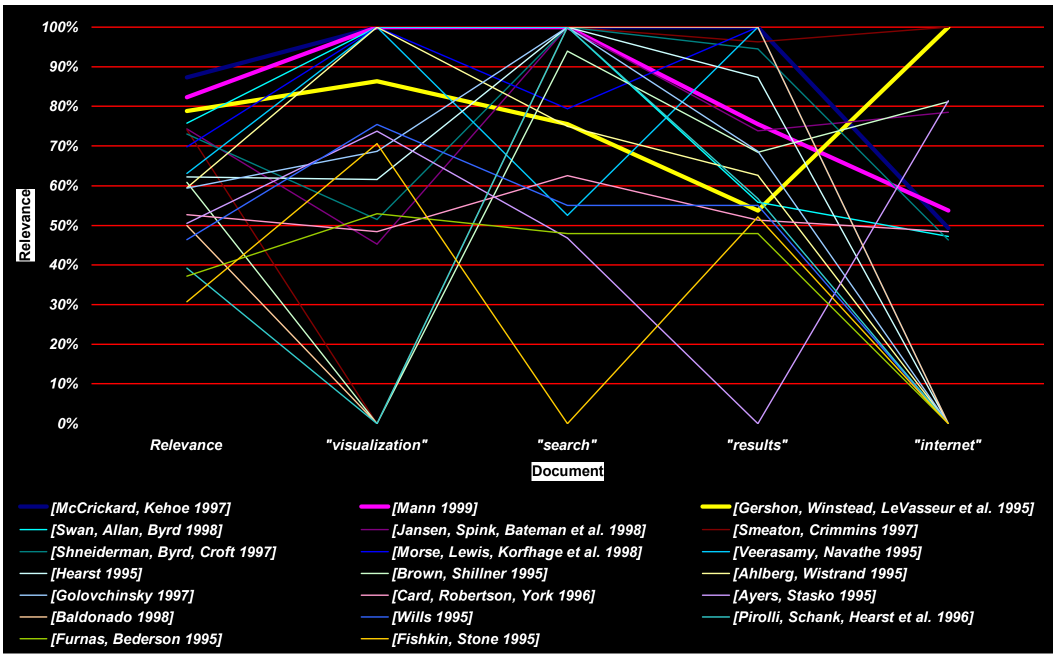


Figure 92: Principle of Parallel Coordinates

The Perspective Wall [Mackinlay, Robertson, Card 1991] of the Information Visualizer system tries to integrate smoothly detailed and contextual views for data tables containing one linear dimension such as date or time. A wide 2D layout of the dataset is folded in such a way that a center panel shows detailed focus data and two perspective panels show the context. The left part of Figure 93 shows the principle using the 20 document result set of the WebViz-example. The year of publication is mapped on the X-axis. The year 1997 is in focus. Relevance grouped in ranges is mapped on the Y-axis with the most relevant documents in the top row. The right part of Figure 93 shows the complete undistorted view of the dataset, which is quite small with 20 documents, and may therefore not need distortion. [Mukherjea, Foley, Hudson 1995] had the idea to integrate a Perspective Wall in the Navigational View Builder. [Mukherjea, Hirata, Hara 1997] describe the integration of a Perspective Wall in the AMORE (Advanced Multimedia Oriented Retrieval Engine) system.

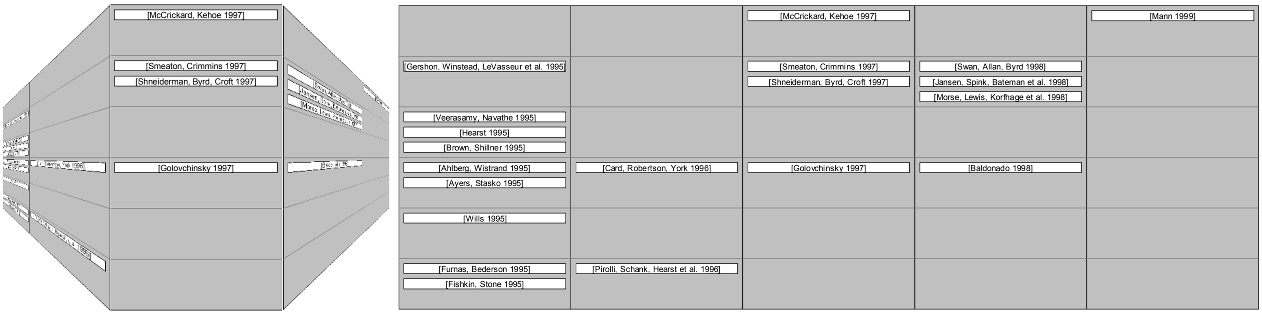


Figure 93: Principle of the Perspective Wall by [Mackinlay, Robertson, Card 1991]

As mentioned on page 57 when introducing the wall metaphor, the Perspective Wall is a successor of the Bifocal Display by [Spence, Apperley 1982]. Figure 94 shows the same example as in Figure 93 using the principle of the Bifocal Display.

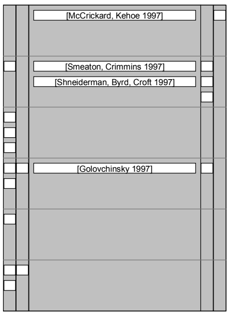


Figure 94: Principle of the Bifocal Display by [Spence, Apperley 1982]

Other visualization ideas specialized in mapping data tables to visual structures, when time is the dominant feature, are LifeLines [Plaisant, Milash, Rose et al. 1996] for biographical data, or ThemeRiver [Havre, Hetzler, Nowell 2000] for changes in thematic variations over time in a document collection.

A multifocal approach using focus-plus-context techniques to map data tables on visual structures can be found in a number of tabular data representations. Scatter/Gather is such an example. Others are interactive tables like the Table Lens [Rao, Card 1994] or FOCUS [Spenke, Beilken, Berlage 1996]. Textual and graphical representations of the data tables are used in both components. Focus-plus-context allows showing more cells of the data table on the screen then without this technique. In both components, the coherence of rows and columns and their labels is preserved when distorting parts of the view. The graphical elements are used for pattern recognition when working with quantitative variables. Whereas in the Table Lens the cases are displayed in rows, in FOCUS they are displayed as columns.

The idea of the Table Lens has been commercialized by Inxight Software, Inc., and is now called Eureka. Figure 95 shows an example using the system and the 20 document result set of the WebViz-example. The columns “Size” and “Year” have been categorized. A row focus is put on the same documents from 1997 focused in Figure 93 and Figure 94 showing the principles of the Perspective Wall and the Bifocal Display. A column focus is put on “Name”, Relevance”, and “Year”. The dataset is ordered by Year. The right part of the figure shows the same dataset ordered by Relevance instead of Year using the same multiple focus points.

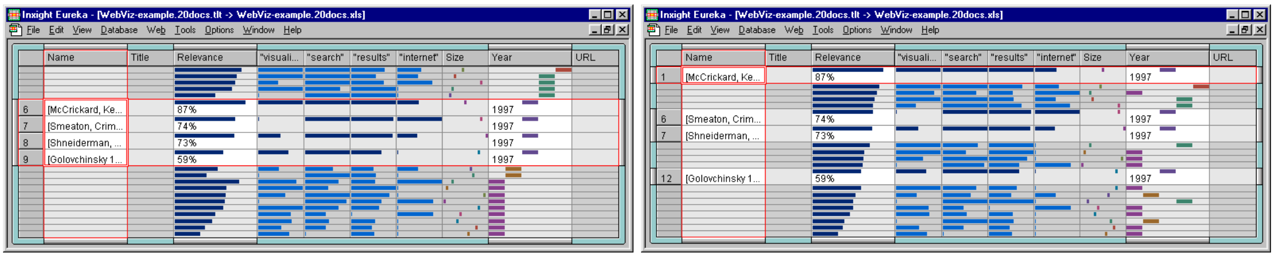


Figure 95: Example of Table Lens / Eureka, ordered by Year (left) and Relevance (right)[128]

The Table Lens and Eureka support a number of interaction possibilities that will not be discussed further. Before turning to the next visualization idea one interesting possibility should be mentioned. The table presentation can easily be reduced to a simple bar-graph view of the dataset. In Figure 96 this step was completed. The view in the left part of the figure is comparable to a skewed bar-graph view from Veerasamy et al. like shown in Figure 48 on page 82. The same is true for the view on the right side of the figure and the Bargraph from the XINQUERY system shown in Figure 59 on page 89. In contrast to the components mentioned earlier, the Table Lens or Eureka support many more interaction possibilities.

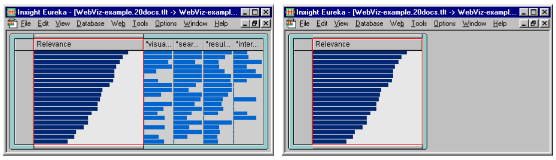


Figure 96: “Bar-graph view” of Table Lens / Eureka

[Spenke, Beilken, Berlage 1996] also describe a system using an interactive table they call FOCUS (Feature-Oriented Catalog USer interface). Some similarities and differences of the Table Lens had already been described above. Figure 97 shows the same examples as for Eureka, by using InfoZoom, which is a version of FOCUS, commercialized by humanIT GmbH. The left figure is sorted by Year, the right part by Relevance. A special feature of the system is the automatic recognition of identical values in neighboring cells. In cases like this the attribute values are only labeled once. The feature does not work for cases in focus.

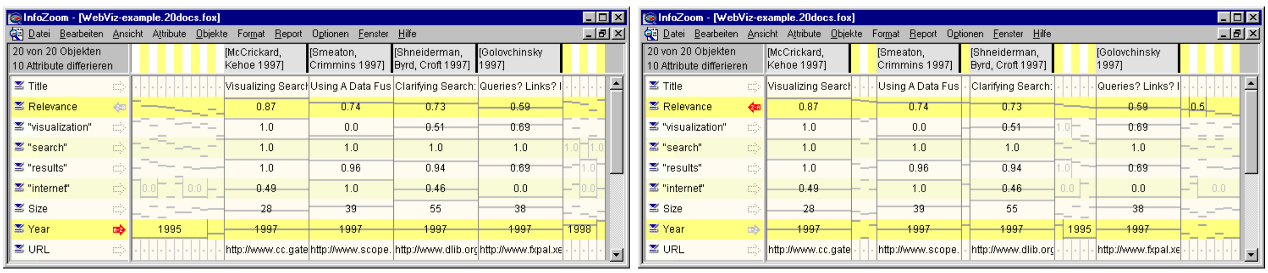


Figure 97: Example of FOCUS / InfoZoom, ordered by Year (left) and Relevance (right)[129]

Another feature InfoZoom provides, is an overview of the distribution of the values. Instead of ordering the values by cases, each dimension is ordered by itself. Figure 98 shows the feature using the 20 document result set of the WebViz-example.

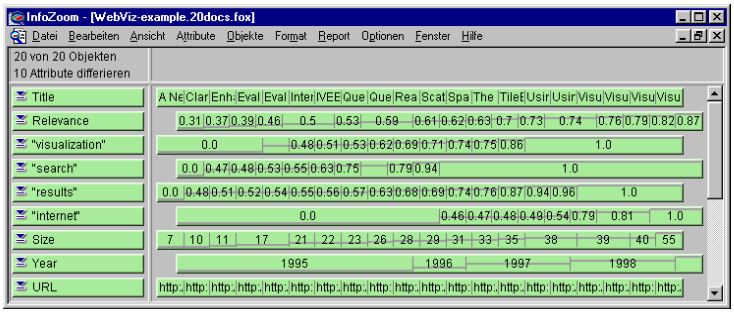


Figure 98: Example of the FOCUS / InfoZoom overview function

This chapter introduced a number of ideas for the visualization of document attributes. Table 19 provides an overview of the components discussed. Besides different forms of using bars or rectangles, we have seen a broad variety of scatterplots or starfields. One of the key issues in mapping raw data over data tables to visual structures on 2D panes is dimensionality reduction. The methods used range from the simple selection of dimensions, to compressions of n-dimensional spaces to low dimensional spaces. A special mechanism to calculate positions was the usage of reference points or Points of Interest (POIs). Landscapes shared a number of principles with starfields. Self-Organizing Maps show how different the mapping from a certain form of data table to visual structures can be. With or after pixel-based techniques, a number of spiral- or circle-based visualizations were introduced. After a number of classical visualizations such as histograms or Parallel Coordinates, the chapter ended with different bifocal or multifocal timeline- or table-based approaches.

Component	Literature	Used in System
Bargraph	[Shneiderman, Byrd, Croft 1997]	XINQUERY
FISH / Treemaps	[Mitchell, Day, Hirschman 1995]	Starfish
Value Bars	[Chimera 1992]	
Scatterplot (geographical)	[Williamson, Shneiderman 1992]	Dynamic Homefinder
	[Fishkin, Stone 1995]	Magic Lens Filters
	[Higgins, Lucas, Senn 1999]	VisageWeb
Starfield / Scatterplot (dimension selection)	[Ahlberg, Shneiderman 1994], [Jog, Shneiderman 1995]	FilmFinder
	[Ahlberg, Wistrand 1995], [Wistrand 1995]	IVEE
	[Spotfire 2001]	Spotfire Pro
	[Golovchinsky 1997], [Golovchinsky 1997a], [Golovchinsky 1997b], [Golovchinsky, Chignell 1997]	VOIR
Three-Keyword Axes Display / 3D-Axes / 3D-Scatterplot	[Cugini, Piatko, Laskowski 1997]	NIRVE

[128] Figures produced by using a trial version of Inxight Eureka V 1.2.0.2. Download from http://www.inxight.com/products_eu/eureka/eureka_download.html [2001-03-11]

[129] Figures produced by using a trial version of humanIT InfoZoom V 3.40. Download from http://www.humanit.de/de/products/infozoom/download.html [2001-03-10]

Component	Literature	Used in System
Starfield / Scatterplot (dimension compression)	[Wise, Thomas, Pennock et al. 1995], [Wise 1999]	MVAB / SPIRE
2D Document Space with Reference Points / Points of Interest (POIs)	[Korfhage 1991]	VIBE
	[Morse, Lewis 1997]	WebVIBE
	[McCrickard, Kehoe 1997]	SQWID
	[Au, Carey, Sewraz et al. 2000], [Carey 2000], [Carey, Kriwaczek, Rüger 2000]	Information Navigator
3D Document Space / Relevance Sphere with Reference Points / Points of Interest (POIs)	[Benford, Snowdon, Greenhalgh et al. 1995]	VR-VIBE
	[Hemmje 1993a], [Hemmje, Kunkel, Willett 1994]	LyberWorld
Landscapes	[Chalmers, Chitson 1992], [Chalmers 1993], [Chalmers 1995], [Chalmers 1996]	Bead
	[Andrews 1995]	Harmony
	[Krohn 1995]	Vineta
	[Wise, Thomas, Pennock et al. 1995]	SPIRE
Self-Organizing Map (SOM)	[Lin, Soergel, Marchionini 1991]	Self-Organizing Semantic Map
	[Chen, Schuffels, Orwig 1996]	ET-map
	http://ai.bpa.arizona.edu [2001-03-03]	Adaptive SOM
	http://ai.bpa.arizona.edu [2001-03-03]	CI Spider
	http://faculty.cis.drexel.edu [2001-03-03]	Visual SiteMap
	[Kohonen 1998]	WEBSOM
	[Rauber, Bina 1999], [Rauber, Bina 2000]	SOMLib
Sammon view	[Au, Carey, Sewraz et al. 2000], [Carey 2000], [Carey, Kriwaczek, Rüger 2000]	Information Navigator
Pixel oriented visualization techniques	[Keim, Kriegel 1994]	VisDB
Circle Segments	[Ankerst, Keim, Kriegel 1996]	
Document Spiral	[Cugini, Piatko, Laskowski 1997], [Cugini, Laskowski, Sebrechts 2000]	NIRVE
Nearest Neighbor Circle (NNC)	[Cugini, Piatko, Laskowski 1997], [Cugini, Laskowski, Sebrechts 2000]	NIRVE
Spoke and Wheel / Document Space	[Cugini, Laskowski, Piatko 1998], [Cugini, Laskowski, Sebrechts 2000]	NIRVE
Spring display	[Hascoët 1998]	
Interactive Histograms	[Tweedie, Spence, Williams et al. 1994].	Attribute Explorer
	[Tweedie, Spence, Dawkes et al. 1996]	Influence Explorer
Parallel Coordinates	[Schmid, Hinterberger 1994]	VisuLab
	[Tweedie, Spence, Dawkes et al. 1996]	Influence Explorer
Perspective Wall	[Mackinlay, Robertson, Card 1991]	Information Visualizer
	[Mukherjea, Foley, Hudson 1995]	Navigational View Builder
	[Mukherjea, Hirata, Hara 1997]	AMORE
Interactive Table	[Rao, Card 1994]	Table Lens / Eureka
	[Spenke, Beilken, Berlage 1996]	FOCUS / InfoZoom

Table 19: Components for the visualization of interdocument similarities

3.3.3.4. Visualization of interdocument connections

This thesis does not cover the visualization of hyperlinks structures or interdocument connections, although there appears to be a obvious connection. The INSYDER team formed an opinion during the early phase of the project. It might be interesting to visualize hyperlinks between the nodes of a coherent hypertext or the pages of a single Web site, or a very few numbers of Web sites. The number of hyperlinks between the documents in a result set of a random search in the Web may be to low to be worth the effort to maintain extra data structures and special visualizations. Discussions with potential users of the system also revealed low interest in visualization of the interdocument connections of a result set. For an overview covering visualizations of interdocument connections or of (document) hierarchies and trees see [Wolte 1998] or [Bekavac 1999]. Examples for components from this area are 3D-worlds or networks used in a number of systems [Fairchild, Poltrock, Furnas 1988], [Hendley, Drew, Wood et al. 1995], [Wood, Drew, Beale et al. 1995], the Bullseye view of the WebQuery / VANISH system [Carrière, Kazman 1997], Cone Trees [Robertson, Mackinlay, Card 1991], [Hemmje, Kunkel, Willett 1994], [Munzner, Burchard 1995], [Carrière, Kazman 1995], [Mukherjea, Foley 1995], [Carrière, Kazman 1997], [Hearst, Karadi 1997], the Cheops approach [Vroomen 1998], Docuverse in the CASCADE system [Spring, Morse, Heo 1996], [Heo, Morse, Willms et al. 1996], the Hyperspace View as an enhancement to the NCSA Mosaic Browser [Gershon, Winstead, LeVasseur et al. 1995], [Gershon, LeVasseur, Winstead et al. 1995], Hyperbolic Trees [Lamping, Rao, Pirolli 1995], [Lamping, Rao 1996], [Munzner, Burchard 1995], [Munzner 1997], Webview[130] in the CASCADE system [Spring, Morse, Heo 1996], [Heo, Morse, Willms et al. 1996], the already in certain aspects on page 90 discussed Treemaps [Pearson, Steinmetz 1993], [Asahi, Turo, Shneiderman 1995], [Mukherjea, Foley 1995], and a lot of others.

3.3.4. Systems

The last item introduced in Figure 15 on page 50 as a possible dimensions for classification of visualizations are “systems”. Many systems using metaphors, visualization techniques, and visualization components were mentioned in the last chapter. The purpose of this chapter is to complete the overview of the state-of-the-art of information visualization for web search results. This overview covers the systems discussed in this thesis. Systems mentioned in the following chapter about multiple coordinated views are included. The overview is structured by using a categorization into four main groups: systems fully or partially used for the visualization of Web search results, Hypertext browsers, classical IR-systems, and others. Literature, components, and metaphors listed are restricted to the ones mentioned in this thesis.

System	Literature	Components	Metaphors
Adaptive SOM	http://ai.bpa.arizona.edu [2001-03-03]	Self-Organizing Map	
AltaVista	[Bourdoncle 1997], [Bourdoncle 1999], [Bourdoncle, Bertin 2000]	Cow9, LiveTopics	
AMORE	[Mukherjea, Hirata, Hara 1997]	Scatterplot, Perspective Wall	
CI Spider	http://ai.bpa.arizona.edu [2001-03-03]	Self-Organizing Map	

[130] The Webview component in the CASCADE system from [Spring, Morse, Heo 1996] should not to be mixed up with the webView system from [Cockburn, Greenburg, McKenzie et al. 1999], [Cockburn, Greenberg 1999], [Cockburn, Greenberg 1999a]

System	Literature	Components	Metaphors
DigOut4U	http://www.arisem.com [2001-02-11]	Relevant Extracts plus Curve of Relevance	
ET-map	[Chen, Schuffels, Orwig 1996]	Self-Organizing Map	
Information Visualizer (WebBook / Web Forager)	[Robertson, Card, Mackinlay 1993], [Card, Robertson, York 1996]	WebBook, Web Forager, Document Lens	Book, Bookshelf, Lens
NetMap	[Chase, D'Amore, Gershon et al. 1998]	Entity Relation Visualization	
NIRVE	[Cugini, Piatko, Laskowski 1997], [Cugini, Laskowski, Piatko 1998], [Cugini, Laskowski, Sebrechts 2000]	Keyword-Concept Matrix / Concept Control, Concept Globe, Iconic Representation, Three-Keyword Axes Display / 3D-Scatterplot, Document Spiral, Nearest Neighbor Circle (NNC) Spoke and Wheel / Document Space	
Self-Organizing Semantic Map	[Lin, Soergel, Marchionini 1991]	Self-Organizing Map	
SOMLib	[Rauber, Bina 1999], [Rauber, Bina 2000]	libViewer applet	Book, Bookshelf
SQWID	[McCrickard, Kehoe 1997]	2D Document Space with Reference Points / Points of Interest (POIs)	
Starfish	[Mitchell, Day, Hirschman 1995]	FISH / Treemap	
TopicShop	[Amento, Hill, Terveen et al. 1999]	Thumbnail views	
VIEWER	[Berenci, Carpineto, Giannini 1998]	Bargraph	
VIR pre-prototype	[Bekavac 1999]	"MapWindow", "tree-view"	
VisageWeb	[Higgins, Lucas, Senn 1999]	Starfield / Scatterplot, Scatterplot (geographical), Thumbnail views	
Visual SiteMap	http://faculty.cis.drexel.edu [2001-03-03]	Self-Organizing Map	
VOIR	[Golovchinsky 1997], [Golovchinsky 1997a], [Golovchinsky 1997b], [Golovchinsky, Chignell 1997]	Retrieval History Histogram, Scatterplot (global overview)	Newspaper
VR-emb	[Bekavac 1999]		Landscape, Tower
WEBSOM	[Kohonen 1998]	Self-Organizing Map	
WebVIBE	[Morse, Lewis 1997]	2D Document Space with Reference Points / Points of Interest (POIs)	Magnet

Table 20: Systems fully or partially used for the visualization of Web search results or hypertext

System	Literature	Components	Metaphors
3D-visualization INQUERY-based	[Swan, Allan 1996] / [Allan, Leouski, Swan 1997]	"ranked list", "text viewer", Document Map, Concept List, Concept Map	
AI-STARS	[Anick, Brennan, Flynn et al. 1990]	Query Reformulation Workspace	
BEAD	[Chalmers 1993], [Chalmers 1995], [Chalmers 1996]		Landscape

System	Literature	Components	Metaphors
BOOK HOUSE	[Pejtersen 1989]		Book, Rooms, Building
Envision	[Nowell, France, Hix et al. 1996]	Matrix of Icons	
FancyV	[Byrd 1999]	VQRa	
GESINE	[Bürdek, Eibl, Krause 1999]	"Bracket"	
InfoCrystal	[Spoerri 1993], [Spoerri 1993a]		
Information Naviga-tor	[Fowler, Fowler, Wilson 1991], [Fowler, Wilson, Fowler 1992]	Request Map	
Information Naviga-tor	[Au, Carey, Sewraz et al. 2000], [Carey 2000], [Carey, Kriwaczek, Rüger 2000]	Treemap View, Radial visualiza-tion, Sammon view	
Information Visual-izer (Butterfly)	[Mackinlay, Rao, Card 1995]	Butterfly, Pile	Butterfly, Pile
J24	[Ogden, Davis, Rice 1998]	Thumbnail view	
LyberWorld	[Hemmje 1993a], [Hemmje, Kunkel, Willett 1994]	3D Document Space / Relevance Sphere with Reference Points / Points of Interest (POIs)	
MosaicG	[Ayers, Stasko 1995]	Thumbnail views	
MVAB / SPIRE	[Wise, Thomas, Pennock et al. 1995]	ThemeScapes, Galaxies / Starfield / Scatterplot (dimension compres-sion)	Landscape, Galaxy
TeSS	[Hertzum, Frøkjær 1996]	Venn diagrams	
TileBars	[Hearst 1995].	TileBars	
Tkinq	[Veerasamy, Hudson, Navathe 1995], [Veerasamy 1996], [Veerasamy, Belkin 1996].	Positive / Negative Feedback, bar-graph	Tkinq
VIBE	[Korfhage 1991]	2D Document Space with Refer-ence Points / Points of Interest (POIs)	
Vineta	[Krohn 1995], [Elzer, Krohn 1997]		Landscape, Galaxy
VR-VIBE	[Benford, Snowdon, Greenhalgh et al. 1995]	3D Document Space / Relevance Sphere with Reference Points / Points of Interest (POIs)	
VQuery	[Jones 1998]	Venn diagrams workspace	
WInquery	[Shneiderman, Byrd, Croft 1997], [Byrd 1999]	Stacked Histograms / VQRa	
XINQUERY	[Shneiderman, Byrd, Croft 1997]	Bargraph	

Table 21: Classical IR and library systems, including those with Web-based interface.

System	Literature	Components	Metaphors
Data Mountain	[Robertson, Czerwinski, Larson et al. 1998], [Czerwinski, Dumais, Robertson et al. 1999]	Thumbnail view	
DeckScape	[Brown, Shillner 1995]	"deck"	Pile
Guided Tour proto-type	[Guinan, Smeaton 1992]		Guided Tour
Harmony Hyper-G / Hyper View	[Andrews 1995]	Harmony VRweb 3D scene viewer	Landscape
Navigational View Builder	[Mukherjea, Foley, Hudson 1995]	Perspective Wall	
Pad++ (PadPrints)	[Hightower, Ring, Helfman et al. 1998]	Thumbnail views	

System	Literature	Components	Metaphors
Pad++ (Web browser)	[Bederson, Hollan, Perlin et al. 1996], [Bederson, Hollan, Stewart et al. 1998]	Portals, Thumbnail views	
SuperBook / Mitey-Book	[Egan, Remde, Gomez et al. 1989]		Book
WebStage	[Yamaguchi, Hosomi, Miyashita 1997]		Television
webView and other unnamed systems	[Cockburn, Greenburg, McKenzie et al. 1999], [Cockburn, Greenberg 1999], [Cockburn, Greenberg 1999a], [Kaasten, Greenberg 2000], [Kaasten, Greenberg 2001]	Thumbnail views	

Table 22: Hypertext browsing systems, including Web browsers and their add-ons

System	Literature	Components	Metaphors
Attribute Explorer	[Tweedie, Spence, Williams et al. 1994]	Interactive Histograms	
CASCADE	[Spring, Morse, Heo 1996], [Heo, Morse, Willms et al. 1996]	Mural, TileBars	
Context Lenses	[Dieberger, Russell 2001]	TileBars, Context Lenses	
DeckView	[Ginsburg, Marks, Shieber 1996]	Thumbnail views	
Dotplots	[Church, Helfman 1993]	Dotplots	
FilmFinder	[Ahlberg, Shneiderman 1994], [Jog, Shneiderman 1995]	Starfield / Scatterplot	
Filter/Flow	[Shneiderman 1991], [Young, Shneiderman 1993]		Flowing Water
FRIEND21 project	[Nonogaki, Ueda 1991]		Television
GroupLens	[Resnick, Iacovou, Sucak et al. 1994]		Lens
Influence Explorer	[Tweedie, Spence, Dawkes et al. 1996]	Interactive Histograms, Parallel Coordinates	
Information City ontology	[Dieberger 1994], [Dieberger, Frank 1998]		City, Building, Rooms
Information Mural	[Jerding, Stasko 1995], [Jerding, Stasko 1997]	Information Mural	
Information Visualizer	[Robertson, Mackinlay, Card 1991], [Robertson, Card, Mackinlay 1993], [Mackinlay, Robertson, Card 1991]	3D/Rooms, Data Sculpture, Perspective Wall, Cone Trees	Rooms, Building, Sculpture, Wall
IVEE	[Ahlberg, Wistrand 1995], [Wistrand 1995]	Starfield / Scatterplot	
Macintosh	[Mander, Salomon, Wong 1992], [Rose, Mander, Oren et al. 1993]		Pile
"Online Store"	[Bryan, Gershman 2000]		Aquarium
Rooms	[Henderson, Card 1986]		Rooms
SeeSoft	[Eick, Steffen, Sumner 1992], [Eick 1994], [Wills 1995]	SeeSoft bar view	
Spotfire Pro	[Spotfire 2001]	Starfield / Scatterplot	
Table Lens	[Rao, Card 1994]		Lens
Toolglass, MagicLens	[Bier, Stone, Fishkin et al. 1994], [Stone, Fishkin, Bier 1994], [Fishkin, Stone 1995]	See-through tools, Movable Filters, Magic Lenses	Lens
VisDB	[Keim, Kriegel 1994]	Pixel oriented visualization techniques	
VisuLab	[Schmid, Hinterberger 1994]	Parallel Coordinates	

Table 23: Other systems

3.4. State of the Art: Multiple Coordinated Views

So far, visualization ideas in this thesis have mainly been discussed as isolated items. Ahlberg and Wistrand stated in 1995: "*Ours and many others' work on interactive visualization seem to point to that successful visualization environments does not depend on one single powerful visualization, quite contrary a whole smörgåsbord of visualizations appropriate for various tasks and datatypes is closer to a successful solution*." [Ahlberg, Wistrand 1995]. Their IVEE system, later commercialized under the name Spotfire Pro, is an excellent example of following this approach. The same idea is behind other systems such as the Snap-Together Visualization by [North 2000], or the ADVIZOR toolkit described in [Eick 2000]. The "Smörgåsbord" itself is important. In addition, coordination between different views plays an important role [North, Shneiderman 1997], [Baldonado, Woodruff, Kuchinsky 2000], [Eick 2000]. The idea of coordinating more than one view has already been mentioned for techniques such as brushing and linking or overview plus detail. For the latter, the difference between time multiplexing and space multiplexing has been discussed. When introducing concrete visualization components a number of them have been examples using two or more coordinated views. Just to name some: Relevance Curve / Relevance Extracts, bar-graphs / list, VQRb / text, Value Bars, ThemeView, the pixel-oriented components in the VisDB system, or Interactive Histograms. What has not been discussed so far, are systems that implement multiple coordinated views by using different components. IVEE / Spotfire, Snap-Together, and ADVIZOR are general-purpose tools of this kind for the visualization of data from spreadsheets, database systems, or OLAP Cubes. [Baldonado, Woodruff, Kuchinsky 2000] list a number of multiple view systems. Good reviews of Coordinated-Visualization systems can also be found in [North, Shneiderman 1997], [North, Shneiderman 2000], and [North 2000].

Focusing on the visualization of search results, some of the components using more than one view, and listed in the last chapters are part of systems from traditional IR or search tools for the Web. In this area exist several systems that use not only one component with different views, but also different components for the visualization of search results. Examples are Envision, the Information Vizualizer, NIRVE, or the Information Navigator. Interesting in this context are the taxonomy by [North, Shneiderman 1997] for multiple window coordination, and the guidelines for using multiple views in Information Visualization by [Baldonado, Woodruff, Kuchinsky 2000].

[North, Shneiderman 1997] constructed a 2x3 taxonomy using the two dimensions "coordination" and "collection of information items" to describe the different possibilities to coordinate multiple windows. Their three coordination possibilities are:

- Selecting items ⇔ selecting items
- Navigating views ⇔ navigating views
- Selecting items ⇔ navigating views

The second dimension defines if the information items that are contained in the coordinated view are the same as in the manipulated view or different ones. The authors explain all combinations in detail and provide several examples.

[Baldonado, Woodruff, Kuchinsky 2000] use three dimensions to characterize multiple view systems and a number of additional explanations to describe differences. The three dimensions are Selection, Presentation, and Interaction. Additional explanations extend for example aspects described by [North, Shneiderman 1997] in their second dimension "information items" (same or different). [Baldonado, Woodruff, Kuchinsky 2000] discuss three examples in which data sets can differ: a data set can be a subset of another, it can contain aggregates of individual values of a second set, or it can contain entirely different information. The "Selection" dimension describes the selected views or visualizations. The "Presentation" dimension characterizes the type of presentation as "sequentially" or "simultaneously". In the "Interaction" dimension "navigational slaving" and "linking" with it's special sub form "brushing" are listed as common techniques. The problem of the combination of these three dimensions is that, the first two are in general uniform for a whole system, but the third one can differ for different combinations of views or visualizations. A system has a selection of views or visualizations, and the type of usage may be sequentially or simultaneously. Special cases are possible, such as a system where the presentation mode can be changed, or a system presenting the visualizations sequentially but using an additional overview plus detail technique. In contrast to these special cases the interaction possibilities and the type of information items displayed, may be different for each combination of views or visualizations. A possible description scheme for multiple view systems could be the following:

- Selection: Visualization A, Visualization B, Visualization C, Visualization D, …
- Presentation: Sequentially / Simultaneously
- Coordination Visualization A - Visualization B
    - Information Items: Same / Subset / Aggregates / Different / …
    - Interaction: Selection slaving / navigation slaving / selection-to-navigation slaving / …
- Coordination Visualization A - Visualization C
    - Information Items: Same / Subset / Aggregates / Different / …
    - Interaction: Selection slaving / navigation slaving / selection-to-navigation slaving / …
- …

When trying to describe multiple view systems precisely using this scheme, a number of additional differentiations may be necessary. [Eick 2000] classifies for example for the ADVIZOR environment the following interface actions as component specific or linked between components. Linked between components in ADVIZOR are color, focus, selection, and exclusion. View-specific are panning and zooming, scrolling, scaling, sorting, label options, and layout. There is no navigational slaving, a number of forms of selection slaving, and some hints that there may be additional coordinations possible such as sorting or label options. Besides these sub forms of selection and navigational slaving, and the detection of possible additional dimensions, the information item dimension will also be a candidate for flexibilization. With uncoordinated or only partially coordinated selection, the type of information items may change during interaction. Even more complicated becomes the attempt to classify multiple view systems exactly when for example ideas are taken into account such as the advanced coordinations described by [North, Shneiderman 1997]. Advanced coordinations give the user the possibility to choose or change coordination between components. Despite the fact that detailed comparisons are not easy, the overviews by [North, Shneiderman 1997], [North, Shneiderman 2000], [North 2000], and [Baldonado, Woodruff, Kuchinsky 2000] deliver important classification ideas, and a good state-of-the-art of multiple view systems and the coordination of different views.

Other vital contributions by [Baldonado, Woodruff, Kuchinsky 2000] are their guidelines for using multiple views in Information Visualization. The eight guidelines are grouped into two parts:

- To support the decision whether or not multiple view systems are appropriate to use in a certain situation, and
- when a multiple view system is created, to support decisions for choices of presentations and interactions.

In other words, the rules are grouped into the two parts when and how to use multiple views. The guidelines also include discussions of possible positive or negative impacts from decisions on the utility of the system and its components. In earlier papers [Mann 1999], [Mann, Reiterer 1999] a number of drawbacks have been mentioned when using multiple views. The user interface of the system becomes more complex, and therefore will be harder to use. It will be more difficult to develop. The user can choose an inappropriate visualization for a specific situation. [Baldonado, Woodruff, Kuchinsky 2000] provide in the discussion of their guidelines a detailed description of a number of additional drawbacks and possibilities on how to avoid problems. Table 24 shows the guidelines when to use multiple views together with possible positive and negative impacts on utility. Table 25 shows the guidelines how to use multiple views and the impacts.

Rule	Summary	Major Positive Impacts on Utility	Major Negative Impacts on Utility
Diversity	Use multiple views when there is a diversity of attributes, models, user profiles, levels of abstraction, or genres	memory	learning computational overhead display space overhead
Complementarity	Use multiple views when different views bring out correlations and / or disparities	memory comparison context switching	learning computational overhead display space overhead
Decomposition	Partition complex data into multiple views to create manageable chunks and to provide insight into the interaction among different dimensions.	memory comparison	learning computational overhead display space overhead
Parsimony	Use multiple views minimally	learning computational overhead display space overhead	memory comparison context switching

Table 24: Rules when to use multiple views and areas of major impact on utility according to [Baldonado, Woodruff, Kuchinsky 2000], Table 1, Lines 1 to 4

Following the rule of Diversity the visualization of Web search results is a definite candidate for the usage of multiple views. The main contributing factor is probably the variety of different levels of abstraction necessary to deal with search results ranging from overviews about the whole result set to detailed views of documents and their parts. The rule of Complementarity makes mainly sense when the presentation of different views or visualizations is simultaneously. There may be cases conceivable when different views may help to bring out correlations when dealing with search results from the Web. An example could be the usage of multiple scatterplots each showing the impact of a pair of keywords or concepts. Using multiple reference points to position documents on a 2D-pane and additional coding and / or interaction such as used in SQWID may also help. Because general tradeoffs, the visualization of search results seems not be a strong candidate for the use of multiple views. The same is true for the Rule of Decomposition. There may be benefits, but it is questionable if they are worthwhile exploring because of the possible negative impacts on utility. Whereas the first three rules are based on reasons supporting the usage of multiple views, the fourth rule is based on opposite arguments. One of the points the authors discuss in the context of their Parsimony rule is particularly interesting: "*Further, when two or more views have very similar semantics, the designer should consider merging them into one view*." [Baldonado, Woodruff, Kuchinsky 2000]. During the design process of the INSYDER system we violated this rule by proposing the usage of a Document Vector [Mann 1999], [Mann, Reiterer 1999] which had very similar semantics to the ScatterPlot we used also. User feedback later led to the decision to integrate both views into one component [Reiterer, Mußler, Mann et al. 2000]. Of particular interest are the thoughts of [Baldonado, Woodruff, Kuchinsky 2000] to apply the rule of Parsimony also to the coupling of views, and considerations of how much value to the user such a coupling is adding, compared to the added complexity of the system. During the development of the INSYSDER system there had been long discussions within the team if sorting should be included in the coordination mechanisms of different views. The considerations were discussed in combination with the rule of Consistency. Is the additional value of having a consistent sorting among some of the visualizations worth to have a more complex interface? It was not possible to sort the documents in all components. A scatterplot is one of the examples where it makes no sense to have such a mechanism in addition to the inherent ordering of the values on the axes. Since the INSYDER visualizations are presented sequentially, it is difficult to help the user understand what is happening by utilizing animation, which is often used in such cases. Animation does not reduce complexity, but it helps the user to understand it. In the end sorting in the INSYDER system was implemented as a view-specific mechanism like other authors had done it too (e.g. [Eick 2000]).

Rule	Summary	Major Positive Impacts on Utility	Major Negative Impacts on Utility
Space / Time Resource Optimization	Balance the spatial and temporal costs of presenting multiple views with the spatial and temporal benefits of using the views.	comparison computational overhead display space overhead	
Self-Evidence	Use perceptual cues to make relationships among multiple views more apparent to the user.	learning comparison	computational overhead
Consistency	Make the interfaces for multiple views consistent, and make the states of multiple views consistent.	learning comparison	computational overhead
Attention Management	Use perceptual techniques to focus the user's attention on the right view at the right time.	memory context switching	computational overhead

Table 25: Rules how to use multiple views and areas of major impact on utility according to [Baldonado, Woodruff, Kuchinsky 2000], Table 1, Lines 5 to 8

The guidelines how to use multiple views are difficult to discuss in the general context of the visualization of Web search results. It makes more sense to mirror them on concrete systems. Taking the INSYDER system as an example all four guidelines can be illustrated. A Space / Time Resource Optimization has been done by presenting the visualizations sequentially, considering the fact that the resolution of the users' screen is not more than 1024x768 pixels. The rule of Self-Evidence has been followed by using the same color for a keyword in every component or using the same symbol to represent a document in all views, including the table. This also poses a question of Consistency, but this rule was followed even further by using double-clicking to open a document in all components. To be honest, the guideline of Consistency was also violated in some minor cases because of resource restrictions during the system development. An example is that the same tooltip to show document details was used in all components, except table and list view, where inconsistently no tooltip was present. A basic form of Attention Management was used in the scatterplot component by presenting a number default combinations of mappings from value to axes. However, these have been always the same default combinations.

When talking about Consistency the INSYDER system seems to be not the only system where this guideline is violated due to development resource restrictions, although the authors tried to reach the goal. The Treemap View (page 90) and the Sammon View (page 103) of the Information Navigator system [Au, Carey, Sewraz et al. 2000], [Carey 2000], [Carey, Kriwaczek, Rüger 2000], both show document clusters, and both use a drill down mechanism to allow navigation through the document space. Whereas a drill-down in the Sammon View is a real drill down by reclustering only the documents contained in one or more selected clusters, the "drill-down" in the Treemap view is based on a complete different mechanism. The most prominent keywords of the selected cluster are used to perform a completely new query on the whole document set, leading to a new Treemap containing a set of documents which may be completely different from the selected one. [Carey 2000] mentioned the idea to have both mechanisms in both components. The described implementation poses a challenge for the user to understand the different behaviors when "drilling down".

A prominent example for a multiple view system dealing with the visualization of search results is the system used by [Hearst, Pedersen, Pirolli et al. 1995] in the TREC-4 interactive track. It has a number of elements in common with the INSYDER approach. The presentation of the result sets combined a ranked title display, Scatter/Gather, and TileBars linked together in one system. Possibilities of interaction included selecting a cluster in the Scatter/Gather component and viewing the contained documents in a TileBar display. Selection of relevant documents was synchronized over all views, and displayed by a dark circle close to the title.

Summarizing this chapter about Multiple Coordinated Views it can be stated that in a number of cases the usage of multiple views may provide advantages. The visualization of Web search results may be such a case. Examples of system using Multiple Coordinated Views with different principles, different implemented visualizations, different levels of flexibility, and different usage scenarios range from general-purpose tools to systems especially dedicated to the visualization of Web search results. Designing multiple view systems is more demanding than designing systems without these possibilities.

3.5. Empirical evaluation of visualizations

John V. Cugini summarized the experience of the Information Access Division of the US National Institute of Standards and Technology (NIST) from their work of dealing with interfaces for search results in the following statement: "*One of the lessons of our experience is that no matter how much intuitive appeal a given interface might have, without some systematic testing, its real value remains unknown. Especially in the field of visualization, it is all too common for technical wizardry to be unaccompanied by any real gain in efficiency.*" [Cugini 2000].

Empirical evaluations of visualizations have quite a long tradition. One of the earlier examples from this field is an experimental study by [Washburne 1927], [Washburne 1927a]. In a test with several thousand junior high school children he compared various graphic, tabular and textual methods of presenting quantitative material. The different types of visualizations he used are shown in Figure 99.

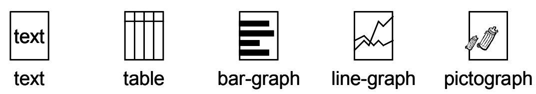


Figure 99: Types of visualizations tested by [Washburne 1927]

Besides visual form and other factors Washburne also varied the logical arrangement of items in a visualization. Despite the fact that a static presentation of quantitative material is in certain aspects far away from nowadays Information Visualization of diverse, often abstract types of information with all its possibilities like animation and interactivity, the findings of Washburne made in 1927 are quite interesting[131]. One of his central questions was: "*Can the forms be given a rank order for* general *effectiveness? That is, is there any form which is more effective in all respects than any other form?*" [Washburne 1927]. Or using the reference model for visualization: Does a best way exist to map raw data over data tables to visual structures and views? "*The answer to this question is simply 'no'.*" [Washburne 1927]. Effectiveness in his studies was depending on type of task, visual form, logical grouping, and number of data. Washburne investigated quite traditional forms of visualization. The lesson learned is that the effectiveness of a certain visualization is depending on a number of factors. We know this at least since 1927. In earlier papers [Mann 1999], [Mann, Reiterer 1999] we argued that a lot of ideas can be found in the Information Visualization literature, but that only some of the recommendations and findings are based on experiments and investigations. And this despite the fact that there are a number of factors influencing the success of a certain visualization. But „*Empirical data involving human users is time consuming to gather and difficult to draw conclusions from.*" [Hearst 1999]. Observing the field it should be mentioned that the situation is changing. The number of empirical evaluations of visualization components is definitely increasing.

[Chen, Yu 2000] made an attempt to give an overview covering the increasing number of empirical studies of Information Visualization features and systems. In their meta-analysis of 35 experimental studies published between 1991 and 2000 they finally compared the results of eight studies. The results are:

- The hypothesis that users with stronger cognitive abilities will perform more efficiently than users with weaker cognitive abilities is supported by [Allen 2000], [Sebrechts, Vasilakis, Miller et al. 1999], and [Swan, Allan 1998] first experiment.
- The hypothesis that visual-spatial information retrieval interfaces will enable users to perform better than traditional retrieval interfaces, is supported by [Allen 2000], [Robertson, Czerwinski, Larson et al. 1998], [Sebrechts, Vasilakis, Miller et al. 1999], and the second experiment of [Swan, Allan 1998]. [Chen, Yu 2000] report that this hypothesis is not supported by [Combs, Bederson 1999]. The later compared four more or less traditional image-browsing tools. This is not really a comparison of visual-spatial information retrieval with traditional IR.
- The hypothesis that users using visualization interfaces in information retrieval will perform more efficiently than their counterparts using a none visualization interface, is supported by [Allen 2000], and [Robertson, Czerwinski, Larson et al. 1998]. Again [Chen, Yu 2000] report that this hypothesis is rejected by [Combs, Bederson 1999]. Please see the second hypothesis for comments.

The discussion of investigations on how users search the Web in Chapter 2.3 showed how difficult it is to compare results from different studies, and how important it is to know the experimental

[131] Reading the paper from Washburne was a nice self-experiment in perception for people like me, reading a lot of HTML-text every day. On the top of page 372 where Washburne explains the line-graph used, there is a legend above the graph, where for each of the three lines the explanation is delivered which guilt is displayed. The guilts are underlined in a form "`... shows how much the Calimala merchants earned.`" For a moment I had the thought that the hyperlink will lead to additional information what "Calimala merchants" are – than I remembered that I was reading a paper from the year 1927.

setting and framework conditions. The short result overview of the meta-analysis of empirical studies of information visualization features and systems by [Chen, Yu 2000] also shows how important it is to look at the details. The goal of this chapter is not to create a detailed comparison of studies or a summary of results. The goals of this chapter are:

- To give some ideas how effective or efficient some of the introduced visualizations ideas have been.
- To recommend literature where to find more information about the evaluation of certain visualization ideas.
- To show which factors influence the success of a visualization idea.

Included are only studies performed with users. Analytical evaluations of visualization ideas such as [Allan, Leouski, Swan 1997], [Leouski, Allan 1998], [Leouski, Allan 1998a] or user evaluations using components not discussed above, such as [Chen, Czerwinski 1997], [Wiss, Carr 1999], [Stasko, Catrambone, Guzdial et al. 2000], or [Risden, Czerwinski, Munzner et al. 2000] are not included. Also not included are general user interface comparisons, even if they deal with the visualization of search results, such as the TREC interactive track (e.g. [Hearst, Pedersen, Pirolli et al. 1995]).

[Hertzum, Frøkjær 1996] performed a study with 87 computer science students comparing the Venn Diagrams (See Figure 28 on page 69) of the TeSS online help prototype with conventional Boolean retrieval, browsing, a combination of the three, and the use of printed manuals. User performed fastest and with the highest quality of the answers in the printed manual setting. From the TeSS modes, browsing was fastest and caused the fewest operation errors. It was followed by the Venn diagrams and then the conventional Boolean retrieval. The combination mode performed worst in the measures of objective performance, but had been preferred by nearly all subjects. It is interesting to mention that browsing had the best average performance of the TeSS modes, but was found to be unsuited for three of the twenty tasks.

[Jones, McInnes, Staveley 1999] compared in a study with 12 university students the Venn Diagram based query workspace (See Figure 29 on page 69) of the VQuery interface with a standard textual Boolean interface. When using VQuery users took significantly longer to form queries and made more errors. The authors attribute this to the necessary three-step process in VQuery and overhead in managing the circles of the Venn Diagrams in the query workspace.

[Bederson, Hollan, Stewart et al. 1998] compared in a first study with 30 students the zooming Web-browser Pad++ with a conventional Netscape browser in different scenarios using a set of 31 Web pages. The users answered questions slightly slower with Pad++ than with Netscape. The authors implemented several changes to the Pad++ Web browser, and than repeated the experiment using 7 developers of the system instead of students and only the condition where Pad++ performed best in the first experiment. The result of the second experiment was that subjects performed better with Pad+ than with Netscape.

[Byrd 1999] compared in a study with 6 college students the FancyV prototype with and without the VQRb enhancement of the scrollbar (See Figure 56 on page 86). Using a number of carefully selected TREC topics, documents, and queries the users had to judge as many documents a possible in five minutes from a result set containing 30 documents. By analyzing the number of documents judged, the number of documents correctly judged, and the accuracy query- and participant-dependent results were significant. There was no significant difference for the objective measurements between the condition with or without VQRb, but the users fairly strong preferred the version with VQRb. The author received the same results when repeating the test with 20 users after fixing some problems.

[Sebrechts, Vasilakis, Miller et al. 1999], [Cugini, Laskowski, Sebrechts 2000] performed a study with 9 university students and 6 professional GUI and / or IR users comparing the 2D Global View (See Figure 34 on page 72), the Concept Globe which is the original 3D-Version of it, and a textual representation. The text condition showed overall the fastest response times, the 3D-version the slowest. The 3D-condition showed the greatest decrease in response time during the experiment. Training seems to be important. In all conditions and tasks, color coding of concepts seemed to have a strong positive impact on efficiancy, at least for up to five different concepts.

[Dieberger, Russell 2001] compared in a study with 12 researchers a horizontal Context Lens, a vertical Context Lens and a textual representation of search results from a pool of 255 resumes. Color highlighting has been used in all conditions. The average execution times with Context Lenses have been shorter than in the condition without Context Lenses. The horizontal version performed slightly faster than the vertical one.

[Eibl 1999] compared recall and precision reached by 8 professional searchers using the “Bracket”-visualization (See Figure 31 on page 71) system and a result set of 30 documents with values from other tests done with Messenger and freeWAIS. The “Bracket”-visualization performed in both metrics between Messenger and freeWAIS, and in a second condition better than the two other systems.

[Grewal, Jackson, Wallis et al. 1999] performed an experiment with 34 users comparing the R-Wheel (See Figure 51 on page 83) with their initial 3D-visualization idea named “tepee”. The task used to compare these two ideas, both showing the contribution of different keywords for the overall relevance, was to order the symbols by overall relevance or to draw the symbols for given figures. In both cases, the R-Wheel performed better than the tepee. In another experiment with 30 users, [Grewal, Jackson, Burden et al. 2000] compared the R-Wheel with Bar-chart, Slider-bar (See Figure 52 on page 83), and a textual representation. Again the task was to order the symbols by overall relevance. The R-Wheel performed best, and was in addition ranked the easiest visualization tool by the users. The authors also performed a number of experiments where the contribution of the distinct keywords mattered [Grewal, Burden, Jackson et al. 1999]. Concerning the overall relevance it would be interesting to know what would have happened if they had used a stacked bargraph, shown as b) in Figure 100 instead of their version shown as a).

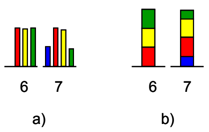


Figure 100: Bar-chart tested by [Grewal, Jackson, Burden et al. 2000] plus untested alternative view

[Hascoët 1998] compared the spiral display (See Figure 89 on page 107), a spring display, and a random display in an analytical experiment and a user study. The analytical result was that the spring display represented the distances in the multidimensional document space better than the spiral or the random display. In the user experiments however, where the task was to identify objects with similar numbers (as substitute for document content), the spiral display performed better than the spring or random display.

[Heidorn, Cui 2000] performed an experiment with 42 users comparing the webVIBE version of VIBE (See Figure 70 on page 96) in combination with a ranked list display with a ranked list display only. Half of the users had been students in a graduate library and information science program. The other half had been from other graduate school departments. The additional reference point display did not help the users in known-item search. Overall, the number of tasks solved was higher in the list only configuration, but the reference point display showed a higher improvement in completion time for later trials. Familiarity with the presentation or in other words training might have been an important factor influencing the results. The authors also looked for correlations between cognitive skills and retrieval performance. Therefore, they also used a design where only half of the subjects had been students of library and information science, because a study by [Allen, Allen 1993] revealed that spatial abilities of students in library and information science are lower then in the general student population, but their verbal scores are higher. In contrast to a previous experiment using VIBE in the webVIBE test no correlation between cognitive skills and retrieval performance had been found.

In a study with 68 users [Lin 1995] compared map displays constructed automatically by a neural network (SOM, see Figure 75ff on page 100ff), manually constructed by human subjects, and a random map. For a known-item search, there were no statistically significant differences between the machine-generated SOM and the two human-generated versions of the map (association-based and category-based). Times spent to locate a title on the random map display have been significantly longer. SOM and association-based maps showed learning effects from first three tasks to the last three tasks, the category-based and the random maps did not.

[Morse, Lewis, Korfhage et al. 1998] performed a paper-and-pencil exercise with 218 members of undergraduate courses in the USA and Norway comparing five types of presentation: ordered text, ordered icons, a table matrix, a simple scatterplot, and a two POI reference point display. All visualizations showed results of a two-term Boolean query. The tasks have been relative simple and been either of the type "*Circle the item(s) that contain term X and Y*" or "*How many items contain the term X?*". The authors recorded task performance (number of correct answers) and user preference rankings of the visualizations. Ordered icon list and text list had the best task performances. Asked for their preferences, user preferred the visualizations icon list and reference point display. The text list was the least desirable form. Gender, age, amount of prior computer experience, or current year in academic program did not affect task performance. Positive influence on overall performance had the level of instructions given. Learning effects during the test influenced the success of the later presented novel visualizations. In a subsequent study [Morse 1999] the number of terms was extended to three and a number of other settings had been changed. The Internet had been used for 191 subjects. 32 users performed the test as paper-and-pencil exercise. The text display showed the contained keywords only instead of the titles. The scatterplot was not included in the test. The results of this three-term Boolean study confirmed the findings from the two-term Boolean study. The users who performed the test via the Internet had the shortest times to completion when using POI reference point display or table matrix. In a further extended study with 195 users [Morse 1999], [Morse, Lewis, Olsen 2000] used two- and three-term queries, a vector-space ranking instead of Boolean conditions, and more different types of questions. Instead of titles, the text display showed keywords repeated according to the number of occurrences. The scatterplot had only been tested in the two-term condition. Whereas the user preference ratings are in general consistent with the Boolean studies, task performance measures showed varying patterns. The POI reference point display had the shortest completion times by having reasonable good scores in the number of correct answers. The icon list that was in fact a bargraph like a vertical implementation of version b) in Figure 100, scored second in completion time with a comparable number of correct answers like the POI reference point display. The study includes a number of interesting detail findings such as the observation, that users who expressed preference of the POI reference point display, received high scores when using it. The order of presentation of the visualizations had a notable effect on time to completion but none on the number of correct answers.

[North, Shneiderman 2000a] performed a study where they examined if users are able to construct and operate coordinated views. Six employees of the US Bureau of Census or students of computer science successfully constructed a user interface with multiple coordinated views using the Snap-Together Visualizations from [North 2000]. 18 students or staff members from the campus participated in a test about the benefits of coordinated visualizations. The authors used an overview-plus-detail scenario with three conditions: detailed list only, overview plus detailed list uncoordinated, and overview plus detailed list coordinated. For tasks where the information from the overview window was sufficient to answer the question, uncoordinated and coordinated overview plus detail performed nearly equal, and both significantly better than detail only. For tasks where information from the detailed window was necessary to answer the question, detail only and uncoordinated overview plus detail performed nearly equal. Coordinated overview plus detail performed significantly better than the other two.

Interesting in the context of multiple view systems is a study by [Allen 2000]. After studying effects of cognitive abilities and design features on search performance, Allen investigated if users optimize the system configuration according to their cognitive abilities and therefore leading to better search performance when they select features. This was not the case.

[Nowell, France, Hix et al. 1996] performed a formative evaluation of the Envision system with its matrix of icons (Figure 67 on page 94) using 5 computer scientists (a faculty member and four students). They compared the performance of the users with the performance of one of the developers and asked a number of questions. The users performed better than the developer and the interface got high positive rankings.

[Veerasamy 1996], [Veerasamy, Belkin 1996] compared in a study with 36 undergraduate students from a course in library searching a retrieval system with and without a bar-graph visualization (See Figure 48 on page 82). The users had been divided into three groups. From 24 topics they had to perform two searches: one group performed both searches with bargraph (w:w), one both searches without bargraph (wo:wo), and one the first search without and the second search with bargraph (wo:w). Because of huge differences in the interpretation of the queries, a second study was performed using the same two topics for all users, and only two settings: (w:w) and (wo:wo). Measuring precision, documents saved per search, interactive TREC precision, and interactive user precision, the authors found no significant differences between the conditions w:w, wo:w, wo:wo in the first experiment. The reasons for this they identified as an insufficient number of test users and great inter-subject and inter-topic variability. In the second experiment, again no significant differences for precision could be found. The other three measures showed a significant superiority for the w:w condition only for the warm-up task performed by the users. In a third experiment with 37 users [Veerasamy 1997], [Veerasamy, Heikes 1997] found that the additional bar-graph helped the users to identify document relevance (in particular non-relevance) quicker, especially when precision was low.

What are the lessons learned from this spotlight on different evaluations of visualizations? The usefulness of visualization ideas is not always given. The traditional benchmark dimensions for Information Retrieval, precision and recall, are sometimes also used to measure the success of visualizations. The usage of these dimensions to judge interactive systems is not without critics [Hearst 1999]. Popular alternative measures for the success of visualization are:

- Time / effort: task completion time, number of steps
- Accuracy: quality / correctness of the answers, error rates
- or changes over time in the above listed values

There appear to be many factors influencing the success of visualizations. Users seem to like visualizations, or at least the want to do experimenters a favor by rating the visualization conditions positively. Hard facts such as effectiveness or efficiency measures sometimes indicate advantages for visualizations compared to textual presentations. In many cases, visualizations perform only equal to, or less successful than their textual counterparts. Different visualizations seem to be different successful in different situations. In the next chapter factors influencing the success of visualizations will be categorized roughly.

3.6. Influencing Factors: 5T-Environment

Independent from the question of visualizations [Marchionini 1997] lists a number of factors influencing the information seeking process. Among them is the information seeker, with his mental models and other factors, the task, the search system, or the domain. As we have seen in the last chapter, there are also a number of factors influencing the success of visualizations. Trying to structure the experiences with different visualization approaches, application areas, taxonomies, and experiments in earlier papers [Mann 1999], [Mann, Reiterer 1999] we proposed the 4T-environment as a classification model. Further investigation of the literature suggested later to introduce a fifth dimension [Mann, Reiterer 2000]. Following this "5T-environment" approach there are five main factors influencing the usefulness of a given visualization:

- Target user group,
- Type and number of data,
- Task to be done,
- Technical possibilities, and
- Training.

Target user group does not only mean a scientist before the screen or a blue-collar worker. There are also interpersonal differences in information perception and processing, which depend for example on the way people think in spatial dimensions. For further exploration of this point in addition to the studies listed above see for example [Egan 1988], [Borgman 1989], [Shneiderman 1998], or [Hearst 1999].

The **type of data** to be displayed is essential for choosing a graphical representation. If there is e.g. a hierarchy in the data it makes a sense to exploit this for visualization. But it is not only the type, but also the number of data that influences the success of visualization. Examining fifty documents represented as tilebars may be very satisfying to find the most relevant ones, doing this with 5000 documents the user will probably like to have a refinement step with another form of visualization.

The **task to be done** is also a very important factor influencing the effectiveness of a chosen visualization. There are a considerable number of attempts to classify or rate visualizations for different forms of tasks, with a wide variation of the level on which "tasks" are defined. For a good discussion how important the type of task is for the success of a certain visualization see [Casner 1991].

The **technical possibilities** are also a determining factor for utilization and success of a visualization idea. Example for such technical factors are the choice to use a Web browser based user interface or the usage of a slow computer leading to bad performance values for a visualization.

Training or experience seems also an important factor influencing the success of visualizations. Like for all other user interfaces it is important to find the right balance between learnability and efficiency [Nielsen 1998a]. But the tradeoff between simplicity versus power [Hearst 1999] is not the only factor that is important here. Especially when comparing visualizations with traditional text representations familiarity with the form of presentation seems to be biasing performance results.

When introducing the evaluation of the visualizations components of the INSYDER system, the 5T-environment will be used to structure the explanation of the experimental setting.

4. INSYDER

4.1. The INSYDER project

This chapter provides a brief introduction to the INSYDER system and describes how it supports information seeking on the Web. The impact of the concepts of visual information seeking on the different visualization of INSYDER will be discussed.

The aim of INSYDER is to find business information on the Web. The main goal of the INSYDER project was to create a solution to supply small- and medium-size enterprises with business information from the Web. To make the information accessible, the basic idea behind INSYDER is a software-plus-content approach. The software is a PC-based local meta-search engine with functions for searching and crawling HTML- and TXT-based information, monitoring changes of retrieved documents, handling news and bookmarks, and last but not least managing all this in a topic-oriented way in Spheres Of Interest (SOIs). “Content” means country- and industry-branch-specific predefined SOIs with selected bookmarks, collections of starting points like search engines and URL-lists, specific thesauri to improve the relevance ranking of the semantic analysis module, or rule files to classify hits by user definable host-types. As a whole INSYDER was created as a country- and industry-branch-specific adaptable system to find, evaluate, filter, manage, and monitor relevant business information from the Web. Because the goal of the INSYDER system is to help to transform data from the Web into information and into knowledge it was also classified as a Business Intelligence system [Reiterer, Mußler, Mann et al. 2000].

The INSYDER project was funded by the European Commission under the Fourth Framework of the Esprit Program, Domain 1, Task 1.9 Emerging Software Technologies. Project No. 29232. Project Coordinators has been Telecom Italia S.p.A. - Servizio Telecom Italia Net, Roma, Italy and Innova S.r.l., Roma, Italy. Project Partners were Arisem S.a.r.l., Paris, France; Cybion S.a.r.l., Paris, France; Promoroma - Azienda speziale della Camera di Commercio di Roma, Italy; and the Universität Konstanz, Germany. Promoroma had two associated additional partners: the Chambre de Commerce et d’Industrie de Meurthe et Moselle, Nancy, France and the Chamber of Commerce and Industry for Bedfordshire & District, Luton, UK. Arisem and the University of Konstanz had developed the software. The other project members provided infrastructure, content, feedback, and part of the specifications.

4.1.1. Functions of the INSYDER system

The INSYDER system comprises three main functions: Search, Watch, and Bookmark / News. These functions can be organized in Spheres Of Interest that can be saved and loaded as user environments. Figure 101 and Figure 102 show two examples of the INSYDER user interface and predefined SOIs provided by the project partners responsible for the content of the system.

The Search function is the part of the system that is the precondition for the visualization of search results. It will be described below. The Watch function allows a monitoring of URLs and documents. Any modification of the documents or the emergence of user-defined terms is monitored and registered in user-defined time-intervals. The idea is to support market or technology surveys in order to detect trends or discover strategic movements. The Bookmark function allows normal bookmarking functionality for URLs. Figure 102 shows a bookmarked page. The Bookmark function was also integrated in the system as the basis for a planned News function. Special Web portals can be integrated in the SOIs as bookmarked Web pages. The portals have been designed as an edited service, to be provided by some of the project partners. The pages are structured as collections of predefined links to national and international daily news. The source for this information is the Internet with its electronic newspapers, magazines, and press agencies.

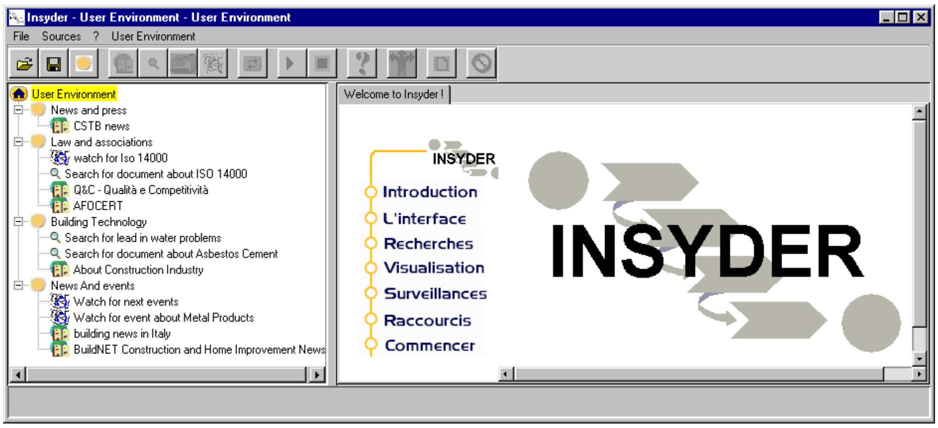


Figure 101: The INSYDER system, example of SOI building and construction

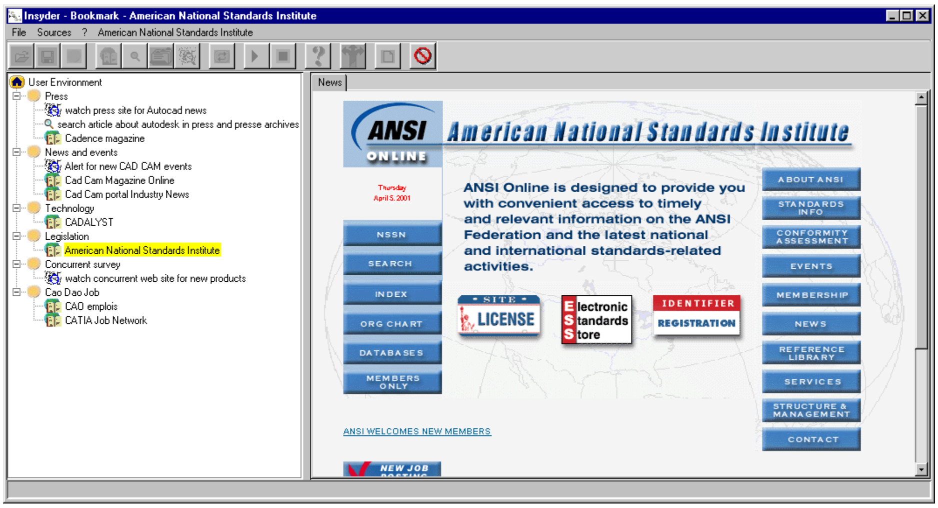


Figure 102: The INSYDER system, example of SOI CAD

As described, the main functions of the INSYDER system cover the areas search, monitoring, bookmarks, news, and administration. Because the visualizations discussed in this thesis focus on the representation of search results, the subsequent list of the system's functions will focus on this area. The following features are implemented in the INSYDER system:

- Searching and loading of HMTL- or TXT-based information from Internet or Intranet.
- Entered search terms are automatically logically ORed. No string search, no Boolean operators, and no proximity functions in the standard search modus (for special functions see [Mußler, Reiterer, Mann 2000], and [Mußler 2002]).
- Any search engine or groups of search engines with URL-controlled interfaces may be used as starting points for a search. In addition, direct specification of URLs or URL lists is possible. In contrast to search engines used as starting points, the directly entered URLs are loaded and analyzed, but in the current implementation are not used for further crawling.
- Own crawling of all links returned by the search engines and further crawling of all links in analyzed documents except documents directly entered as URLs. The exception is a bug rather than a feature (See above).

- Local storage of all crawled documents to allow off-line inspection (without images)
- Concept matching by parsing the entered natural language query in order to extract concepts to match against concepts in the crawled documents.
- Relevance ranking with the use of a thesaurus-based content analysis. No use of rankings from the used external search engines. Thesaurus is at present English and French.
- Automatic classification of host or site type according to the rules of a control file: e.g. “academic” for *.edu, .uni-, .fh-, *.ac.at, …; “european” for *.de, *.fr, *.it, *.at ...; or “competitor” for www.mycompetitor.com, www.meinkonkurrent.de, ...
- Automatic classification of the type of document as “catalog”, “bookmark list”, “text/images”, “frameset”, …
- Determination of the document date (last modified) through analysis of the relevant HTML tags (<META name=“date”... etc.) and the last modified value of the HTTP-protocol.
- Representation of the search results in an interactively configurable and sortable table with the following attributes:
    - Title, URL, document date (last modified), language, size in kB, size in words;
    - Site type (academic, European, …), Document type (catalog, bookmark list, …);
    - Relevance for query, relevance per keyword (concept) in query, 255 characters, query-dependent document extract as a mix of abstract[132] and keywords in context (KWIC)[133];
    - Select flag, relevance feedback flag.
- Visualization of the search results as ScatterPlot, BarGraph, or SegmentView (TileBars or StackedColumn).
- New ranking of already obtained result set after a modification of the query
- Automatic generation of a new query through relevance feedback is possible (find similar, no preference, do not find similar)
- Storage of queries, starting points, and results sets
- Export function for result sets as HTML files with all attributes shown in the table like title, URL, date, extract etc.
- Monitoring of HTML- und TXT-documents for changes or the occurrence of keywords or concepts
- Bookmarking functionality
- Administration of queries and results sets, monitoring jobs, and bookmarks or news in topic-oriented Spheres Of Interest

---

[132] “*An abstract summarizes the main topics of the document but might not contain references to the terms within the query.*” [Hearst 1999]

[133] “*A KWIC extract shows sentences that summarize the ways the query terms are used within the document.*” [Hearst 1999]

Content is in the case of the INSYDER system:

- Country- and industry-specific preconfigured Spheres Of Interest, for example with selected Bookmarks:
- Country- and industry-specific preconfigured lists of search engines and URLs, that can be used as starting points for queries;
- Country- and industry-specific created thesauri for the improvement of the relevance ranking of hit pages;
- Country- and industry-specific preconfigured control files for an automatic classification of hosts or sites in categories.

INSYDER supports the process of collecting, analyzing and classifying unstructured data in documents. For the document analysis and ranking the INSYDER system uses a knowledge base (thesaurus, semantic network). It enables a semantic content analysis of the documents. The INSYDER system thus can find and correctly rank documents also in cases where these do not contain the search words themselves, but contain similar concepts (e.g. synonyms). At present, the thesaurus exists in two languages: French and English. This permits a bilingual analysis of the documents. Accordingly, the system can with an English query, for example find and evaluate documents in French. In addition to the thesaurus for different languages topic-specific thesauri are possible (e.g. for CAD, pharmacy). This enables INSYDER to be adapted to different enterprise needs.

The Spheres Of Interest are representations of the areas in which the user is interested. The user can define various SOIs, e.g. technology, marketing, or competitors. Each SOI is shown as a folder. Within these folders, various searches, watches, and bookmarks can be defined and assigned to a specific interest area. Inside the SOIs, previous searches and watches as well as the current searches and watches and their current status are displayed. Each search or watch is indicated as being currently executed in the background, or as already finished. Moreover, it is possible to create predefined SOIs for the user and deliver them with the INSYDER system. A further possibility discussed has been the subscription to SOIs. An INSYDER system with particular SOIs could then automatically be updated as soon as the provider updates the SOIs. The SOIs represent as well the topic-specific thesauri a further personalization possibility of the INSYDER system. The SOIs also support processes described by [Spink, Bateman, Jansen 1998] as the successive search phenomenon, a process of repeated, successive searching over time.

The INSYDER search is based on a dynamic search approach. The idea is to use an online search to discover relevant information by following links. The main advantage is that the system is searching in the current structure of the Web and not in a possibly outdated index of a search engine. The dynamic search is based on special crawling agents. They use different heterogeneous sources (like search engines, Web directories, Web sites, documents) as starting points for following links. For example, the query terms are submitted to selected search engines and the hyperlinks in the search results are used for further crawling in the Web. Taking the returned hits as a starting point, INSYDER conducts an active search in the WWW. All documents found are analyzed incrementally to find out how well these documents match the query. In this way, the documents presented in the result list can be guaranteed to be up-to-date. Unlike other search systems INSYDER is not designed to crawl the entire WWW and store its contents. Instead, it only crawls selected parts of the Web that seem to be relevant to a given user-query. Every crawled document is then ranked by the INSYDER system. This way of specializing the search by specializing the crawling and ranking is intended to increase the precision compared to other meta-search engines which only rely on the results from the search engines indices.

To start a search with the INSYDER system, the user enters or creates a Sphere Of Interest. In the next step, which is the first phase of the four-phase framework from [Shneiderman, Byrd, Croft 1997] shown in Table 1 on page 12, the user **formulates** his information need as unstructured text (often called a some what misleadingly “natural language”) and chooses from a list sources as starting points for the search (e.g. Web sites, search engines). In the subsequent **action** phase, the search is launched and run until the user stops it. During or after the search, the user may do a **review of the results,** i.e. look at the documents. A Web-query, even when well-focused, can produce so many potentially useful hits as to be overwhelming, i.e. several hundred or more. Recent work in visual information-seeking systems, capitalizing on general information visualization research, has dramatically expanded the limited traditional display techniques (e.g. ranked list of hits). Accordingly, a variety of information visualization techniques displaying search results has been integrated in the INSYDER system. All visualizations simply try to make the result set of documents easier to handle. The **refinement** of the search is supported by relevance feedback. A detailed description of this and other used information retrieval approaches in the INSYDER system (e.g. weighted search terms, semantic analysis) can be found in [Reiterer, Mußler, Mann et al. 2000], and [Mußler 2002].

4.1.2. Architecture and Implementation

The implementation of the INSYDER system has been done as a Java/C++ Application for Windows9X/NT. An executable installation of the system consists of the following components: INSYDER itself, the Arisem analysis engine in form of a DLL, various configuration and data files, and an Microsoft MSDE data base (“SQL server light”). The application software INSYDER itself is a Java application executable on the Microsoft Java Virtual Machine. Spheres of Interest, starting points (sources) for a search, the classification patterns of the server types, and several other configuration files are loaded according to the domain configured at runtime. SOIs and starting points, stored as XML-Files, can be edited or extended at any time during runtime. The analysis engine, contributed by Arisem, is a DLL coded in C++. It is connected to the Java application via a COM-bridge. The knowledge base used for the content analysis is loaded according to the domain configured at run-time. The Microsoft MSDE database serves as a repository for the administration of the hit lists and the results of analysis. It is connected to the Java application via a JDBC/ODBC-bridge. The crawled documents are automatically stored in the windows file system. Figure 103 shows an overview of the system architecture. A detailed description of the components follows below.

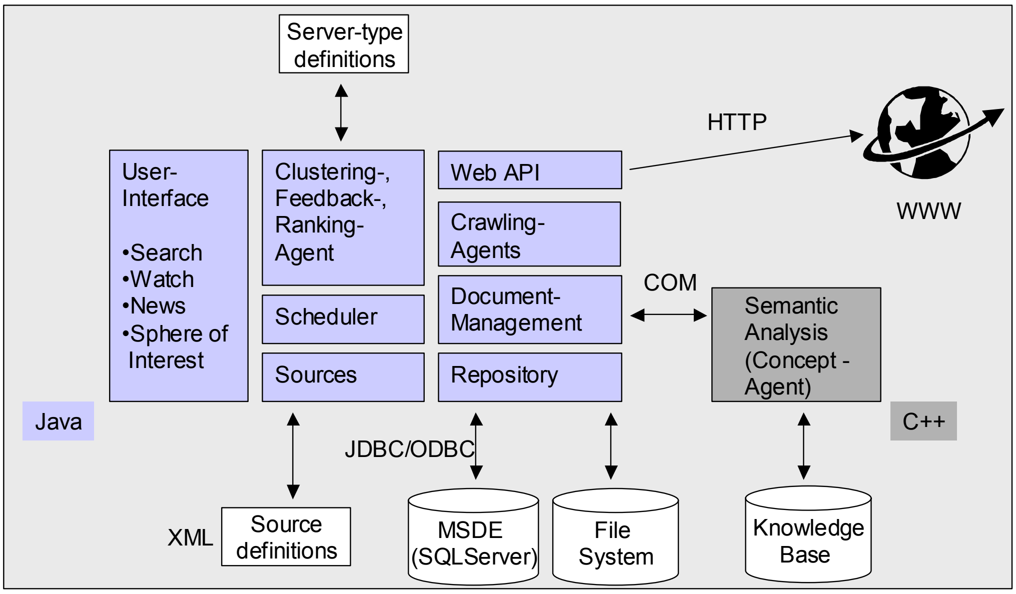


Figure 103: INSYDER Architecture

The **user-interface** and all visualizations have been developed in Java using the Swing JFC (Java Foundation Classes). Swing allows for the decoupling of the data and the different visualization views through its inherent support of the Model Viewer Control (MVC)-concept.

The different **agents** are responsible for special retrieval tasks (e.g. crawling the Web; clustering and ranking of the search results; preparing the relevance feedback for a new crawling).

The **scheduler** is responsible for the Web monitoring process (called watch function). The watch-function is able to regularly check user-defined Web pages for changes.

The **sources** are the representation of starting points of a search, such as URLs or commercially available search-engines. All sources are defined in XML documents, which enables easy maintenance and extension of the sources in a standardized format.

The **Web-API** is a set of functions and methods, which supports an easy access to the documents of the Web. The crawling agents use the Web-API for searching and crawling for Web documents, downloading them, and putting them into the document management component.

The **document management** component is responsible for the management of all documents and their metadata. It is the central component of the architecture and implements the classes and methods for the other components, e.g. when the user interface wants to access a certain document.

For every document the document-management calls the **semantic analysis** C++-module via a COM wrapper to get a relevance value. A semantic network is used that models a controlled vocabulary based on a thesaurus. The semantic network can be individually adapted to various application domains (e.g. building and construction; computer industry). It consists of concepts (nodes) that describe the semantics of the system by using typical relationships (typed links) such as “is-a”, or “consists-of”. With the help of this semantic network it is also possible to find documents that do not contain the terms of the query but contain, for instance, a synonym, acronym, or broader or narrower terms. Another advantage is that the results may be more precise than results from other systems as homonyms can be avoided. For instance, a search for “bank” could result in the financial institute, it could be the computer memory bank, or the bank at the shore. By specifying a domain-specific semantic net, e.g. for the computer industry, INSYDER can determine that bank must relate to a computer and therefore rank results dealing with computers higher than others.

The document-management component stores all retrieved Web documents (without images) in the **file-system** while the corresponding metadata are stored in the **repository** with a link to the Web document in the file system. The communication between the document management component and the repository is made via the JDBC/ODBC interface. The repository uses the MSDE (Microsoft Database Engine), a fully compatible version of the Microsoft SQL-Server RDBMS.

4.1.3. Software development and prototypes

The official INSYDER project had been planned to run from September 1998 to December 1999. Due to delays it was extended until March 2000. The development of the final software system itself started in mid 1999 and continued until February 2000. The development team had members working in Paris, France and Konstanz, Germany. Other than a few face-to-face meetings, the work was coordinated by using CSCW-tools for document sharing and online meetings. Figure 101 and Figure 102 on page 130 present the software development status as of 2000-02-28. Development continued at the University of Konstanz after the official end of the project and is documented in [Mußler, Reiterer, Mann 2000], and [Mußler 2002].

Before the start of the software development in mid 1999, several steps were performed by all project partners including the identification of potential users, an information-needs analysis, a functional analysis of the software system, technical specifications, and system design. These steps were accompanied by the creation of several prototypes. Besides serving as technical feasibility studies, the rationale for the prototypes was to discuss user-interface and visualization ideas with potential users. Figure 104 to Figure 107 show a selection of these prototypes.

Before the final decision was taken to implement the INSYDER system as an application, the intention was to implement a browser-based user interface. The first prototypes to test user interface and navigation concepts were developed in HTML/JavaScript (Figure 104) and in VisualBasic (Figure 105).

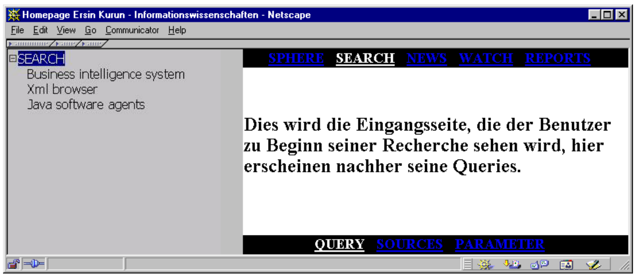


Figure 104: INSYDER HTML/JavaScript-prototype

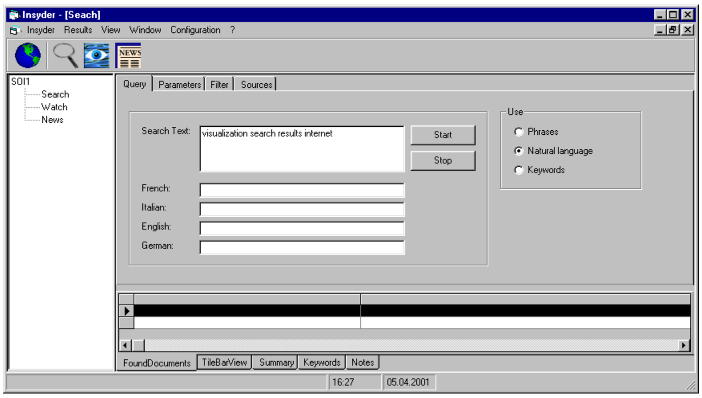


Figure 105: INSYDER VisualBasic-prototype

In a second series of prototypes, which favored still the browser-based concept, visualization ideas were included as a basis for discussions with project partners and potential users. The visualization ideas, the navigation, and the interface concepts were implemented partially in JavaScript like the navigation buttons in Figure 106, partially as Java-applets like the Sphere of Interest in the same figure, and partially as integrated images like the Scatterplot.

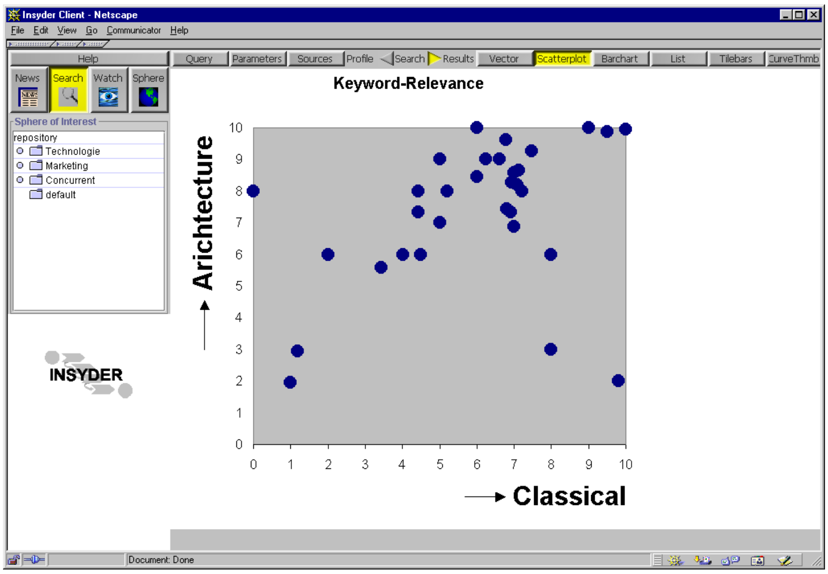


Figure 106: INSYDER HTML / JavaScript / Applet – file- prototype

In the further development process, communication was tested integrating a HTTP-server, instead of using the machines local file system. Figure 107 shows an example from this period. Some visualization ideas, such as a Scatterplot, were integrated as Java-applet instead of as still images. Other visualizations remained as images but came “alive” by combining them with JavaScript components. “Ordering” the elements of a bar graph was, for example, simulated by providing three images and changing them according to the selection of appropriately labeled radio-buttons. These prototypes looked in some ways so realistic that we received an error report from one of our project partners stating that there must be a problem with the search engine, because they always got the same result set regardless of which queries they entered. At that time, we had no search or crawling mechanisms at all.

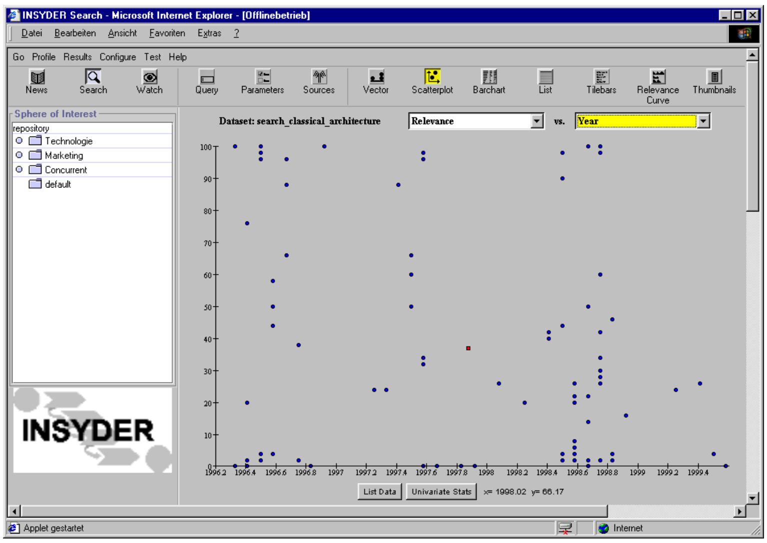


Figure 107: INSYDER HTML / JavaScript / Applet – HTTP – prototype

For several reasons including technical problems and time pressure, the decision was finally taken to drop the browser-based approach, which had been favored by the University, and to use the architecture described in Chapter 4.1.2 and shown in Figure 103. The user interface was therefore ultimately implemented inside a Java-application. Nevertheless, the browser-based period facilitated easy creation of prototypes and valuable discussions concerning navigation concepts and visualization ideas with project partners and potential users.

The specification process of the system was structured by using mindmaps that integrated all the ideas concerning the functions of the system. Figure 108 shows the main branches of such a mindmap. The right side, which represents the functions supporting the information seeking episodes, is structured according to the four-phase framework of information seeking. Figure 109 shows the complete mindmap. Figure 110 shows a sample branch. Details for all branches can be found in the Appendix starting on page 264. Unfortunately, it is difficult to work with mindmaps in DIN-A4 format. When showing branches in readable format as single figures, it is not easy to maintain the overview; when showing the whole structure it is impossible to identify details. The software used to produce and work with mindmaps does provide zooming functions, an overview-plus-detail mode, and details-on-demand functions that help to keep track of the whole structure while working on specific parts. Nevertheless, during the project printouts in table-size format turned out to be a much better representation when discussing the planned functions of the system than the interactive version on the computer screen with overview plus detail functions. In this way discussing the functions of the system was sort of an information visualization study in itself.

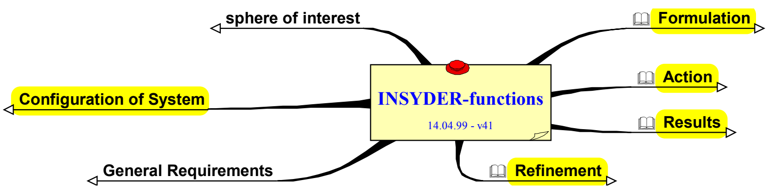


Figure 108: Main branches of the mindmap of planned functions for the INSYDER system

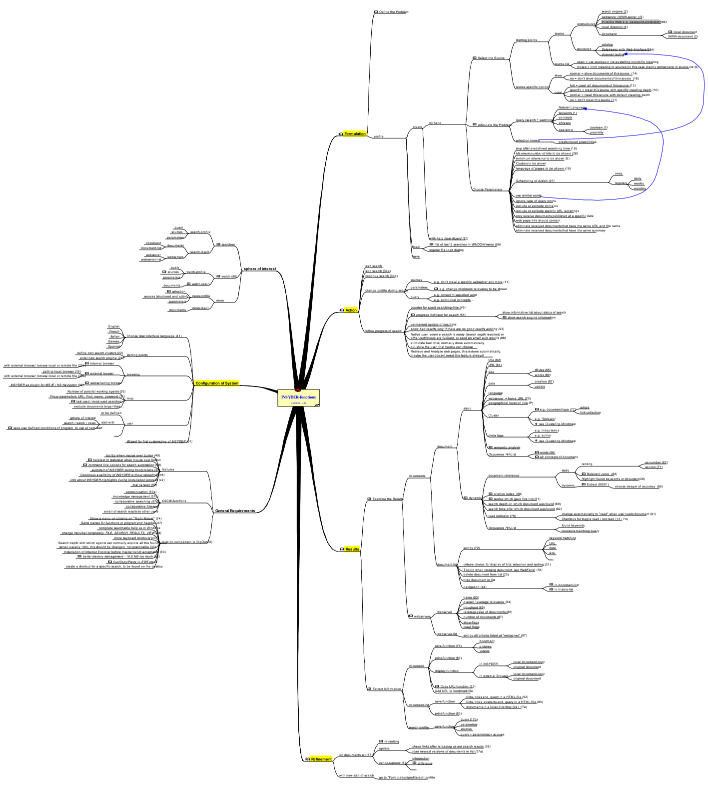


Figure 109: Complete mindmap of planned functions for the INSYDER system

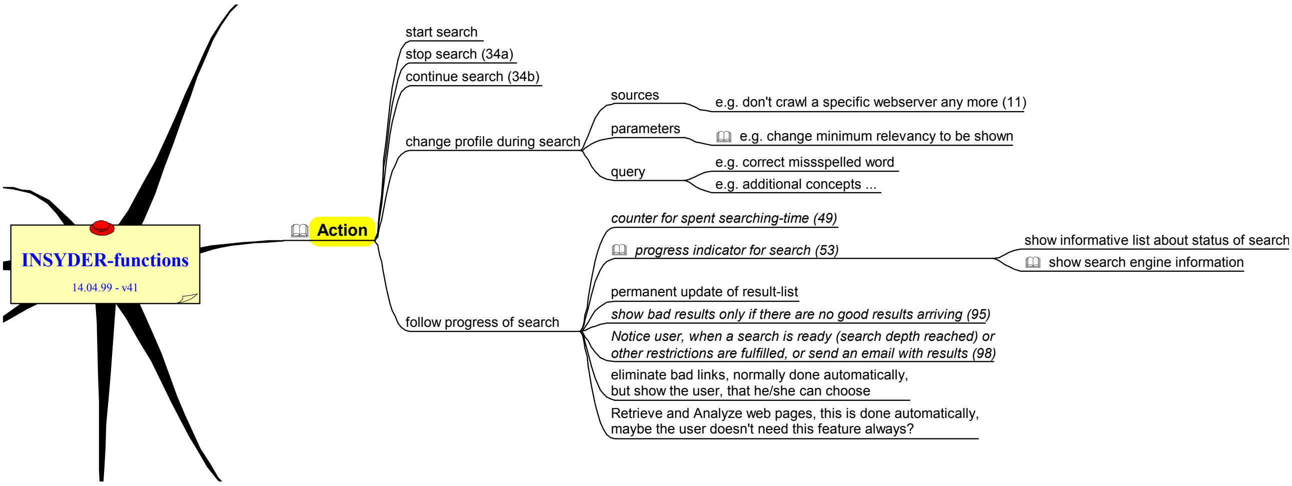


Figure 110: Example branch "Action" from the mindmap of planned functions for the INSYDER system

4.1.4. Formative evaluation during the project

During the official EU project (September 1998 – February 2000), several interviews with potential users and three usability tests (formative evaluations) with users from small- and medium-size enterprises (SMEs) in Luton / Great Britain, Nancy / France, and Rome / Italy were conducted in order to discuss ideas and test the overall system, especially the user interface and the visualizations of the search results. The tests followed the GUIDE-method, as being proposed in [Redmond-Pyle, Moore 1995]. The results were mainly qualitative, but they did influence several design decisions and gave many helpful hints for improving the system. The earlier tests and discussions had been done with the help of the mock-ups and prototypes. In the later phases, pre-versions of the final system were used.

As part of the final usability test with SMEs, the users saw a ScreenCam movie as a short introduction to the system. The users then had to accomplish different test tasks (e.g. create a sphere of interest, launch a search, and analyze documents using different visualizations). During the test tasks, the users were requested to "think aloud", in order for us to be able to understand and record their current actions. The data was recorded with ScreenCam movies (user interactions and spoken comments) and with written records. The session was moderated so that if problems arose, the experimenter could help. In total, 38 companies attended the evaluations (18 companies in Rome, 13 in Nancy, and 7 in Luton). The overall number of users was 48. The majority of the participants had good knowledge a of the Internet, though there were also some beginners. Each user had 45 minutes in which to fulfill the tasks. The user tests have shown that the basic idea of the system, giving the user the possibility to create his user environments and sphere of interests, is appreciated.

An interesting finding in the evaluation of the visualizations was that the test users in Rome preferred the BarGraph view and the ResultTable. ScatterPlot and SegmentView only should be presented as an option. The HTML-List had not yet been integrated. When using the TileBars, it was very important to the users that they be able to jump immediately to a tile of a document by clicking on it. A feature not available at that time. It seems to be the real added value of this visualization and therefore has been included in the final version. The BarGraph was adopted well: only minor problems occurred during its use. As an improvement the users found that it will be necessary to be able not only to sort by global quality or by quality of the single keywords but also by a variable number of keywords. The ScatterPlot view was well understood by most users. This contrasts with results from [Kleiboemer, Lazear, Pedersen 1996], who found anecdotal evidence in a test with several different systems and five users that the scatter-plot display was perceived as confusing. Additional very helpful feedback from our users was that the VectorPlot view could be integrated in the Scatterplot by adding predefined views.

4.2. The INSYDER visualizations

4.2.1. Ideas behind the INSYDER visualization components

The motive for the use of visualizations beyond pure presentation in list form was to improve access to the abstract result sets from WWW-searches following the classic goal of "Information Visualization". The final implementation of the INSYDER system included five components for the presentation of search results: a HTML-List, a ResultTable, a ScatterPlot, a BarGraph, and a SegmentView with two modes: TileBars and StackedColumn. For details-on-demand functions there are also a segment tooltip, a document tooltip, a text window, and a browser. Why were these visualizations chosen for implementation? The main considerations were:

- Focus on the visualization of the search results,
- Multiple Coordinated View approach,
- Orientation on Business Graphics.

In general, the development of the INSYDER system followed a user-centered approach. The visual representations are focused on the review-of-results phase of the four-phase framework, since this is the most interesting one from the user's point of view. Here the user gets the suggestions satisfying his information need, and it would be a good idea to help him find the needle in the haystack by applying suitable visualizations. On the result set level, an overview of all search results to identify which documents fit best with the user's information needs would be useful. On the document level, the user is interested in seeing which parts of a document fit best with his information need. The general design principle was to support the review of the results phase following the visual-information-seeking mantra: Overview first, zoom and filter, then details on demand [Shneiderman 1998]. Besides the visualizations for the review of results phase, that are documented in this thesis, other visual views used in INSYDER support the interaction of the user with the system during the formulation of the query (e.g. visualization of related terms of the query terms by a graph), and during the refinement of the query (e.g. visualization of new query terms based on a relevance feedback inside the graph representing the query terms). These parts of the system are documented in [Mußler, Reiterer, Mann 2000], and [Mußler 2002].

An important design decision for the result phase was to use a multiple view approach. This is in harmony with the rule of Diversity (i.e. use multiple views when there is a diversity of attributes, models, user profiles, levels of abstraction, or genres) from [Baldonado, Woodruff, Kuchinsky 2000]. Knowing that there is no "best visualization" and that the success of a specific visualization depends on several factors including the target user group, the current task, and the type and number of data, we decided to use a combined approach. As shown in Chapter 3.4, the visualization of search results is a natural candidate for multiple view approaches because of the variety of different levels of abstraction necessary to deal with search results ranging from overviews about the whole result set to detailed views of documents and their parts. Multiple view approaches offer the user the possibility to choose the most appropriate visualization view for his current demand or individual preferences. Due to the restrictions involved in running the software on standard business PCs with 17-inch- or sometimes only 15-inch-screens a space-multiplexed approach simultaneously showing different visualizations was dismissed. Instead, a time-multiplexed solution arranging the different components on tabbed panes was implemented. To avoid the possible drawbacks of multiple view approaches several guidelines have been considered. The number of used visualizations has been reduced to a small number. This accords with the rule of Parsimony (i.e. use multiple views minimally) from [Baldonado, Woodruff, Kuchinsky 2000]. Only simple visualizations have been chosen. Feedback from real users has been used to make the final choice and improvements of the selected visualizations. The visual structures have been adapted to each other in color, orientation, and the overall style. The visualizations are synchronized in such a way that a selection in one representation of the result set will be updated immediately in the other representations too. These points are in harmony with the rules of Self-Evidence (i.e. use perceptual cues to make relationships among multiple views more apparent to the user) and Consistency (i.e. make the interfaces for multiple views consistent and make the states of multiple views consistent) from [Baldonado, Woodruff, Kuchinsky 2000]. In addition, grouping of the chosen visualizations around the traditional result list was planned. This should have been the default view, because it is the most familiar one for many users. The visualizations should have been ordered with an increasing level of detail information from the left to the right, with the list positioned in the middle of this row. Figure 111 and some of the prototypes in the last chapter show the initial ideas. These figures contain in some cases visualization ideas not ultimately included in the system. These omitted components are discussed below.

Figure 111: Navigation concept

In the implemented version of the software, the HTML-List or the ResultTable have been kept as the default view, because they are familiar to many users. The ordering with increasing levels of detail from left to right has also been kept, except for the List and the Table that are positioned now at the beginning of the row. Figure 112 shows the final implementation.

Figure 112: Tabbed pane

Not naming it explicitly Attention Management, in keeping with the rule from [Baldonado, Woodruff, Kuchinsky 2000] of using perceptual techniques to focus the user's attention on the right view at the right time, there was a discussion in the project about whether techniques to automatically select the mapping from data tables to visual structures should be used. Automatic selection, as for example by [Andrienko, Andrienko 1997] for the visualization of data with geographical elements, goes even one step further than attention management. This approach was not used for the INSYDER project, because we felt that we are still far from having enough insight into the efficiency and the effectiveness of certain visualizations in certain situations when dealing with the visualization of search results. Instead, the approach chosen was to offer the user different visualizations, that he can select, and if necessary also combine sequentially, according to his current situation.

The visual-information-seeking system INSYDER is not a general purpose system like traditional search engines (e.g. AltaVista). Its context of use is to support small- and medium-sized enterprises of specific application domains finding business information in the Web. Accordingly, the findings of general empirical studies like those mentioned above are in principle useful but had to be supplemented with more specific requirements. At the beginning of the project, a field study was conducted using a questionnaire that has been answered by 73 selected companies (SMEs) in Italy, France, and Great Britain. The aim was to understand the context of use [ISO 9241-11] in keeping with a human-centered design approach [ISO 13407]. The following requirements are based on this field study. The typical users of the INSYDER system are experts from business domains like CAD software or building and construction. These two business domains had been chosen as test areas in the project. Experts from these domains are typically not specialists in using information retrieval systems. They are familiar with the Web and have some limited understanding of search engines. The scenarios show the typical information sources, the typical information needs of the users (e.g. data about new technologies, data about the market, technical regulations, and call for tenders), and the expected functionality (search, monitoring, portal for news). These results correspond very well to an empirical study conducted by [Choo, Detlor, Turnbull 1999], which show that information seekers on the Web typically use a combination of start pages (news or portal sites), a regular check of selected pages (monitoring), and a systematic work through several search engines or meta search engines. Our field study showed that the information needs are normally formulated in unstructured text. The typical technical environments of the users are business PCs. The study showed that the processing power, the RAM, and the size of the screen are limited. It was therefore not possible to use sophisticated 3D visual structures only available on high-end PCs or special workstations. Based on the experiences of the field study, different task scenarios using an information-seeking system like INSYDER to find business information have been developed. The final selection of the visual structures was based on the above suggestions of the field study, an extensive study of the state-of-the-art in visualizing text documents, that is partially documented in Chapter 3.3, and the design goal of orienting our visual structures as much as possible on typical business graphics. The field study showed that all users have a good understanding of this kind of graphics and use them during their daily work (e.g. in spreadsheet programs). Similar conclusions, based mainly on an overview of the research done in the area of visualization of search results in document retrieval systems, can be found in [Zamir 1998]. The author suggested that for a document visualization technique to appear on the Web. Additionally the visualization must be very easy for novice users to understand; it must require minimal CPU time and other resources; and it must be useful for a considerable proportion of searches performed on the Web. Systems that relate the documents to the query terms (like bar charts, tile bars) or to predefined document attributes (like scatter plots) seem to be useful visualization techniques providing additional information about retrieved Web documents.

It was not the intention during the development of the INSYDER system to come up with new visual metaphors supporting the retrieval process. The main idea was to select existing visualizations for text documents and to combine them in a novel way. We tried to select expressive visualizations keeping in mind the target users (business analysts), their typical tasks (to find business data in the Web), their technical environment (typically a desktop PC and not a high-end workstation for extraordinary graphic representations), the type of data to be visualized (document sets and text documents), and minimal necessary training. The major challenge from our point of view was to combine intelligently the selected visualization supporting different views on the retrieved document set and the documents themselves. The primary intention was to present additional information about the retrieved documents to the user in a way that is intuitive, may be quickly interpreted, and can scale to large document sets.

4.2.2. INSYDER and the reference model for visualization

For the discussion of the use of the design principles followed during the development of the visual information seeking system INSYDER the reference model for visualization [Card, Mackinlay, Shneiderman 1999] will be used.

The raw data of the INSYDER system is potentially all Web documents. In Chapter 4.1.2, the general system architecture of the INSYDER system used to handle this data was introduced. One of the system design decisions was to build a multi-agent based meta searcher. After the formulation of the query, the Web documents are collected, analyzed, classified, and ranked with the help of different retrieval agents named crawling, classification, and ranking agents. The output of these agents is the retrieved, classified, and ranked search results of the query. The first **data transformation** step is to transform and save all the search results (Web documents) and their characteristics either in a local repository (MSDE RDBMS) with a specific data schema (metadata), in the file system (document itself), or in the Sphere of Interest (query, sources). A small number of document attributes are not stored permanently but are calculated on the fly when necessary. Each document found for a query is a specific case and will be characterized by predefined attributes and the data type of the attributes. Table 26 shows an overview of the document attributes used in the INSYDER system. Table 27 shows their data types, their usage in the visual structure, the processing location, and their storage.

Dependency	Attributes
Document (fixed)	Title, URL, Size in kB, Fulltext
Document (processed)	Size in words, Size in segments, Date (last modified), Language, Document type (catalog, bookmark list, …), Stripped Text, Stripped Text per Segment
Host	Site type (academic, European, …)
Query	Relevance for query, Relevance for query per segment, Relevance per concept, Relevance per concept and segment, Document extract (255 characters)
User interaction	Select flag, Relevance feedback flag
System	Document ID, Storage Date, Local path and Filename

Table 26: Overview of the document attributes used in the INSYDER system

Variable	Data Type	List	Table	ScatterPlot	BarGraph	SegmentView	Segment Tooltip	Document Tooltip	Text window	Browser	Processed	Stored
Document ID	Nominal	I	I	I	I	I	-	-	-	-	MSDE	Database
Storage Date (local)	Quantitative	-	-	-	-	-	-	-	-	-	Java	Database
Title	Nominal	V	V	I	I	V	-	V	-	-	-	Database
URL	Nominal	V	V	I	I	V	-	V	-	-	-	Database
Size in kB	Quantitative	V	V	V	-	-	-	V	-	-	-	Database
Size in words	Quantitative	V	V	V	-	-	-	V	-	-	C++	Database
Size in segments	Quantitative	-	-	-	-	V	-	-	-	-	C++	- (runtime)

Variable	Data Type	List	Table	ScatterPlot	BarGraph	SegmentView	Segment Tooltip	Document Tooltip	Text window	Browser	Processed	Stored
Date (last modified)	Quantitative	V	V	V	-	-	-	V	-	-	Java	Database
Language	Nominal	V	V	V	-	-	-	V	-	-	C++	Database
Document type	Nominal	V	V	V	-	-	-	V	-	-	Java	Database
Server type	Nominal	V	V	V	-	-	-	V	-	-	Java	Database
Relevance for query	Quantitative	V	V	V	V	V	-	V	-	-	C++	Database
Relevance for query per segment	Quantitative	-	S	-	-	-	-	-	-	-	C++	Database
Relevance per concept	Quantitative	V	V	V	V	-	-	V	-	-	C++	Database
Relevance per concept and segment	Quantitative	-	-	-	-	V	-	-	-	-	C++	- (runtime)
Select flag	Nominal	V	V	V	V	V	-	-	-	-	Java	Database
Relevance feedback flag	Nominal	-	V	-	-	-	-	-	-	-	Java	- (runtime)
Local path and filename	Nominal	I	I	-	-	-	-	-	-	-	Java	Database
Fulltext	Nominal	-	-	-	-	-	-	-	-	V	-	File system
Document extract	Nominal	V	V	-	-	-	-	V	-	-	C++	Database
Stripped Text	Nominal	-	-	-	-	-	-	-	V	-	C++	- (runtime)
Stripped Text per Segment	Nominal	-	-	-	-	-	V	-	-	-	C++	- (runtime)

Table 27: Data Table of the Documents

“V” = Visible by text, position, color, …; “I” = Invisible, but used for interaction; “S” = Special Relevance Curve in Table; “- (runtime)” = not stored, but processed during runtime

Every document processed in the analysis engine on the C++ side is structured in segments. In most cases, a segment corresponds to a sentence. If the number of segments of a document exceeds a threshold given as a parameter to the analysis engine, segments are grouped in super-segments such that the whole document is splitted into several pieces according to the threshold value. In other systems from Arisem S.A. Paris, these segments and their individual ranking are used to calculate a Relevance Curve and the Document extract (See Figure 43 on page 78). In the INSYDER system, they are also used for the SegmentView.

The document date (last modified) is calculated through analysis of the relevant HTML tags (<META name=“date”... etc.) and the last modified value of the HTTP-protocol. The easiest way to find information about the age of a document in the Web is to use the last modified data provided through the HTTP-protocol. Discussions and tests during the development of the system revealed that this data would often not mirror the real age of the document for several reasons. Problems included HTTP-servers, that when asked for the last modified value of a document always deliver the current day or zero corresponding to the year 1970, regardless of the real last modified date of the document. Accordingly the idea was hatched to calculate the document age in a three step process: by taking the last modified information from the HHTP-protocol, by looking for the relevant HTML-tags dealing with document creation dates, modification dates and the corresponding Dublin Core elements, and by a semantic analysis of the document looking for information in the text like “last edited …”. It then has to be decided which of the calculated values will

be presented to the user, because we wanted to present only one value to ease understanding. Due to development resource restrictions only the HTTP and the HTML/Dublin Core steps have been implemented. The system now checks the documents for relevant HTML-tags. If they are not present, which is the case for the great majority of the documents, or not in a plausible range between the year 1970 and the current day, the last modified value from the HTTP-protocol is examined. If it is inside the plausibility range, it is taken. If it is before 1970, the year 1970 is taken. If it is in the future, the current day is taken. Additionally there are some extra algorithms for example to catch up Y2K-problems in the HTTP-protocol or user formatting errors in the HTML-tags. A year value of 100 delivered is handled as 2000, 101 as 2001 and so on. Or the HTML-tags are examined a second time when they do not use ISO-8601-format for the date but something else. Despite all these mechanisms experience showed that from 0% to over 50%, with typical values around 30%, of the documents of a result set in INSYDER have a last modified 1970-01-01 (data taken from the result sets collected for the evaluation in February 2000). In the examples of the INSYDER system below, please note that the format used to display dates at the user interface is not according to ISO-8601 but depends on the general user settings of the PC where INSYDER is run. In the examples, it will be a German format.

The language of a document is detected by the semantic analysis engine. Basis for the detection are typical words of languages stored in the knowledge base. Due to the fact that the INSYDER project has been focused on English and French and to the exclusion of other thesauri, the language detection only works correctly for these two languages. All documents in other languages are either categorized as English or French. Because semantic analysis is used for the relevance ranking of the documents, the language detection plays an important role for the mapping from keywords to concepts and the ranking of documents.

The document extract presented to the user is query dependent. It is up to 255 characters long and a cross between an abstract and a KWIC extract as defined by [Hearst 1999]. The extract seeks to summarize the main topics of the document and presents sentences or parts of sentences that show the ways the concepts behind the query terms are used in the document. The exact algorithm is a company secret of Arisem S.A. Paris. Table 28 shows three sample extracts and the corresponding queries. The document used for the example is [McCrickard, Kehoe 1997].

Query	Document extract
visualization search results internet	Visualization can be particularly useful in interpreting Web search results for several reasons. [..] This paper discusses several systems that use these visualization techniques and introduces the SQWID (Search Query Weighted Information Display) tool, a
mccrickard kehoe georgia atlanta	D. Scott McCrickard & Colleen M. Kehoe Graphics, Visualization, and Usability Center [..] Georgia Institute of Technology Atlanta, GA 30332 [..] SQWID is implemented in Java and runs locally at Georgia Tech as a Java applet under the HotJava browser.
hypertext java query graph	This paper discusses several systems that use these visualization techniques and introduces the SQWID (Search Query Weighted Information Display) tool, a graph-based system developed to illustrate how these techniques can be used to visualize Web search r

Table 28: Sample of document extracts and corresponding queries from the INSYDER system

The stripped text used inside the analysis engine and in a special text window is a version of the document where all HTML-tags are removed. The fulltext is the original HTML-document.

The data schema for each document has been shown in the **Data Table** in Table 27. There is also a Data Table for each query available shown in Table 29. The Data Tables show all variables, the visual structure where the variable will be used, processing module, storage place, and the data type which is important for the mapping on visual structures and different controls to interact with the visual structure.

Variable	Data Type	List	Table	ScatterPlot	Bargraph	SegmentView	Segment Tooltip	Document Tooltip	Text window	Browser	Processed	Stored
Number of documents	Quantitative	V	V	V	V	V	-	-	-	-	Java	Database
Keyword 1 of query	Nominal	V	V	V	V	V	-	-	-	-	-	SOI
Keyword 2 of query	Nominal	V	V	V	V	V	-	-	-	-	-	SOI
…	Nominal	V	V	V	V	V	-	-	-	-	-	SOI
Keyword n of query		V	V	V	V	V	-	-	-	-	-	SOI
Concepts derived from keywords	Nominal	-	-	-	-	-	-	-	-	-	C++	- (runtime)
Sources	Nominal	-	-	-	-	-	-	-	-	-	-	SOI

Table 29: Data Table of the Queries

The main idea behind our visual information seeking approach is to present additional information about retrieved documents to the user in a way that is intuitive, may be quickly interpreted, and can scale to large document sets. Unfortunately, several problems arose in mapping the raw data to data tables, and the data tables to visual structures. Some had to do with the raw data itself, and some with our architecture to map the raw data to data tables or the general concepts of the INSYDER system. An example of a problem with the available raw data itself is the last modified date of the documents discussed above. An example of a problem with the architecture and general concepts of the system is the difficulty involved in implementing query term highlighting. Highlighting and color highlighting of query terms have been shown to be useful and important features of information access interfaces in several cases [Hearst 1999]. Query term highlighting was from the beginning of the design of the INSYDER system a planned feature. Nevertheless, it was not been included in the system. As described above, the semantic analysis engine for the ranking of documents and segments uses not only the entered keywords itself but also synonyms, acronyms, and broader or narrower terms. For example, it could therefore happen that a segment of a document receives a high rank for the concept "internet" derived from the keyword "internet", despite the fact that the segment does not contain the keyword. Figure 124 on page 153 shows such an example, where the segment "*Interactive user interfaces, information navigation, interaction techniques, World-Wide Web, Mosaic.*" got a high rank for the keyword "internet". Due to the fact that the analysis engine is encapsulated on the C++ side and not all details of the analysis are exposed at the COM-interface, the Java user components receive no information which of the terms in the segment contributed to the high ranking for "internet". An inspection of the thesaurus may reveal the types of connections between "internet", "World-Wide Web", and "Mosaic". Neglecting the exact types of connection, let's assume that there is a strong connection between "internet" and "World-Wide Web" and a somewhat weaker connection between "internet" and "Mosaic". Both terms may have contributed to the ranking. Which one should be highlighted? Both? In the same way? In different ways? The question is hypothetical because the information is in the current implementation not available at the user interface level. Figure 113 shows this fact mapped on the reference model of visualization. But even when it is available, the question will remain as to how to map this detail information from the highly sophisticated ranking mechanism to a simple query term highlighting feature.

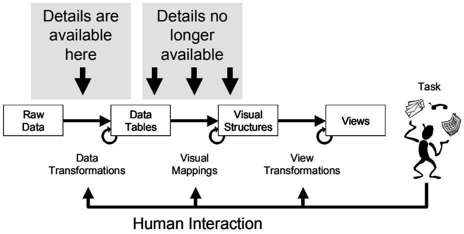


Figure 113: Availability of term ranking details in the INSYDER system

There were several other problems influencing the visual mappings or the potential success of the visualization components of the INSYDER system. These problems will be discussed below in the context of the visualizations in which they occurred.

The next step in the development process after the mapping from raw data to data tables was the **visual mapping** of the data tables to good visual structures, which augment a spatial substrate with marks and graphical properties to encode information. To constitute a good visual structure, it is important that this mapping preserves the data. Some aspects of this second mapping step were mentioned in the last chapter. Our goal was to find expressive mappings for our *target users*, their *typical task*, their *technical environment*, the *type of data,* and the *training.* The rationales behind our selections and decisions have already been mentioned, in particular in Chapter 4.2.1, or will be mentioned when discussing the visual structures in detail below.

View transformations interactively modify and augment Visual Structures in order to turn static presentations into visualizations by establishing graphical parameters to create views of Visual Structures. During the development of the INSYDER system, we decided to use the following different view transformation techniques.

Location probes are view transformations that use location in a Visual Structure to reveal additional data table information. The components of the INSYDER system support five main location probes mechanisms which use a Document Tooltip, a Document-group Tooltip, a Segment Tooltip, a Text window, and a Browser. In the ScatterPlot, the BarGraph, the TileBar, and the StackedColumn a tooltip reveals details about the document mapped to the symbol or bar. Details include the document title, the URL, server type, relevance, date, size, and abstract. Examples can be seen in Figure 120 on page 150 and Figure 123 on page 152. Interesting to note is that in a first version there had been a delay of 500 Milliseconds from the time the mouse cursor begins hovering over an object to the time that the tooltip is displayed. This is a usual delay time for tooltips to appear. Tests by the developers revealed however, that it was quite annoying browsing through a document set to wait 500 milliseconds for the location probe to be activated. The delay time was reset to zero. This speeded up browsing noticeably. At that time, we did not know that other authors had been faced with the same problem and found the same solution. „*A standard tool-tip uses a hover time before the tip is displayed. We determined in a pilot study that the hover time was not effective since it precluded rapid inspection of multiple titles. Hence, the title appears as soon as the mouse moves over a page.*“ [Robertson, Czerwinski, Larson et al. 1998]. Another difference between the INSYDER Document tooltip and usual tooltips is the disappearance behavior. Normally a tooltip disappears after a few seconds of hover time. Because the INSYDER document tooltips contain much more text than standard tooltips, in the first versions it was often annoying to have the tooltip disappearing before finishing inspection. As a result, disappearance was decoupled from hover time and is only triggered when the mouse leaves the area of the object. Location probes using tooltips are also available in the INSYDER system for document groups in the ScatterPlot, where the tooltip shows the titles of the first ten documents included in the group, and for the presentation of the text from a segment in the TileBars or StackedColumn views. Figure 114 shows an example for a document group tooltip, Figure 124 on page 153 for a segment tooltip.

Figure 114: Example for Document Group Tooltip

In addition to the segment tooltip, a text window is used to show users the text of the selected segment in the context of the stripped text of the document (See Figure 125 on page 153 for an example). In all visual structures, a double-click on the visual representation of the document launches the Web-Browser to show the document.

Viewpoint controls are other view transformations that are used to zoom, pan, and clip the viewpoint. Figure 119 and Figure 120 on page 150 show the possibility of zooming into a part of the ScatterPlot. With the help of the right mouse button the user can select the area he wants to zoom in. The pop-up menu offers the zoom in function. If the user wants to step back, he can use the zoom-out button or the full-view button in the zooming group box. The predefined ScatterPlots represented with different radio buttons allow the user to change the viewpoint with one mouse-click. The user can also define his own views, deciding what variables of the Data Table will be shown on the X- and Y-axis. In the ResultTable, the BarGraph, the TileBar, and the StackedColumn, the user has the possibility of sorting the documents by clicking on the headings of the columns or using a drop-down list box.

In all different views, we have made extensive use of different **Interaction** techniques (e.g. direct manipulation, details-on-demand, zooming, direct selection) to give the user control over the mapping of data to visual form.

4.2.3. The INSYDER visualization components

INSYDER offers the option of showing search results in a traditional HTML-format with 30 hits per page. Figure 115 shows an example using the 20 document result set of the WebViz-example. The documents are the same as in Chapter 3.3.3. Please note that the ranking of the INSYDER semantic analysis is used for all INSYDER examples. This ranking differs from the one used for the other examples. In Chapter 3.3.3, a somewhat simpler ranking suitable for the manual creation of figures had been used instead of the highly sophisticated ranking mechanisms of the INSYDER system.

The HTML-List can be used with the integrated browser of the INSYDER system or with an external browser. Common HTML search engine navigation elements as shown in Figure 116 are used when the document set is larger than 30 documents. The HTML-List offers the user a familiar visualization and allows comparisons with usual presentations in common search engines. The document titles are linked to the local copy of the document, the URLs to the original document.

The blue or red point left on the left side of the document relevance represents the status of the Select flag. A red point stands for "selected", a blue point for "not selected". In the HTML-List, the point is a static representation of the attribute. In all other components, it is an interactive element that allows toggling of the select / deselect status.

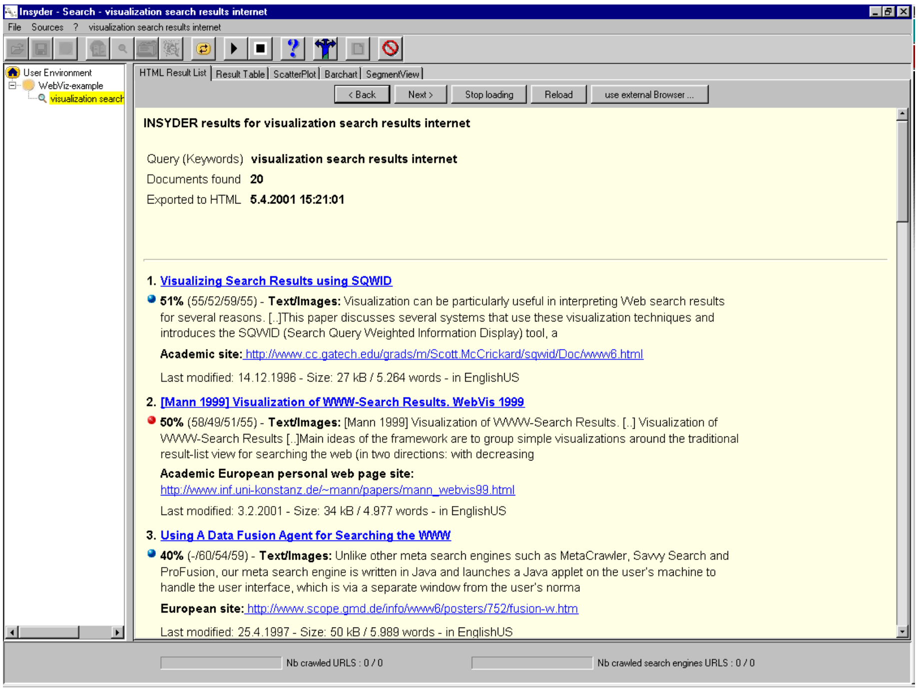


Figure 115: HTML-List, INSYDER integrated browser

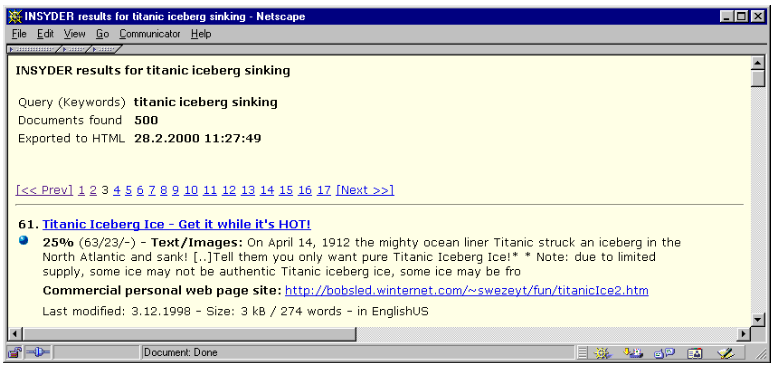


Figure 116: HTML-List with navigation elements, external browser

The second component is a ResultTable implemented in JAVA. Nearly the same attributes as shown in the HTML-List are presented in the columns of a table. Each row shows one document. Figure 117 shows an example using the 20 document result set of the WebViz-example. Relevance Curve and the Relevance feedback flag for each document are additionally displayed, in comparison with the HTML-List. The only attribute not displayed using text form but position instead is the rank number of the document. The user can sort the documents by each variable in an increasing or decreasing order or customize the table to his personal preferences (e.g. to show only the variables he is interested in or to rearrange the order of the columns). On the same pane as the ResultTable, a Browser is integrated, which shows the locally stored version of the currently selected document.

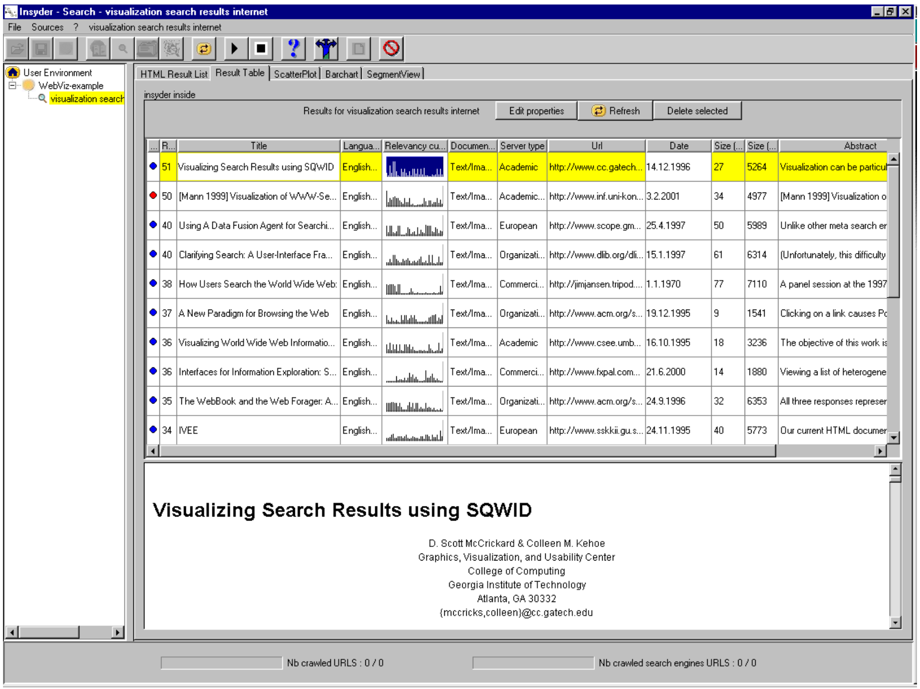


Figure 117: ResultTable from the INSYDER system

The Relevance Curve plays a different role in the ResultTable of the INSYDER system, than in the original implementation where it is combined with the Relevant Extracts of the DigOut4U-system from Arisem (Figure 43 on page 78). In DigOut4 the Curve is mainly used to control the amount of displayed text and to give an impression of the overall relevance and distribution of relevant text segments in the document. In the ResultTable of the INSYDER system, the Relevance Curve also gives an impression of the overall relevance and the distribution of relevant text segments but more crudely. In addition, it may allow a faster recognition of doubles in the ResultTable. The crawling module implemented in the INSYDER system eliminates doubles just by URLs, even though the semantic analysis engine used on the C++ side may have offered much better possibilities. Informal tests by the project team led to the impression that the Relevance Curve will allow fast detection of two identical documents with different URLs, which usually appear close to each other, because they have the same attributes. This impression has so far not been formally evaluated, and, as Figure 118 demonstrates it, will, if ever, be true mainly for adjacent documents.

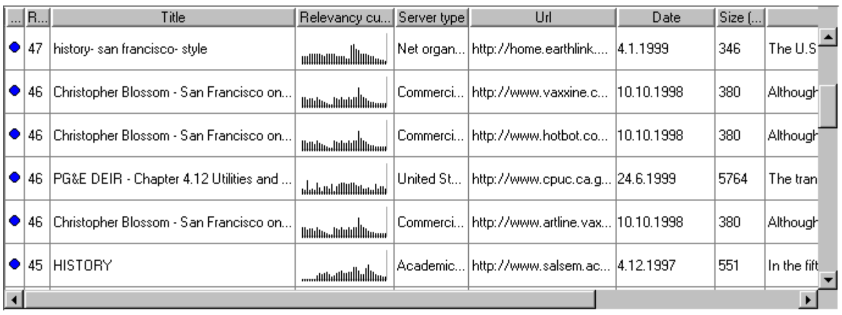


Figure 118: Doubles and Relevance Curve in the ResultTable of the INSYDER system, San Francisco example

Besides considerations about using business-graphic-like visual structures, because business users are the target user group of INSYDER, use of the ScatterPlot was inspired by visual information-seeking systems like the FilmFinder [Ahlberg, Shneiderman 1994], IVEE [Ahlberg, Wistrand 1995], Spotfire Pro [Spotfire 2001], and Envision [Nowell, France, Hix et al. 1996]. In the INSYDER ScatterPlot, each document is represented by a blue or red colored dot. The X and Y dimensions encode two variables. There are three predefined ScatterPlots available, each with a fixed definition of the X and Y dimensions: Date/Relevance, Server type/Number of documents, and Relevance/Server type. The user has also the possibility of selecting his own combination of X and Y dimensions from a subset of variables for each document listed in Table 27 on page 143. During the prototyping phase of the INSYDER project, there were no predefined combinations (See for example Figure 107 on page 136). Instead the user had always to choose sense-making combinations of attributes. Informal tests with the prototypes revealed that it may be a good idea to guide the user by offering a small number of selected, predefined possibilities while still offering power users the possibility of choosing their own combinations. The ScatterPlot thus offers an easy way of navigating through the document space on the set level to find interesting search results. The guidance by predefined combinations also goes in the direction of the Attention Management rule from [Baldonado, Woodruff, Kuchinsky 2000]. Figure 119, Figure 120, and Figure 121 show ScatterPlots from the INSYDER system using the WebViz-example.

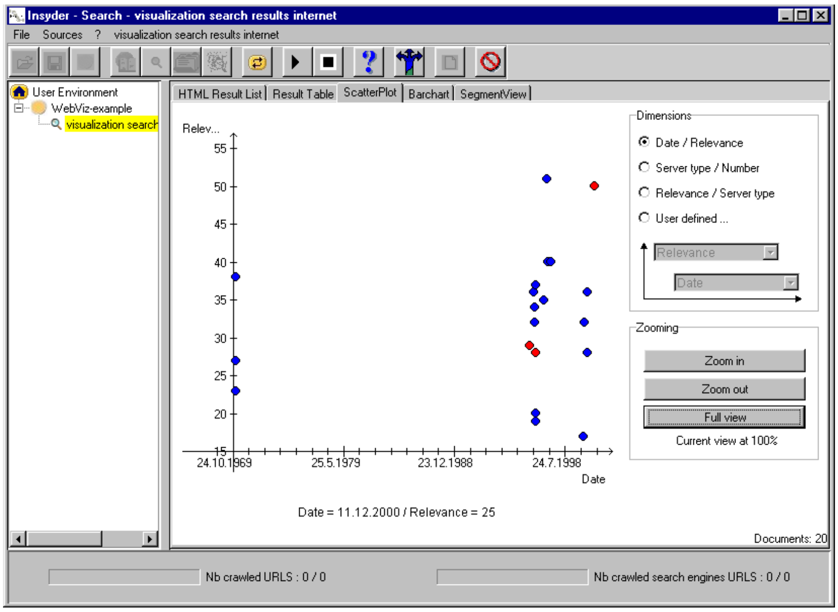


Figure 119: ScatterPlot Date / Relevance from the INSYDER system

Figure 119 shows the typical case described above in which some of the documents have a last modified date of 1970-01-01. You may remember from examples using the same document set in Chapter 3.3.3. that the documents had been from the years 1995 to 1999. In addition to the three documents that seem to be from 1970, the same Figure and the zoom-in in the left part of Figure 120 show that four of the documents seem to have a last modified date from the year 2000 or later.

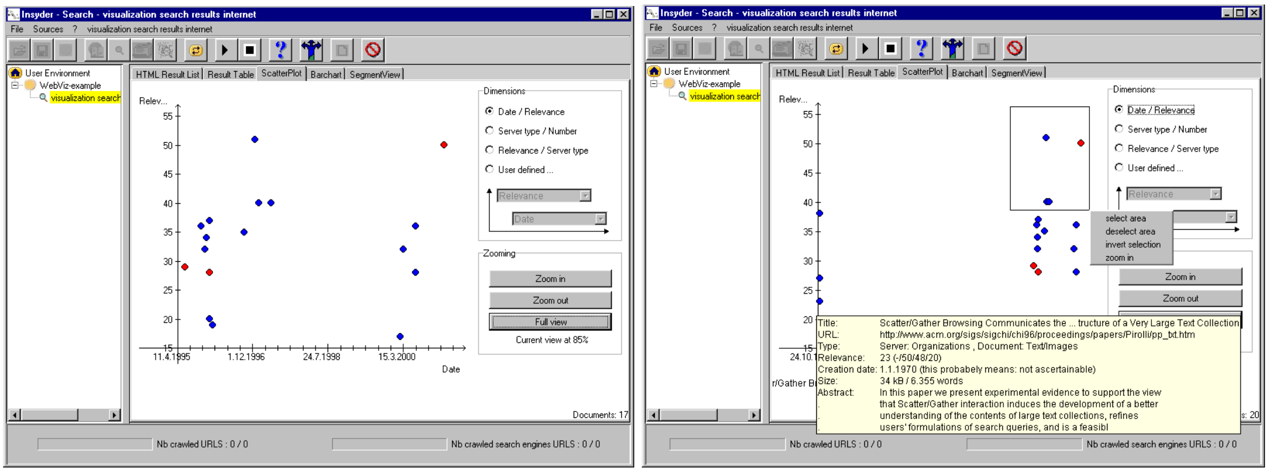


Figure 120: ScatterPlot Date / Relevance zoomed 1995 – 2001, Tooltip and options

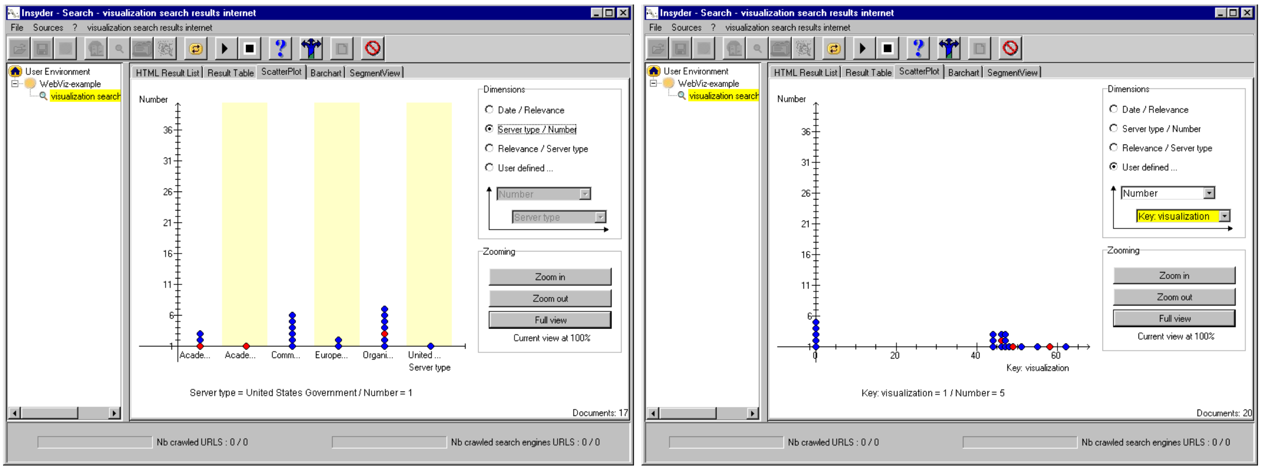


Figure 121: Scatterplot server type (category) / number, and vector mode (keyword visualization / number)

A square-box with a numeric label indicating the number of documents contained represents a document group having the same X/Y-values (e.g. belonging to the same category and having the same relevance). Small document groups with two or three documents are often doubles, as for example all the groups shown in Figure 122. Groups or any interesting single documents, can be selected with the mouse. A single selection is possible with a left mouse-click. Multiple selections are possible with the right mouse drawing a rectangle around the dots or squares. A pop-up menu as shown in Figure 120 appears and the user can select or deselect multiple documents in one step, invert the selection, or zoom into the selection. The selected documents will then be highlighted (selected documents are represented in red, unselected in blue) in this and all the others views, including the next export of HTML-List. The selection can be changed in all views except the HTML-List. Document groups in which not all of the documents are selected are shown in red and blue, as for example the group with two documents on the left side, near the "45", of Figure 122.

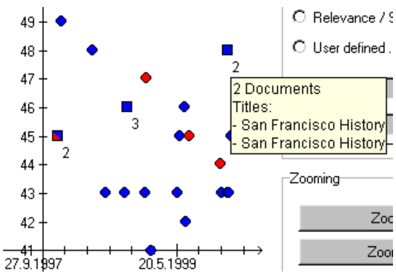


Figure 122: Selected documents (red dots), Document groups (rectangles), San Francisco example

 The use of the BarGraph was inspired by the work of [Veerasamy 1996] / [Veerasamy, Belkin 1996]. The principle behind their visualization has been shown in Figure 48 on page 82. The original idea of bar-graphs, showing overall and single keyword relevance using the length of bars, has been adapted in several ways. First, a horizontal orientation has been chosen. The BarGraph is rotated 90 degrees: top down instead of right to left to have the same vertical orientation displaying the documents as in the other views where document details are given. Second, the impression of a document as an entity is emphasized using Gestalt principles, without disturbing the keyword orientation too much. Figure 123 shows an example using the 20 document result set of the WebViz-example. The colors used for the different keywords are the same as for TileBars and StackedColumns. Each row of bars represents one document and shows the distribution of the relevance for each keyword of the query and the total relevance for the document. It is therefore easy to detect if a document deals with one or more of the different keywords of the query. The headings of each column (Select flag, Relevance, Keyword 1, …) can be sorted in an increasing or

decreasing order. This function offers the user the possibility of viewing the distribution of the relevance of each keyword individually.

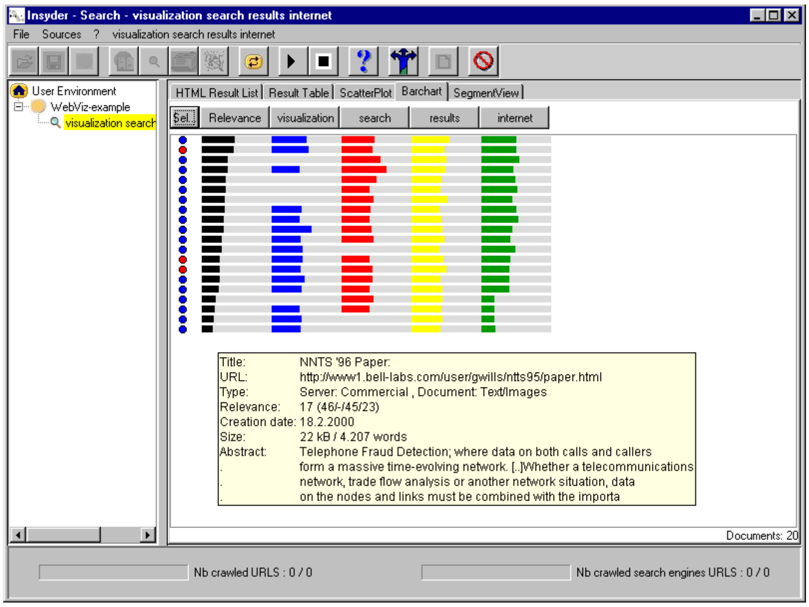


Figure 123: BarGraph from the INSYDER system[134]

Whereas the above-described visualizations aim to show the complete document set as much as allowed by screen space, the SegmentView with TileBar and StackedColumn focuses on single documents. The visual structures TileBar and StackedColumn facilitate a more detailed visual analysis on the document level, whereas the ScatterPlot and BarGraph are helpful on the document set level. The integration of TileBar and StackedColumn into one component is similar to the later discussed integration of DocumentVector and ScatterPlot in keeping with the rule of Parsimony (i.e. use multiple views minimally) from [Baldonado, Woodruff, Kuchinsky 2000]. As mentioned above, documents are broken down into segments for ranking purposes by the semantic analysis module from Arisem. These segments are used in the INSYDER system for the TileBars as well as for the StackedColumns. Both use the same data, but the display differs slightly. For reasons of screen space and performance during the analysis, we limited the maximum number of segments to 100. If a document contains more than 100 sentences, they are automatically grouped in a way that all text is shown but 100 displayed segments are not exceeded. Only 25 segments for a document are displayed at a time, except for one of the StackedColumn variants, where up to 100 segments are displayed. The 25-segment bars have buttons with arrows right and left of each bar to allow vertical scrolling.

The use of TileBars was mainly inspired by the work of [Hearst 1995]. In contrast to the original TileBars, we did not use gray levels to show the keyword relevance for a segment. Instead, each concept is represented with a different color (the same color map as used for the BarGraph and the StackedColumns). Each document is represented by a rectangular bar, which is displayed next to the title of each document. The length of the rectangle indicates the length of the document. The bar is subdivided into rows that correspond to the keywords (concepts) of the query. The bar is also subdivided into columns, each column referring to a segment within the document. Concepts that overlap within the same segment are more likely to indicate a relevant document than concepts that are widely dispersed throughout the document. The patterns are meant to indicate

[134] Please note that the bars for the keyword "results" are difficult to perceive in gray level printouts of this thesis.

whether concepts occur as a main topic throughout the document, as a subtopic, or are just mentioned in passing. The darkness of color (display variant called 3 steps) or the size of the colored area of each square (display variants called 3 sizes or continuous size) corresponds to the relevance of the concept for that segment of text: the darker the color of the square (tile) or the larger the colored area of the square, the higher is the relevance. The different display variants were developed for evaluation purposes (See Chapter 4.3). A white tile indicates no relevance for the concept. The user thereby can quickly see if some subsets of concepts overlap in the same segment of the document. In the original TileBars, the user enters the query in a faceted format, with one topic per line [Hearst 1999]. In the INSYDER system, a single input field for the query is used. No topic grouping is therefore done by the system. Every entered keyword (concept) is displayed on a separate line. Figure 124 shows an example using the 20 document result set of the WebViz-example. It is somewhat atypical that all the documents displayed are longer than 25 segments and therefore require vertical scrolling. The reason is the manual selection of twenty scientific papers, which are longer than the majority of Web documents found in normal INSYDER searches. Figure 138 on page 164 shows some typical examples from the Web.

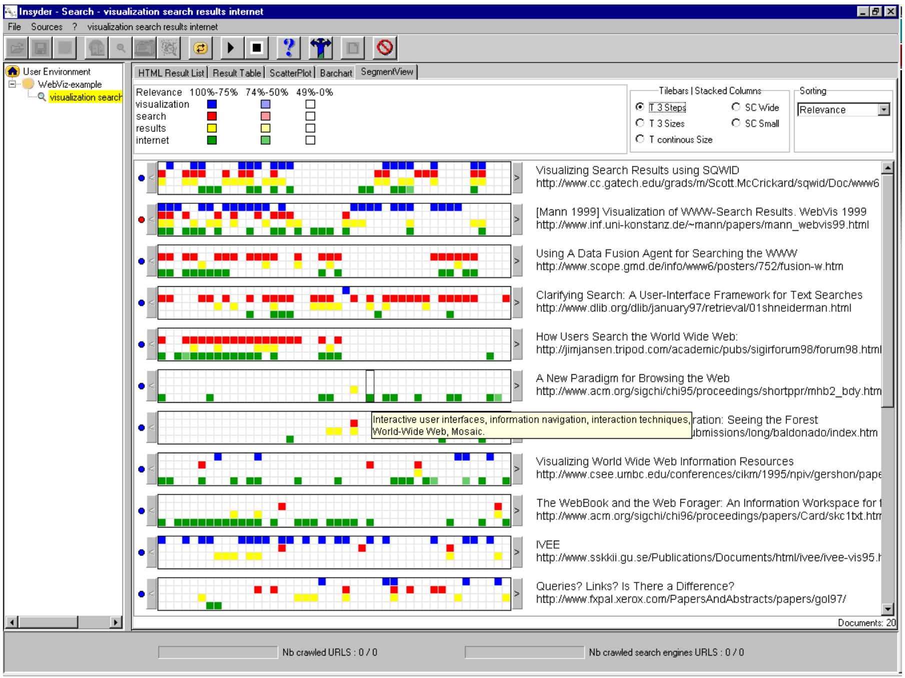


Figure 124: TileBar view with Tooltip

Figure 125 shows the pop-up window that appears when clicking with the right mouse button in the selected segment from Figure 124. The text of the segment is highlighted and put in the context of the stripped text of whole document. The user can now start browsing through the document.

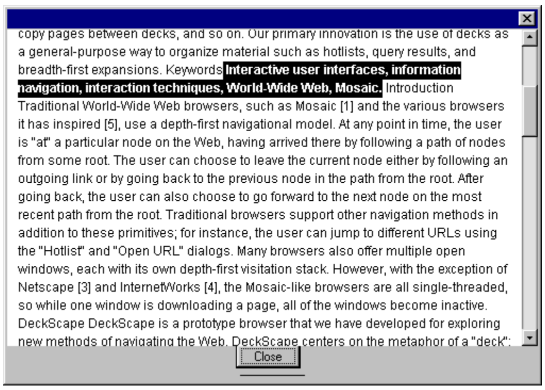


Figure 125: SegmentView - pop-up window

Figure 126 to Figure 128 show the three different display variants of TileBars implemented in the INSYDER system. An evaluation was planned to compare the effectiveness and efficiency of the variants, but it has not been performed so far. Tests with the TileBars revealed that the visualization is highly dependent on the ranking algorithm used to calculate the relevance of keywords or concepts per segment. The selected intervals 0 – 49%, 50% - 74% and 75% - 100% for the three-step variants take this dependency into account. They reflect an idiosyncrasy of the INSYDER segment-ranking algorithm. It was originally tuned to support the creation of the Relevant Extracts, where the size of the extract can be controlled by using the slider of the Relevance Curve. (See Figure 43 on page 78.) Segments with a relevance below 50% are in most cases segments that do not themselves contain the concept but are adjacent to such a segment. In the implementation of the analysis engine that we used, adjacency means only following a segment with the concept. Lowering the slider in the Relevance Curve / Relevant Extracts combination of the DigOut4U system has the effect that step for step more following context of segments which contain the concept is displayed. Without this "adjacency ranking" the segments following may have a ranking of 0% and therefore be only displayed when the slider is at the lower end. Using the ranking algorithm for TileBars cause segment representations to indicate the presence of a concept, though the concept is not contained in that segment, but in the segment before. A threshold of 50% in the three step variants suppresses the display of these unwanted rankings. Comparing the continuous size variant displayed in Figure 128 with the other two variants shows this effect.

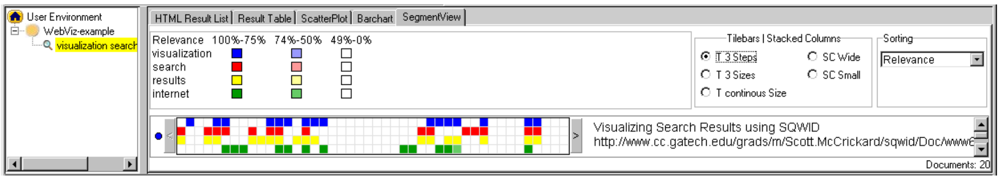


Figure 126: TileBar 3 Steps

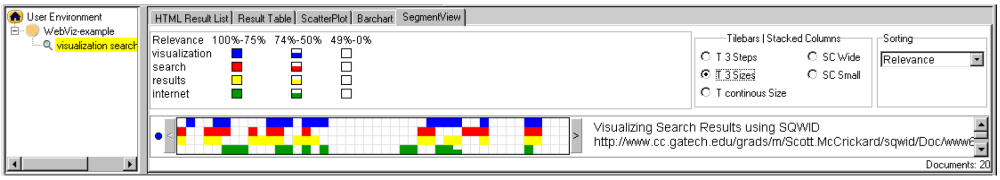


Figure 127: TileBar 3 Sizes

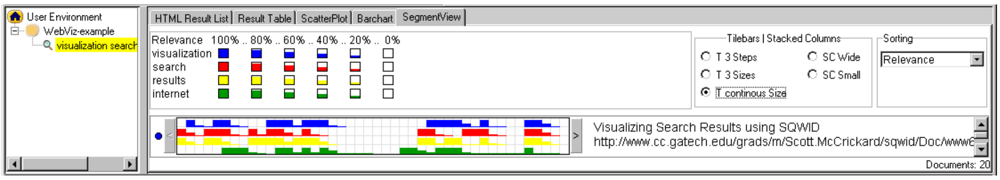


Figure 128: TileBar continuous size

The use of the StackedColumn was inspired by the Relevance Curve from Arisem S.A. Paris, France. Originally it was planned to be integrated in the INSYDER system as a "Enhanced Relevance Curve" in a separate component. Based on the original Relevance Curves some enhancements were planned. First, the number of columns shown corresponds to the number of segments. The original has a fixed number of columns. Second, the original shows only the relevance for the whole query per segment; a colored indication of the single concepts contributions should be added. Third, a show-segment-text-as-tooltip feature was implemented, which is displayed when a segment is crossed with the cursor. Fourth, a jump-to-segment feature was added, showing the document text in a separate window, scrolled to and highlighting the current segments text. At a certain point, we realized that the ideas lead to a component that is very close to the continuous size variant of our TileBars. Accordingly, the Enhanced Relevance Curve was integrated in the TileBar component as a special case. Later, the whole component was named SegmentView, and the Curve variant was named StackedColumn (SC) due to its visual appearance, which did not have much of a "curve". As with the TileBars, we experimented with different versions of the StackedColumn. Each segment is represented as a vertical column. The height of each column corresponds to the relevance of the concepts for that segment. The contribution of the different concepts is shown using the same color map as for BarGraph and TileBars. The first version shows the segments in the same width as the TileBar. This requires vertical scrolling for longer documents. In the second version, we use the same text segment size, but the display is narrowed. All segments of a document can thereby be usually viewed without scrolling. Figure 129 shows the SC Wide variant of the same document as was used to demonstrate the different TileBar variants. Figure 130 shows an example of the second display variant called SC Small.

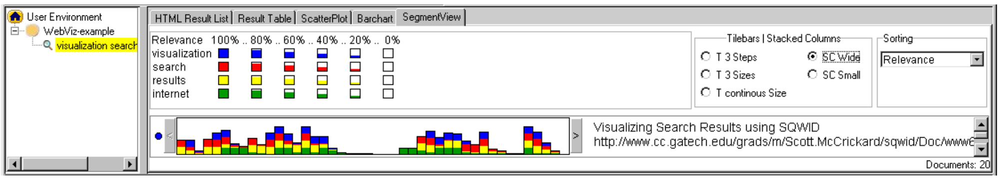


Figure 129: StackedColumn Wide

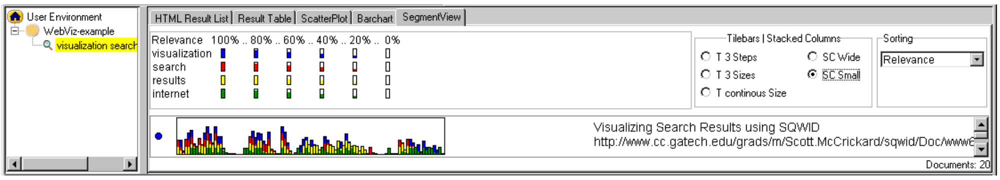


Figure 130: StackedColumn Small

Besides these finally implemented visualizations, several other ideas had been discussed during the specification and development process of the INSYDER system. In informal evaluations using the mock-ups and prototypes user expressed interest in easy-to-understand overviews of the document set found by the INSYDER system. The document spiral idea from [Cugini, Piatko, Laskowski 1997] shown in Figure 86 on page 105 was one of the ideas presented. To reach the goal of having such visualizations, the original document spiral idea was simplified in two ways. First, the representation of the document was explained to users to be just an icon (i.e. no use of colors or little bars). Second, when changing the weight of a keyword, we explained to the users that the icons would be rearranged on the spiral such that the now most important documents for this high weighted keywords would tend to the middle of the spiral and concentrate there. Cugini et al. used elevation to separate user weighting from the relevance calculated by the system. We did not explain the elevation idea; we explained instead that the sliders would change the ranking of the system, because this seemed to us to be more readily understandable by a larger group of users. Our simplified version was very much welcomed by most users (even if we just had a small sample). There was, however, criticism from power users, who remarked that there is a tendency to think that documents near to one other will be in some close relation. A document on the opposite side of the spiral can be much "closer" than a nearby document on the next "ring". This possible "misinterpretation" was later also reported by [Cugini, Laskowski, Sebrechts 2000]. It is easy explainable using Gestalt principles. We decided to transform the spiral to a simple vector which is not so good in using screen space but which does not have the problem of misinterpreted closeness. A drawback, is that the straight line is much shorter than the spiral line. To avoid having too many dots on one point we “stacked” the dots as shown in Figure 131. The Document Vector [Mann 1999], [Mann, Reiterer 1999] was born. It is in fact a sort of a histogram. It is laid out in one dimension. Each document is represented by a dot. If there is more than one point at a column of the scale, the document is displayed by a dot in a second row and so on. The attribute displayed should have been chosen by the user from a list. Example attributes planned were the relevance or the last modified date of the documents. Figure 131 shows the figures used in the prototypes to discuss the idea. As mentioned above, it was ultimately decided to integrate the Document Vector as a special case in the ScatterPlot, where the Y-axis shows the number of documents. The Document Vector as a dedicated component was dead. The right part of Figure 121 on page 151 shows the final implementation.

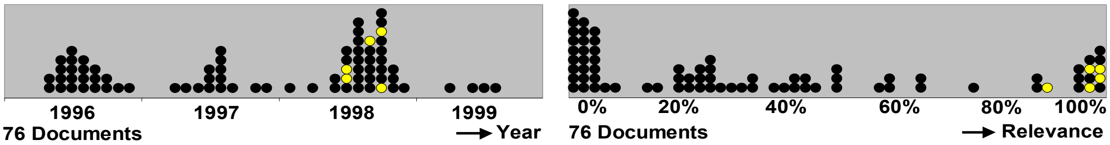


Figure 131: Document Vector

Another component considered for integration was thumbnail views of documents. The idea was to combine them with other visualizations on the document level and with some textual information. The main idea behind representing a document as a thumbnail was to give users who often work in the same document spaces some hints about probably known documents. There may also have been some support for getting a first impression for unknown documents or identifying doubles. A problem with offering thumbnails is the crawling time. For all other visualizations and mechanisms implemented in the INSYDER system it is sufficient to crawl the HTML text file. To produce thumbnails all the images of the documents must be crawled too. Thereby significantly increasing the time for crawling the hits. This would only have been a minimal problem when using the system in an intranet. Because of corporate layout rules, however, the documents of an intranet will often look alike, making the value of thumbnails for an intranet application questionable. Thumbnails were not in the end integrated due to development resource restrictions and severe doubts concerning crawling time. The target users of the INSYDER system typically use modem or ISDN-connections.

The last component idea considered in the INSYDER mock-ups and prototypes was a Keyword-Concept Matrix or Concept Control like that used in the NIRVE system by [Cugini, Laskowski, Piatko 1998], [Cugini, Laskowski, Sebrechts 2000] and shown in Figure 33 on page 72. The main idea was to offer the possibility of simplifying visualizations by mapping different keywords on one concept and therefore one relevance value, color, and row or column. A literature survey of how users search the Web revealed that having too many keywords in a query seems not to be a major problem in Web searching. (See Chapter 2.3.) The feature was left out due to development resource restrictions.

Figure 132 shows the history of the visualizations ideas discussed or implemented for the visualization of elements in the result phase of the INSYDER system.

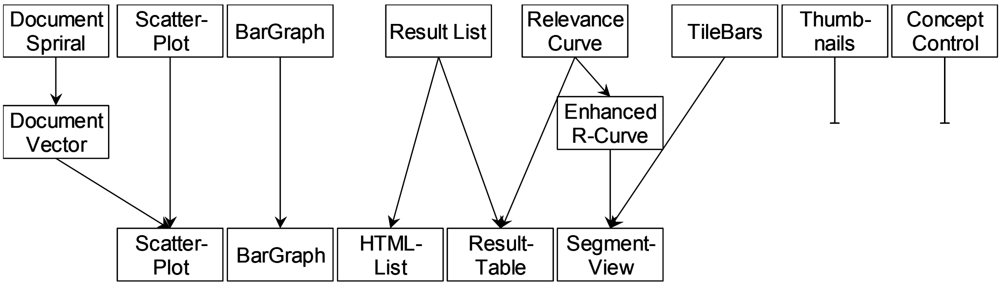


Figure 132: History of the INSYDER visualizations ideas for the result phase

4.3. Evaluation of the visualizations

In addition to the formative evaluations during the project, the University of Konstanz continued the evaluation of the software after the end of the official project in February 2000. An evaluation done with 40 users between February and April 2000 is here described. The evaluation was focused on the different visualizations used to present the search results in the result phase of the search process. The primary goal of this summative evaluation was to determine the usability of the visualization concepts in dependency of different factors. A second goal was to uncover problems with the visualization ideas and components used in the INSYDER system, and to collect suggestions for improvements. The usability evaluation part of the study was focused on the added value of the visualizations (ScatterPlot, BarGraph, TileBar, StackedColumn) in terms of their effectiveness (accuracy and completeness with which users achieve task goals), efficiency (the task time users expended to achieve task goals), and subjective satisfaction (positive attitudes to the use of the visualization) for reviewing Web search results. Assuming the advantages of a multiple view approach described in the literature (See Chapter 3.4), we did not intend to measure the effects of using ScatterPlot, BarGraph, and SegmentView instead of the List and Table. We wanted to see the added value of using these visualizations in addition to the ResultTable. Another goal of this summative evaluation was to measure the influence of three of the five factors (*target user group, type and number of data, task to be done, technical possibilities, and training*) on the effectiveness, efficiency, and user satisfaction for each visualization. Table 30 gives an overview of the main hypothesis behind the evaluation, Table 31 of the experimental conditions, and Table 32 of the dependent variables.

Main Hypothesis
The ResultTable and the Visualizations produce results in terms of usability that differ from the results for the HTML-List.
The target user group influences how the usability will be determined by the user interface condition in comparison with the HTML-List.
The task type influences how the effectiveness will be determined by the user interface condition in comparison with the HTML-List.
The number of documents presented influences how the effectiveness will be determined by the user interface condition in comparison with the HTML-List.
The number of query keywords used and shown influences how the effectiveness will be determined by the user interface condition in comparison with the HTML-List.

Table 30: Main Hypothesis of the INSYDER visualization evaluation February – April 2000

5T-Environment	Variable	Characteristics	Type
-	User Interface	HTML-List only, ResultTable only, ScatterPlot + ResultTable, BarGraph + ResultTable, SegmentView + ResultTable.	Independent
Target User Group	IR Experience	Beginners, Experts	
Type and number of data	Number of keywords	1, 3, 8	
	Number of documents	30, 500	
Task	Complexity	Specific fact-finding, extended fact-finding	
Technical Environment	Hardware	Pentium III 400 MHz + 256 MB + 21-inch Monitor + Standard Keyboard + Standard Mouse	Static
	Software	MS Windows NT 4.0 SP 5 + MS Internet Explorer 5.0 + INSYDER Beta 3 2000-02-28 b	
	Settings	Screen resolution 1280x1024 pixels	
Training	Introduction	ScreenCam demonstration of INSYDER Beta 3 2000-02-28 b	
	Learning	15 Minute Warm-up	

Table 31: Experimental conditions

Variable	Measurement
Effectiveness	Completeness of answer in percentage
Task time	Seconds from opening the result set until success or time limit
Temporal efficiency	Effectiveness / Task time
Expected added value	Proportion of usage time between visualization and ResultTable in the Visualization plus ResultTable conditions
Satisfaction	Questionnaire

Table 32: Dependent Variables

4.3.1. Hypotheses

When preparing the evaluation, the results of a literature study, the discussions, and the results of the formative evaluations during the development of the INSYDER system led to several assumptions as to which factors would influence the success of a visualization. One of the assumptions was that the usability of a visualization is highly dependent on its usage context, which is generally described by the 5T-enviroment, but also by factors of the implementation itself. The multiple view concept seemed important. It was thought that a visualization like a ScatterPlot might be useful if used as a standalone visualization, but that the real added value would only come up when it was used together with other user interface components such as a list or a table. It might have been a good idea to compare a standalone ScatterPlot with an interactive combination of ScatterPlot and ResultTable, a standalone BarGraph and an interactive combination of Bargraph and ResultTable, and all other possible useful combinations of two or more components. Such an experimental design would have been possible if we had restricted the independent variables only to the used component. The validity of the results for a very special combination of factors would have been good, but their real-world importance would have been low given the very small section of real-world conditions testable. Accordingly, we restricted the tested combinations of components to a maximum of two at a time; allowed the users to use the visualization, the ResultTable or both; and varied some other factors listed in the tables above. Of the five factors influencing the success of using visual structures we decided to vary *target user group*, *type and number of data,* and *task to be done*. The remaining factors, *technical environment* and *training*, were identical for all tests, as we provided identical training sessions and technical equipment. To test the usability of the visualizations, the HTML-List was used as a baseline. Five main hypotheses were defined when preparing the evaluation. The first one concerned the visualization components themselves. The general expectation was that the visualizations influence the usability of the system in terms of effectiveness, efficiency, and user satisfaction. Formed in concrete hypotheses:

- H1a: The ResultTable and the Visualizations produce results in terms of user satisfaction that differ from the results for the HTML-List.
- H1b: The ResultTable and the Visualization plus ResultTable conditions produce results in terms of effectiveness that are different from the results for the HTML-List.
- H1c: The ResultTable and the Visualization plus ResultTable conditions produce results in terms of temporal efficiency that are different from the results for the HTML-List.

A second very important factor seemed to be the user himself. Given the possible differences among individual subjects we assumed that previous Information Retrieval experience might have an influence on the success of the visualization components. The expectation was that the usability of the different visualizations is dependent on the target user group (Beginner / Expert). The second hypothesis therefore was:

- H2a: The target user group influences how the user satisfaction will be determined by the user interface condition in comparison with the HTML-List.
- H2b: The target user group influences how the effectiveness will be determined by the user interface condition in comparison with the HTML-List.

Efficiency had not been included in the second hypothesis. Concerning the usability it would have been interesting to test effectiveness, efficiency, and user satisfaction for all hypotheses. As will be shown later, it is very difficult to handle the efficiency values of the experiment. They therefore had only been included for the statistical validation on the global level. Nevertheless, efficiency values will also be reported for the other cases but not included in the hypotheses and the statistical validations.

Not to exaggerate the questionnaires, the user satisfaction was measured only on a global level, and is therefore not included in the remaining three hypotheses.

Of the factors to be examined on a more detailed level, the first one was the type of task the users had to perform. The hypothesis derived from this assumption is:

- H3: The task type influences how the effectiveness will be determined by the user interface condition in comparison with the HTML-List.

Last but not least, the success of visualizations seemed to us to be dependent on the size of the result set on one hand and on the number of keywords used and visualized on the other. Therefore the last two hypotheses were:

- H4: The number of documents presented influences how the effectiveness will be determined by the user interface condition in comparison with the HTML-List.
- H5: The number of query keywords used and shown influences how the effectiveness will be determined by the user interface condition in comparison with the HTML-List.

The study of the results of the evaluation is restricted to these five hypotheses but will include also the examination of several other questions and interesting points.

4.3.2. Independent Variables

4.3.2.1. User Interface

From the various possible settings and combinations of components, the following user interface configurations were tested:

- HTML-List only
- ResultTable only
- ScatterPlot + ResultTable
- BarGraph + ResultTable
- SegmentView + ResultTable.

For a long time during system development and preparation of the evaluation, the HTML-List was not planned to be implemented or tested. It had therefore not been mentioned in earlier publications like [Mann 1999] or [Mann, Reiterer 1999]. The ResultTable as traditionally presented in form of a list had been considered as the standard against which the visualizations would be compared. During the final preparations of the evaluation, the idea came up that the usability of an interactive JAVA-Table with possibilities for configuration and sorting might be quite different from a really traditional HMTL Result List. The HTML version was therefore quickly implemented. It was included in the evaluation as a baseline for the usability values.

A special case in the user interface dimension is the SegmentView. As described in Chapter 4.2.3, we had implemented five variants of TileBars and StackedColumns with the idea of comparing the different implementations. Initially, the plan had been to perform first an evaluation of the different SegmentView versions, and then to select the most successful one for the comparison with the other components. Due to time and resource restrictions this intermediate step was skipped and the users had the possibility of using the version(s) they wanted.

It is important to keep in mind for the later discussion of the usability results for the various components, that what was tested is the INSYDER implementation of a ScatterPlot or the INSYDER implementation of a TileBar. Studies show that even the wording used in a user interface may influence the success of a user interface [Shneiderman, Byrd, Croft 1997]. Accordingly, the results for the INSYDER implementations of certain components may or may not be comparable with the evaluation of other implementations.

To avoid side effects caused by additional functions of the INSYDER system, it was modified in such a way that all functions were suppressed that were not needed to perform the task or allow refinement steps other than view transformations. All functions used to create new searches or to access watch or bookmark / news functionality were removed or deactivated. When using the visualizations, the subjects had functions like zoom or mark / unmark documents, but they did not see functions that the INSYDER system normally offers such as generating new queries; using relevance feedback; or re-ranking existing result sets by changing, adding, or deleting keywords. In addition, special configuration files ensured that only the Sphere of Interest and the components for their current task were available to users. (See Figure 133 for an example.)

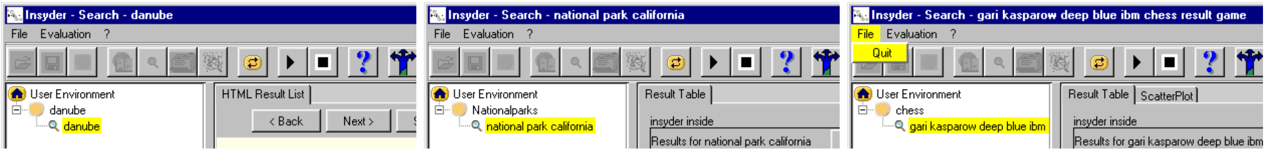


Figure 133: Example user interface conditions for the first three tasks of group 1.

4.3.2.2. Target User Group

As described above, the target user group for the INSYDER system was business analysts from small- and medium-sized enterprises. The decision to choose visual structures common to standard business graphics was motivated by this fact. During the INSYDER projects, members from the target user group were involved in the definition of requirements and formative evaluations of mock-ups and prototypes. There are several known problems performing studies with students as subjects. Nonetheless the summative evaluation of the visualizations was done with students from different disciplines and university staff. This decision was possible because searching the Web is an activity not restricted to the special target user group of the INSYDER system. For many people, especially students, it is nowadays an everyday task. Moreover, because the evaluation had been focused on the visualization components, most of the special functions of the INSYDER system created for the usage in the context of business intelligence played only a marginal role. Last but not least, business graphics are quite common in everyday life, and the visualizations implemented in the INSYDER system are simple compared to many other ideas found in the literature.

That the initial target user group of the INSYDER system was not tested, does not imply that we believe that the success of the visualization components is independent from the individuals or groups using the system. Our idea was to perform the study differentiating between "beginners" and "experts" as had other studies before us. Using an approach comparable to that of [Golovchinsky 1997a] we had an expert group, characterized by having at least received the formal training of a Faculty of Information Science Information Retrieval course, and a beginners group without this formal training. According to the headings from our 5T-Environment, this differentiation could also have been theoretically classified as changing the Training-dimension. "Training" in the 5T-Environment focuses, however, on the actually used system and not on the general characteristics of the user group. To eliminate other variables as much as possible only subjects were used for the study who had at least some practical computer usage experience on the one hand, and some basic knowledge about the World Wide Web as well as browsing and searching it on the other hand. (For a discussion about the influence of expertise on search success, see Chapter 2.3.2.)

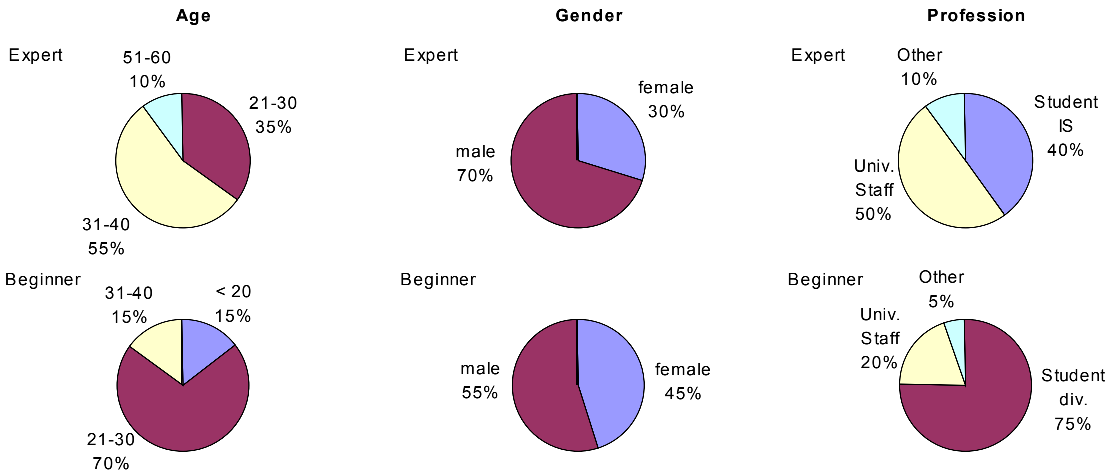
Another important factor possibly biasing the results is the language used for the study. As already mentioned, the ranking engine of the INSYDER system works best for English and French. The tests were performed in Germany with German users. A German version of the thesaurus was not available. We therefore decided to use English keywords to rank the documents. In Chapter 3.5, a study by [Morse 1999] was mentioned. In this study conducted in the United States of America and Norway, non-native English speakers performed more slowly in each of the visualization conditions except for the table display[135]. Similar results appeared in an earlier study by [Morse,

[135] The "table display" used by [Morse 1999] is completely different from the ResultTable of the INSYDER system. Whereas Morse's table contains text only in the cell headings and "+" / "-" signs or digits in the cells, the INSYDER table is a text table.

Lewis, Korfhage et al. 1998]. All subjects in the INSYDER study were non-native English speakers, which might have biased the results. In order to minimize those influences only subjects with a sufficient level of English language skills were chosen. The questions were formulated in German to eliminate problems in understanding the task. In addition the tasks were restricted to specific and extended fact-finding tasks, performable with only basic knowledge of English.

The four users for a pre-test and the 40 additional volunteer subjects, 20 beginners and 20 experts, for the main study were all recruited at the University of Konstanz, Germany. A movie voucher was offered as motivation for participating in the main study. Figure 134 and Figure 135 show the characteristics of the user population of the main study.

The experts were in most cases either students of Information Science or staff from the Information Science Group including research assistants and one professor. Most users classified as beginners were students from other University departments including mathematics, physics, law, and psychology.



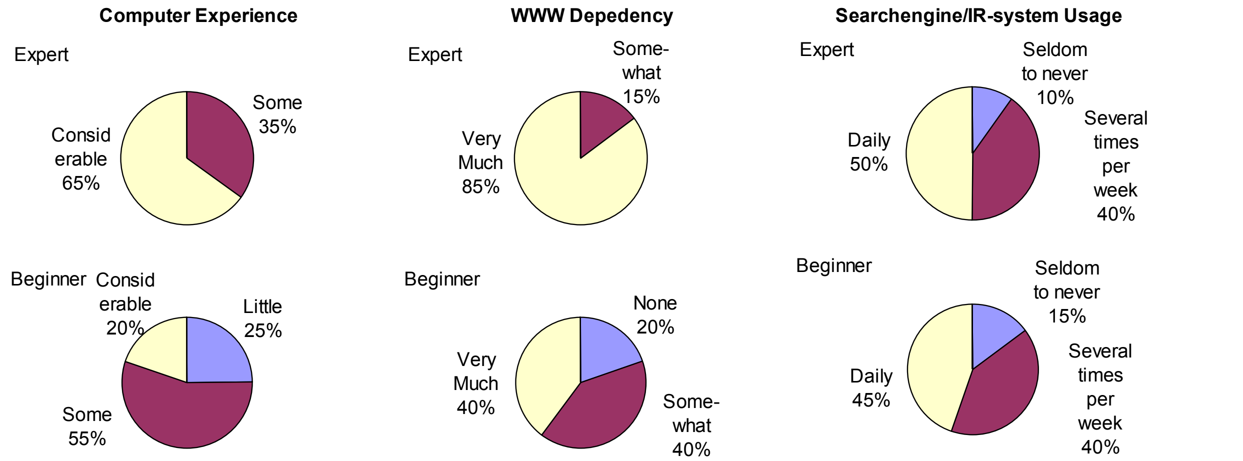
	Expert	Beginner
Age 51-60	2	
Age 31-40	11	3
Age 21-30	7	14
Age < 20		3

	Expert	Beginner
Gender male	14	11
Gender female	6	9

	Expert	Beginner
Other	2	1
Univ. Staff	10	4
Student div.		15
Student IS	8	

Figure 134: User characteristics: Age, Gender, and Profession

Concerning computer and software experience the users had to classify themselves as beginners (little experience), advanced users (some), or experts (considerable). In a second question the users were asked how much they depend in their work on information from the Web: very much, somewhat, or none. Finally, they were asked how often they use search engines or other Information Retrieval systems: seldom to never, several times per week, or daily. Whereas computer experience and Web-dependency showed the expected differences between users classified as experts and beginners, the values for the usage of search engines or IR-systems showed surprisingly little variation.



	Expert	Beginner
Considerable	13	4
Some	7	11
Little		5

	Expert	Beginner
Very Much	17	8
Somewhat	3	8
None		4

	Expert	Beginner
Daily	10	9
Several times per week	8	8
Seldom to never	2	3

Figure 135: User characteristics: Computer Experience, WWW Dependency, Search engine/IR-system Usage

4.3.2.3. Type and number of data

Visualization components for Web search results should be tested using Web search results. For our study we therefore used real data collected from the World Wide Web. In Chapter 2.3 several findings were presented about how people search the Web. Besides other factors the number of keywords used to formulate queries and the number of documents examined in the result set were discussed. To summarize it was found that the average length of a query is around two keywords, with an increasing tendency, and that only a small number of hits is examined by the users. For the purposes of the summative evaluation of the INSYDER visualizations we planned to perform the test with varying numbers of keywords and varying sizes of result sets. The initial plan for the number of keywords was to use one, two, or three keywords. This corresponds to common values when searching the Web. Discussing the visualization of search results and the number of concepts displayed [Cugini, Laskowski, Sebrechts 2000] report their experience that the resulting display became complex and difficult to interpret, when the number of concepts reaches seven or eight. John V. Cugini had reported the same information in personal communication with the author in 1999. In view of this boundary we changed the plan and used queries with one, three, or eight keywords. For the number of results displayed we wanted to compare the effects of small and large result sets. We ultimately settled on two different sizes of result sets: 30 and 500 hits. A 30-document border is discussed in several papers such as [Koenemann, Belkin 1996], [Eibl 1999], [Cugini, Laskowski, Sebrechts 2000]. The 500-hit border emerged when preparing the result sets for the evaluation. The INSYDER system and its visualization components had been tested during development with result sets of up to 2000 hits. The time needed to load into the visualization component a locally stored result set with 30 hits was about one second on the machines used, for a 500-hit result set about three seconds, and for a 1000-hit result set about six to seven seconds. This loading time occurred for every switch from the ResultTable to a Visualization. The other way around it was always less one second. Tests by the development team revealed that the three-second waiting time seemed tolerable, but that six seconds was considered definitely too long. 500 hits was therefore chosen as the value for large result sets. Figure 136 to Figure 139 give some impression as to how the visualizations looked like with one, three, or eight keywords and with 30 or 500 hits. To improve recognition of details the surrounding parts of the INSYDER user interface are clipped in the reproductions for this thesis.

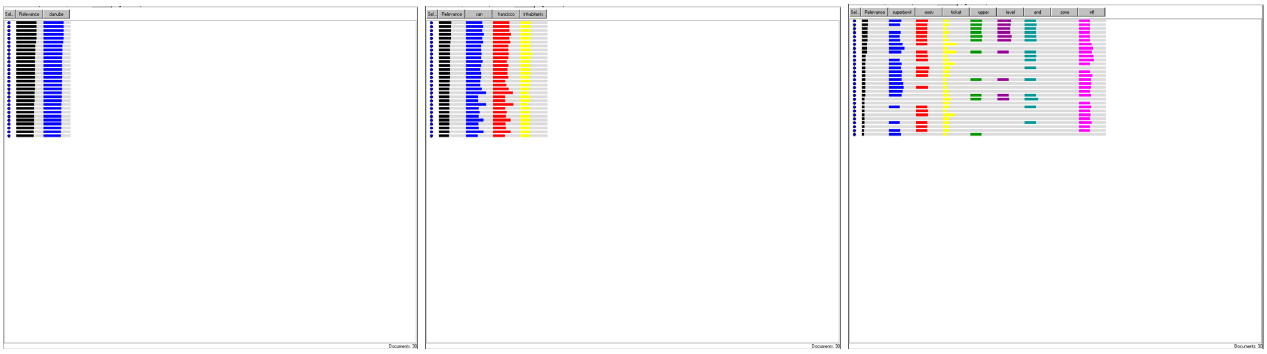


Figure 136: Bargraph with 30 hits: one, three, or eight keywords

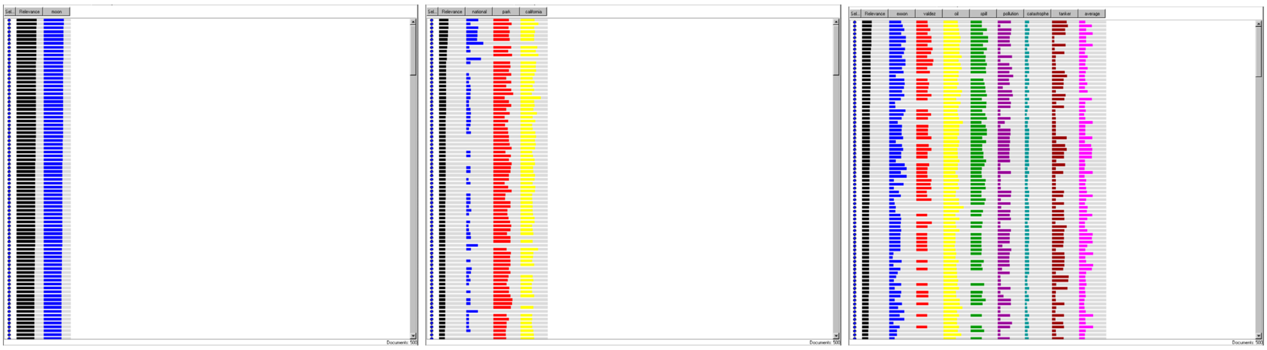


Figure 137: Bargraph with 500 hits: one, three, or eight keywords

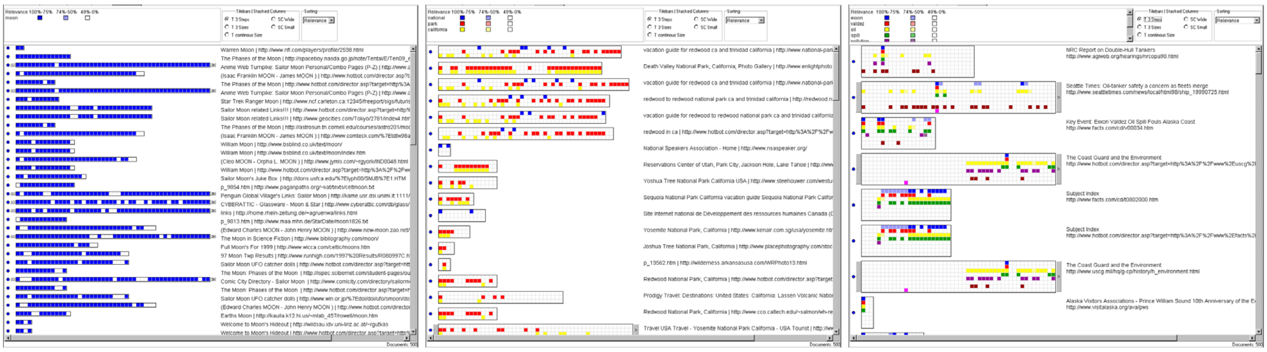


Figure 138: SegmentView (TileBars 3 Steps): one, three, or eight keywords

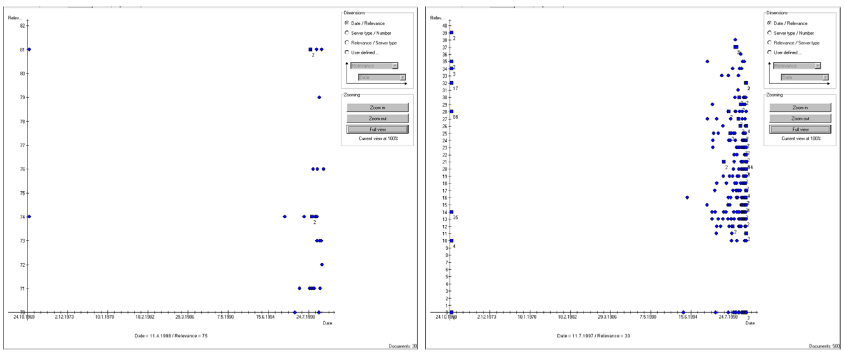


Figure 139: ScatterPlot: 30 or 500 hits

Another important aspect of the data sets used for the evaluation is their quite heterogeneous content. The datasets that were prepared for the evaluation by searching the Web with different keywords for 12 topics showed a great variation, especially in top-30-precision. The first 30 documents had a very low precision in particular for queries with three or eight keywords. This low precision was a product of a speciality of the crawling algorithm of the INSYDER system. For multiple keywords queries with n keywords and s search engines as starting points, INSYDER sends (n+1)*s queries to get the first seed documents to start the analysis. The important factor is the "n+1". As mentioned above, keywords are automatically OR-ed from INSYDER. To broaden the range of seed files every search engine used as a starting point is not only queried with all keywords in one query, but also with every single keyword in additional queries. Thus a query such as "visualization search results internet" leads to the n+1 = 5 queries: "visualization OR search OR results OR internet", "visualization", "search", "results", "internet". Theoretically this is redundant, but this holds only true when crawling time is not considered. The top-ranked links will be dependent on the ranking algorithm of the search engine(s) used as a starting point. In order to bypass this dependency and use the full power of the INSYDER analysis engine, the n+1-approach is used. The analysis engine and the knowledge base with all its synonyms and semantic connections are also used to detect promising links to follow in seed documents. This crawling mechanism is a science on its own and could not be changed in the project. The approach is quite powerful, but it has the side effect that the first documents crawled and analyzed contain too often only one of the keywords. The second "generation" of documents crawled from this seed files are in general much better ranked for the overall query. To counterbalance this effect for all the document sets prepared INSYDER was run until more than 500 documents per query had been crawled and ranked. The resulting document sets had then been clipped at 30 or 500 documents.

The local storage of the documents together with the fact that the machines had been disconnected from the Internet during the evaluation meant that the document sets presented to the users consisted of pure HTML-documents without pictures.

4.3.2.4. Task

In order to observe possible influences caused by the task to be done, we decided to use two of the four different types of information-seeking tasks described in [Shneiderman 1998] and listed in Table 2 on page 20. Half of the tasks that the users had to fulfill were of the type "specific fact-finding (known-item search)"; the other half were of the type "extended fact-finding". For several reasons including potential problems with the English language and the question of how to measure effectiveness we did not include tasks of the types "open-ended browsing" or "exploration of availability". The general concept behind the evaluation was to concentrate in the information-seeking process on the phase or step named variously review of results, evaluate results, or examine results. (See Figure 140 for the position of this step in the whole information seeking process.)

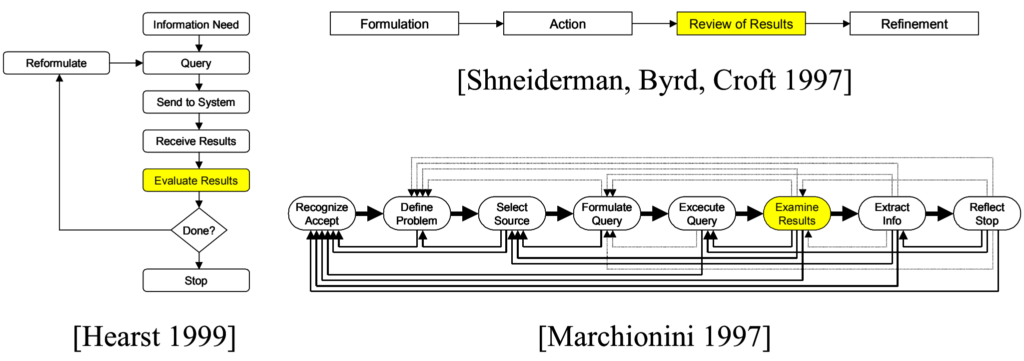


Figure 140: Selected tasks and their position in the information seeking process

The situation so evaluated is somewhat artificial. We created an information need for the user by asking a question. The user than had to skip several steps, because we already performed them for all users so as to eliminate influences from these phases. Even in the review of results we restrained the user by not allowing steps like reformulation of the query or selection of other sources. In addition, we forbad browsing. The goals of the user may differ from real-world information needs; the Information-Seeking Strategy (ISS) is biased; and we cut off important mechanisms for information seeking in the Web.

What are the participants' goals when working with the information seeking system in this experiment? In most cases, a genuine information need does not stand behind the goals. Maybe the question asked waked the interest of the participant, and the assumed information need really contributed to the goals he pursued. In many other cases, the goal may merely have been to answer the question as quickly as possible so as to come back to the cafeteria as soon as possible, to do a favor for the questioner, or to get the promised movie voucher with minimum effort.

In light of the Information-Seeking Strategy defined by [Belkin, Marchetti, Cool 1993] / [Belkin, Cool, Stein et al. 1995] (See Table 3 on page 22), we assumed a situation that may be characterized as ISS15 (Method: Search, Mode: Specify), but we tested a task that may be typified as ISS5 (Method: Scan, Mode: Recognize). The common elements for both strategies are Goal: Select, and Resource: Information.

In Chapter 2, the importance of the iterative nature of the information-seeking process in general, and the following of links in result sets of Web searches in particular, was explained. For example, [Hölscher, Strube 2000] reported when documenting information-seeking strategies of twelve internet experts, that in 47% of the cases in which the experts used a search engine browsing episodes of varying length occurred. Nevertheless, we decided not to allow the following of links from documents of the result set to other documents. Our test setting allowed the machines used to be disconnected from the Internet because all documents in the result sets had been locally stored. Consistency in the systems answer times could thus be guaranteed for all users and all conditions. If browsing to the Internet had been allowed, this controlled environment condition would have been defeated.

The preparation of the tasks and the corresponding result sets turned out to be really hard work. Points discussed included whether tasks should be included that tend to favor certain visualizations, for example, a question like "What was the gross national product of Germany in 1999". Using the ScatterPlot with its default dimensions date / relevance seemed to have in this case been an advantage. All documents with last modified dates before 1999 and not having 1970 could be excluded from examination. For the questions finally chosen all documents of the result set had to be examined manually to create lists of correct answers and to eliminate all documents from the result set that would allow the extended fact-finding tasks to be completed by referring to a single document. With the latter we tried to ensure that the extended fact-finding tasks really were different in nature from the specific fact-finding tasks. The main difference between these two types is that in the latter case, there is a clear stop criterion, when the user finds a document that answers the question. In the former case, there is no such clear criterion to stop the examination of a result set, and therefore the investigation process of a result set will be much broader in scope and possibly of longer duration. For example, if the task was to find all books by John Irving, we eliminated documents listing all books by John Irving. Otherwise by finding this document very early in the process there would have been no difference in scanning effort compared to a specific fact-finding task. This step did not influence the size of the result sets. When we eliminated a document from the set prepared to be presented to the users, it was substituted by the first document not included so far, i.e. number 31 or 501. Two tasks using document sets with 500 hits but none of the tasks with 30 hits were manipulated in this way.

The example of [Dempsey, Vreeland, Sumner et al. 2000] shows also how difficult it is to find neutral questions, when designing a study. They found, that in two from four of their questions, which contained per chance proper names, an unguided Web search outperformed their carefully designed subject gateway. The authors' explanation was that in the two tasks the proper names seemed to be an excellent discriminator for a Web search (Ethan Katsh and Benjamin Edward Peterson) in contrast to the other ones (President Clinton and Kunbok Lee / Gisun Lee).

Table 33 and Table 34 show the finally tested tasks, the corresponding keywords, and the size of the result sets. The original German formulations and the answers can be found in the appendix.

#	Task	Keywords	Hits
1	How long is the Danube river?	danube	30
3	Who lost the second game in the chess match between Gary Kasparov and Deep Blue in 1997?	gari kasparow[136] deep blue ibm chess result game	30
5	How many inhabitants has San Francisco at present?	san francisco inhabitants	30
7	What is the mass of the moon?	moon	500
9	How much oil (t, l, barrel) did the tanker "Exxon Valdez" spill during its accident?	exxon valdez oil pollution catastrophe tanker average spill	500
11	On which day did the Titanic sink?	titanic sinking iceberg	500

Table 33: Specific fact-finding tasks

#	Task	Keywords	Hits
2	List names of national parks (NP) in California!	national park california	500
4	List cities, whose museums exhibit works of art from the Venetian painter Titian!	tizian[137]	500
6	List mountains that are higher than 8000 meters (26248 feet)!	mountain himalaya altitude height top peak reinhold messner[138]	500
8	List books by John Irving!	john irving book	30
10	Which things (companies, projects etc.) bear today the name of the philosopher Plato?	platon[139]	30
12	What are the prices for tickets in the category "upper level end-zone" offered by the different online-shops for the 34th Superbowl?	superbowl nfl national football league ticket xxxiv atlanta	30

Table 34: Extended fact-finding tasks

4.3.3. Static Variables

4.3.3.1. Technical Environment

As explained above the technical environment was identical for all 40 user sessions. All tests were carried out on two identical 400 MHz Pentium III PCs, running MS Windows NT 4.0 SP5 with

[136] Unfortunately, the German version of the name "Gari Kasparow" was used, rather than the English "Gary Kasparov". In addition in German the name is most of times not spelled "Gari" but "Garri". The quality of the ranking may have been thereby adversely affected, because not all documents might contain the name and / or the proportion of German documents retrieved might have been higher.

[137] Unfortunately, the German version of the name "tizian" was used, rather than the English "titian". See above.

[138] Reinhold Messner is one of the at least six human beings who has climbed all mountains higher than 8000m.

[139] Unfortunately, the German version of the name "platon" was used, rather than the English "plato". See above.

256 MB RAM, identical software-configuration, hard disk and 21-inch monitors with a screen resolution setting of 1280x1024 pixels. We used standard keyboards and mice. The INSYDER software version was INSYDER Beta 3 2000-02-28 b, the external Browser MS Internet Explorer 5.0. Each PC was located in a separate office and disconnected from the local area network and the Internet to minimize disruptions from events independent from the evaluation itself.

By using 256 MB RAM, 21-inch monitor, and 1280x1024 pixels we definitely went beyond the typical environment of the INSYDER target user group, who tend to use standard PCs and 17" or only 15" monitors. The 256 MB RAM was necessary to ensure the desired short answer times of the system in the beta version that we used for the tests. Internal system performance was at that time still subject to improvement to reach the foreseen corridor defined by the market requirements. The usage of a 21-inch monitor and 1280x1024 pixels was also a concession to the beta status of the software. Elements of the user interface like the Sphere of Interest panel or the status line of the crawling engine could not be closed or configured to disappear. They always took up a certain amount of screen space. Despite the fact that the beta version was also running on 800x600 pixels or less, the space that could be used for the visualization components running with 1280x1024 pixels may be equivalent to the space available in an optimized planned product version at 1024x768 pixels. It would have been interesting to compare effectiveness and efficiency of the users with different screen resolutions. Unfortunately, this would have only been possible if we dropped therefore one of the other dimensions manipulated.

4.3.3.2. Training

Training was also added in our ventilations of which factors to vary. Several authors emphasize, that the subjects' experience with a certain type of system or visualization can be an important factor influencing test results. Like most of our predecessors, however, we did not really have the chance to investigate the effects of training because it was expendable enough to run the test in the described setting. To really see what the effects would be when users are experienced with ScatterPlot, BarGraph, or SegmentView, a design would have been necessary which a significant number of people would be forced to use our system for a certain number of weeks and then to repeat the evaluation. As a substitute we could have given training sessions of significantly different lengths for two groups. The overall test requires more than two hours per person, however, even with the single, up-to-15 minute training session we finally had. An extension of training to investigate its effects was ultimately rejected.

4.3.4. Dependent Variables

To measure the effectiveness and the efficiency of the visualizations we used high-level metrics as defined by [Cugini 2000]. Such high-level metrics are characterized as follows:

- They measure some broad property of a user session.
- They are result-oriented (what got done, how fast?). They summarize overall performance but do not explain why.
- They treat the implementation as a black box, hence they are less dependent on specifics of the prototype than low-level metrics (e.g. path length of search).
- They are easier to interpret than low-level metrics.

Findings from other authors evaluating visualizations such as [Sen, Boe 1991] strengthened our motivation to measure both soft and hard facts. "*Sophisticated Software interfaces, like graphical information displays, could lead to increased confidence in decision making without significant improvement in the quality of decisions made*" [Sen, Boe 1991]. Accordingly, in addition to the performance facts of effectiveness, task time, and temporal efficiency, we tried to measure the soft facts expected added value and satisfaction. With the exception of the expected added value the ideas behind the measurements are drawn from the studies listed in Chapter 3.5 as well as from [ISO 9241-11] and the IUSR [Industry USability Reporting project 1999]. Details will be explained below. Precision and recall as the classical dimensions for the evaluation of IR-systems were not used because in this study we did not evaluate the IR system just the visualization components.

4.3.4.1. Effectiveness

In general, effectiveness can be measured by the accuracy and the completeness with which users achieve the goals of the test tasks [ISO 9241-11]. Other possible measures include the number of assists [Industry USability Reporting project 1999]. In the case of our evaluation, the effectiveness was measured as a completion rate scored on a scale of 0 to 100%. The completion rate for the specific fact-finding tasks was either 0% or 100%. The completion rate for the extended fact-finding tasks was calculated by comparing the number of correct answers given by the user to the number of correct answers that could be found in the actual result set. For example, if 14 mountains higher than 8000 m or 26248 feet could be found in the result set and the user found eight, his effectiveness was recorded as 57%. Users were not punished for additional wrong answers. If the user listed 10 mountains including two below that limit or not to be found in the result set, effectiveness was still recorded as 57%.

4.3.4.2. Task time

The task time was measured in seconds from the moment the result set was opened until either the question was successfully answered; the user terminated the task; or the time limit to complete each test task was reached. All times were measured by the experimenters with stopwatches. The time taken to read and understand the task itself was not measured. Users were asked not to open the result set before they understood the question. Drawing on the results of a pre-test, the time to answer specific fact-finding questions was limited to 5 minutes per question, for extended fact-finding tasks to 10 minutes per question. The goal behind these time limits was to avoid overall test times per user longer than approximately two hours.

4.3.4.3. Temporal efficiency

The temporal efficiency was calculated as effectiveness divided by task time. Other efficiency measures like mental or physical effort, materials or financial cost related to effectiveness were not used.

4.3.4.4. Expected added value

For all conditions where the users were free to utilize the ResultTable and / or one of the three visualization components ScatterPlot, BarGraph, or Segment View, we sought to measure how these multiple view choices would be used. We therefore measured the usage times separately for each component and calculated the proportion of usage time between the Visualization and the ResultTable in the Visualization plus ResultTable conditions. In addition, we calculated which tasks had been solved by using the ResultTable alone, the Visualization plus the ResultTable, or the Visualization alone. Our thinking was that the users would utilize the visualization to support their task based on their expectations about the added value. The choices may have been biased by the test subject's expectations of what would be tested in the study. Please note that the usage times recorded for a condition include detail view times using tooltips or the browser to view documents.

4.3.4.5. Satisfaction

Last but not least, we measured satisfaction with a questionnaire. Test users were asked to rate their satisfaction in terms of: ease of use, self-descriptiveness, suitability for learning, layout, suitability for the tasks, and conformity with expectations.

4.3.5. Procedure

The overall test procedure for every subject included the five main steps shown in Figure 141. Each user had his own session. Each session, which lasted approximately two hours, followed the same pattern. After filling in an entry questionnaire with six questions to collect demographic data, the users were given a standardised introduction to the INSYDER system with the help of a ScreenCam™ movie, which demonstrated and explained the main concepts and visualizations of the system. Each user then had a warm-up learning period with a test result set and all five visualizations. After completing this introductory phase, the users had to accomplish the twelve test tasks. All users had to perform the same twelve tasks in the same sequence. Users had been randomly assigned to one of five groups. Each group used all user interface conditions, but each started with another condition. During the tasks, the users were requested to "think aloud" so as to allow the evaluation team to understand and record their current actions. Two persons made a written record of data. An experimenter moderated the test session. After accomplishing the tasks, the users answered a closing questionnaire of 30 questions regarding their subjective satisfaction and their proposals for the improvement of the system.

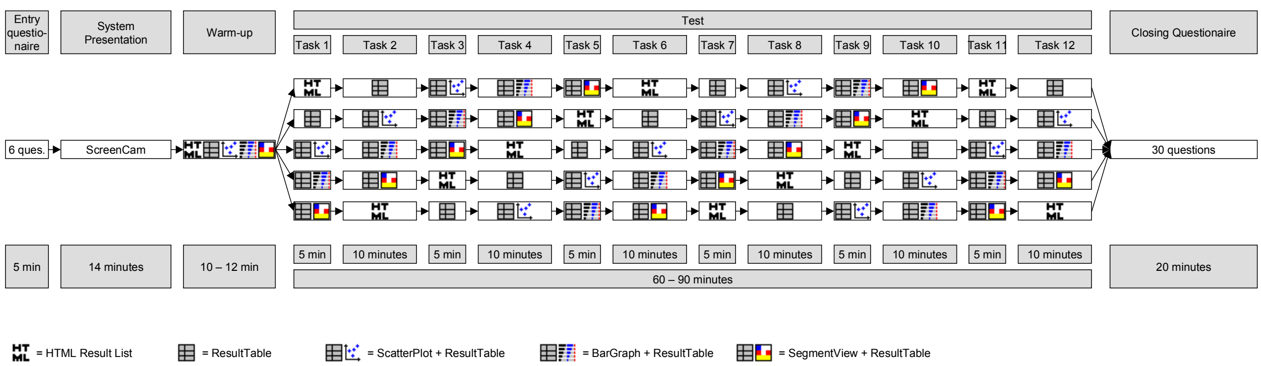


Figure 141: Overview of the final test procedure for the members of the five different groups

4.3.5.1. Pre-test

A pre-test with four users was conducted prior to the main evaluation of the system with 40 subjects. The four test users were did not participate in the main evaluation. Three of them were experts. One was a Beginner. The first goal of the pre-test was to ensure that the questionnaires, the

ScreenCam introduction, and the task descriptions were clear, concise, and comprehensible. A second goal was to test the intended ScreenCam recording of the user sessions. A third goal was to find out how long a test session would last for a user. In contrast to the subsequent main evaluation the pre-test included no time limits for the tasks. The pre-test led to some minor changes in the questionnaires and the task descriptions. The ScreenCam recording of the user sessions crashed several times during the pre-test and was therefore rejected for the main evaluation. A large problem was the overall duration of the user sessions which ranged from 2 hours 25 minutes to 3 hours 13 minutes. The time needed for ten or eleven of the twelve tasks was between one and two hours and is shown in Table 35. Some of the tasks were skipped because the INSYDER prototype crashed when displaying the SegmentView (indicated as "/" in the table). The crash problem was largely solved when performing the main evaluation. Nevertheless, a few system crashes occurred during the main evaluation. In this event, time recording was stopped and when the former system status was reached again, restarted.

User	Specific fact-finding		Extended fact-finding		Total task time
	30 hits	500 hits	30 hits	500 hits	
Beginner	9:00	7:00	16:00	20:15	**1:52:43**
	5:03	5:00	12:05	11:11	
	/	3:27	14:15	9:27	
Expert 1	4:55	3:00	6:35	10:20	**0:54:59**
	1:43	1:07	/	11:37	
	/	0:26	4:16	11:00	
Expert 2	6:22	1:30	11:47	15:32	**1:44:36**
	2:40	4:35	8:55	26:28	
	/	4:31	7:00	15:16	
Expert 3	4:22	16:55	8:30	22:47	**2:02:22**
	1:02	4:52	/	28:42	
	0:22	4:40	7:09	23:01	
Average	**3:57**	**4:45**	**9:39**	**17:08**	**1:38:40**
Total Average		**4:24**		**13:44**	

Table 35: Task times of the pre-test

The pre-test users spent up to half an hour on a single task out of the twelve. In view of the task times and results from the pre-test it was decided to restrict the time for specific fact-finding tasks to 5 minutes and for extended fact-finding tasks to 10 minutes in the main evaluation. Accordingly, the maximum time to solve the twelve tasks was 1 hour 30 minutes. Because we hoped a user would not reach the limit for each question, we expected overall task times of about one hour.

4.3.5.2. Entry Questionnaire

The one-page entry questionnaire contained questions about gender / age, profession, computer and software experience, dependency on the World Wide Web at work, and the frequency of usage of search engines or other Information Retrieval systems. The results were listed in Chapter 4.3.2.2. An additional question asked after the current mood of the user.

4.3.5.3. ScreenCam introduction

The intention behind the use of a 14-minute ScreenCam movie, demonstrating and explaining the main concepts and visualizations of the system, was to ensure that every user received the same introduction to the system. Each user performed the test in a separate session. We therefore needed to give the same presentation of the system 40 times to avoid different levels of explanation or different information about the system biasing the results.

The ScreenCam movie was produced with the Lotus ScreenCam recording software for Windows NT. The demonstration of the INSYDER system was recorded together with spoken explanations. The ScreenCam movie started with a very brief introduction into the INSYDER project, which concentrated on the search aspects and did not mention things like watch, bookmarks, or news. This was followed by a neutral explanation about the goal of the evaluation, with the message that different forms of search result presentations would be compared and that the searches themselves had already been performed. In the main part, the possibilities of the ResultTable, the ScatterPlot, the BarGraph, and the SegmentView were demonstrated and explained. To do so, a three keyword query (jack nicholson birthday), a 90 document result set, and the question “Which is the birth date of Jack Nicholson?” was used. For the SegmentView the users were encouraged to try out all versions in the subsequent warm-up phase and use the one(s) they preferred in the main test.

4.3.5.4. Warm-up Phase

After the end of the ScreenCam presentation, the users could use the INSYDER system with the Jack Nicholson result set shown in the presentation. For this training phase all five components were available. The goal of the warm-up phase was to enable the user to become familiar with the system. In addition, problems that the subjects had using the system could be detected and corrected. The users were free to end the warm-up when they felt familiar enough with the system. The phase lasted about ten to twelve minutes but no longer than 15 minutes.

4.3.5.5. 12 Tasks

As shown in Chapter 4.3.2, we sought to test the following independent variables with their corresponding values: five different user interface conditions, two different target user groups, two types of tasks, two sizes of result sets, and three different numbers of keywords (i.e. 5 x 2 x 2 x 2 x 3 = 120 experimental cells). Each cell would be tested with at least four subjects, providing 480 values. As we decided that both beginners and experts would perform the same tasks in the same conditions, 60 combinations remained. In a within subject design with four experts and four beginners, every user would have had to perform 60 tasks. This was far too much. We therefore decided to mix a within subject and a between subject design by spreading the five visualization conditions over the remaining 12 cells. This led to the final design shown in Figure 141. At the end, we had 480 values, but with 40 users each performing 12 tasks rather than of 8 users each performing 60 tasks.

The final test setting covered all combinations of the above described different user interface conditions, target user groups, types of tasks, sizes of result sets, and numbers of keywords. Each cell of the test table was tested with 8 users (4 beginners, 4 experts). All users processed the same 12 questions with the same keywords and number of hits in the same order. The difference between the five groups was the visualization that the user could use to answer the question. The system ensured that for each task that a user had to fulfill he could only see the result set and visualizations provided for this step. The setting for this controlled experiment assured that the five combinations of visualizations were distributed equally among all variables. Table 36 and Table 37 show the test setting, each from a different angle. Table 36 offers an example of the way of an expert from group one.

Type of task	User Group	Hits	Keywords	User Interface condition				
				HTML				
Specific fact-finding	Beginner (B)	30	1	B1	B2	B3	B4	B5
			3	B2	B3	B4	B5	B1
			8	B4	B5	B1	B2	B3
		500	1	B5	B1	B2	B3	B4
			3	B1	B2	B3	B4	B5
			8	B3	B4	B5	B1	B2
	Expert (E)	30	1	**E1**	E2	E3	E4	E5
			3	E2	E3	E4	E5	**E1**
			8	E4	E5	**E1**	E2	E3
		500	1	E5	**E1**	E2	E3	E4
			3	**E1**	E2	E3	E4	E5
			8	E3	E4	E5	**E1**	E2
Extended fact-finding	Beginner (B)	30	1	B2	B3	B4	B5	B1
			3	B4	B5	B1	B2	B3
			8	B5	B1	B2	B3	B4
		500	1	B3	B4	B5	B1	B2
			3	B5	B1	B2	B3	B4
			8	B1	B2	B3	B4	B5
	Expert (E)	30	1	E2	E3	E4	E5	**E1**
			3	E4	E5	**E1**	E2	E3
			8	E5	**E1**	E2	E3	E4
		500	1	E3	E4	E5	**E1**	E2
			3	E5	**E1**	E2	E3	E4
			8	**E1**	E2	E3	E4	E5

Table 36: Combination of test tasks by variables

In general, we tried to ensure that the variables changed between each question, starting with the user interface condition (HTML - - - - - HTML - - - - - …), followed by the type of task (specific – extended – specific – extended - …), followed by the number of keywords (1 – 3 – 8 – 1 – 3 - 8 - …), and last but least by the number of hits (30 – 500 – 30 – 500 - …). For the last variable the alternation was not possible between questions 6 and 7.

Question	Fact-finding	Keywords	Hits	Group 1 (B / E)	Group 2 (B / E)	Group 3 (B / E)	Group 4 (B / E)	Group 5 (B / E)
1	Specific	1	30	HTML	Table	Table + Scatterplot	Table + Bar graph	Table + Segment view
2	Extended	3	500	Table	Table + Scatterplot	Table + Bar graph	Table + Segment view	HTML
3	Specific	8	30	Table + Scatterplot	Table + Bar graph	Table + Segment view	HTML	Table
4	Extended	1	500	Table + Bar graph	Table + Segment view	HTML	Table	Table + Scatterplot
5	Specific	3	30	Table + Segment view	HTML	Table	Table + Scatterplot	Table + Bar graph
6	Extended	8	500	HTML	Table	Table + Scatterplot	Table + Bar graph	Table + Segment view
7	Specific	1	500	Table	Table + Scatterplot	Table + Bar graph	Table + Segment view	HTML
8	Extended	3	30	Table + Scatterplot	Table + Bar graph	Table + Segment view	HTML	Table
9	Specific	8	500	Table + Bar graph	Table + Segment view	HTML	Table	Table + Scatterplot
10	Extended	1	30	Table + Segment view	HTML	Table	Table + Scatterplot	Table + Bar graph
11	Specific	3	500	HTML	Table	Table + Scatterplot	Table + Bar graph	Table + Segment view
12	Extended	8	30	Table	Table + Scatterplot	Table + Bar graph	Table + Segment view	HTML

Table 37: Combination of test tasks by question

To perform the main evaluation the users were given a three-page question and answer paper. It contained some short written instructions, the questions, a simple test condition code, the key-words used, and after every question, some empty lines to write down the answers. One line was provided for each specific fact-finding task and three lines for every extended fact-finding task. The users were told to answer the questions as quickly as possible.

During the tasks, the users were requested to “think aloud” so as to enable the evaluation team to understand and record their current actions. The written recording and the taking of times using stop watches was carried out by two persons. One of them moderated the test session as experi-menter so that in the event of problems this person could help. The protocol / experimenter-team consisted altogether of five persons, who worked in different constellations as two-party teams.

The ResultTable was preconfigured in all tasks where it was available to show in order: select-flag, relevance for query, title, Relevance Curve, server type, URL, date last modified, size in words, abstract and the relevancies per concept. Document language, document type, size in kB, and relevance feedback flag were omitted for different reasons including a simplification of the display. A script ensured that all select-flags in the documents had been reset before a user started his tasks.

4.3.5.6. Questionnaire

After accomplishing the test tasks, the user had to answer a questionnaire of 30 questions regard-ing their subjective satisfaction. The questions concerned the eight different areas of usability listed in Table 38. Four different types of questions were used:

- Attitude alternatives: agree, undecided, disagree
- Attitude five point rating- / Likert-scale: anti-statement –2 –1 0 +1 +2 statement
- Selection: different possibilities with additional field for comments
- Open questions with the possibility of writing down comments

Category	Number of Questions	Alternatives	Likert	Selection	Open
Introduction	1	-	-	1	-
Suitability for the task	4	2	2	-	-
Ease of use	4	-	1	1	2
Self descriptiveness	4	1	-	2	1
Suitability for learning	4	2	1	-	1
Confidence	3	1	1	1	-
Design and Layout	3	-	1	2	-
Conformity with user expectations	3	-	-	-	3
Mood	4	1	1	2	1
Total	30	7	7	8	8

Table 38: Questions grouped by category and question type

In the questionnaire, the questions were grouped according to their type so as to ease the answering procedure for the users. In addition the questions were mixed in a way such that two questions of the same category did not follow each other. The idea behind this design was to be able to check if the answers from a user seemed to be consistent inside the categories without making it too easy for the users to "copy" answers from one question to the next.

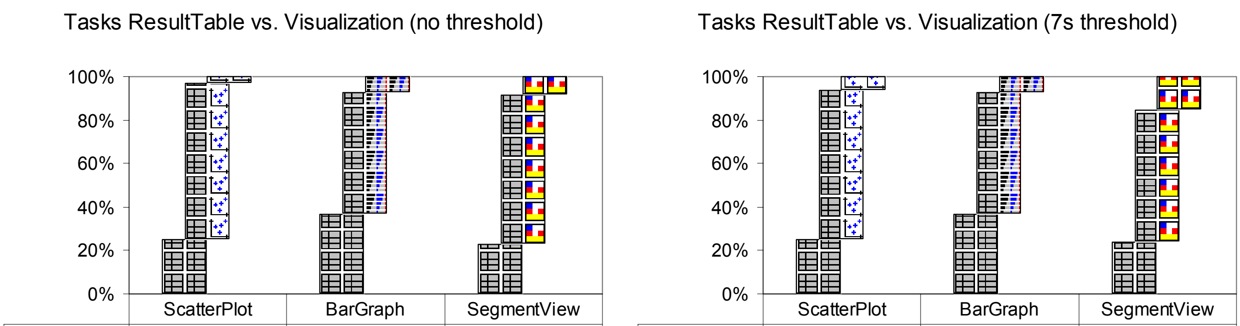
One question asked if the introduction to the system was understandable and detailed enough. The four questions in the category "Suitability for the task" included the suitability of the visualizations, their helpfulness, their joint value, and whether the users would like to have this type of possibilities in the future. The four questions in the category "Ease of use" included the visualization with the easiest usage, unnecessary disruptions, the orientation of the user after the change of the visualization, and whether any visualizations seemed unnecessary. The four questions in the category "Self descriptiveness" included the most and the least understandable visualizations, requests for additional information, and whether a successful usage of the visualizations would still be possible after a longer period of time. The four questions in the category "Suitability for learning" included the intuitive learnability of the system, training effects, confusion, and to what degree the users thought they had mastered the application. The three questions in the category "Conformity with user expectations" included missed functionalities, inconsistencies, and improvement possibilities. The three questions in the category "Confidence" included the visualization that provided the best support, subjective assessment of whether everything had been done correctly, and whether the visualizations are better then the systems already known to the users. The four questions in the category "Mood" included mood before and after the participation in the test, fun, frustrations, and whether the usage of the visualizations was a waste of time. The three questions in the category "Design and Layout" included design, layout improvement possibilities, and which of the visualizations were badly structured or overloaded.

Due to an error six users were given an outdated page in the five-page questionnaire. They filled out on this page the earlier version used in the pre-test, instead of the final version for the main evaluation. Fortunately, this page of the earlier version had only slightly different formulations for seven questions. When reporting the results, both versions will be listed.

4.3.6. Evaluation: results

4.3.6.1. Expected added value

As regards the user interface the evaluation setting contained two conditions where the users were forced to use to use a single component: the HTML-List () or the ResultTable (). In three conditions, the users were free to use the ResultTable only (), the Visualization plus ResultTable (/ /), or just the visualization (/ /). As explained in Chapter 4.3.4.4, we had the idea of measuring the value of a visualization expected by the users by measuring usage times, assuming that the users would utilize the visualization to support their task based on their expectations about its added value. Figure 142 shows in how many cases each of the three possibilities was used to solve a task. Which of the components finally led to success, was not recorded for several reasons. The most important ones were that there was not always a success and that often the ResultTable as well as the additional visualization may have contributed. In several cases, the users clicked on a tab to make a component appear, but they did not really use it. After discussing whether this should be recorded as usage or not, we decided to define a threshold value for usage or not. All components that were used no longer than seven seconds are not considered to be really used for a task. The threshold value of seven seconds was chosen, because the shortest task to complete a task was eight seconds. The left half of Figure 142 shows the original results without the seven second threshold, the right part with this threshold. In the remainder, the results from the threshold variant were used. Despite the fact that visualizations were available, the users decided in about 28% of the tasks to use only the ResultTable: 25% in the ScatterPlot condition, 36% in the BarGraph condition, and 24% in the SegmentView condition.

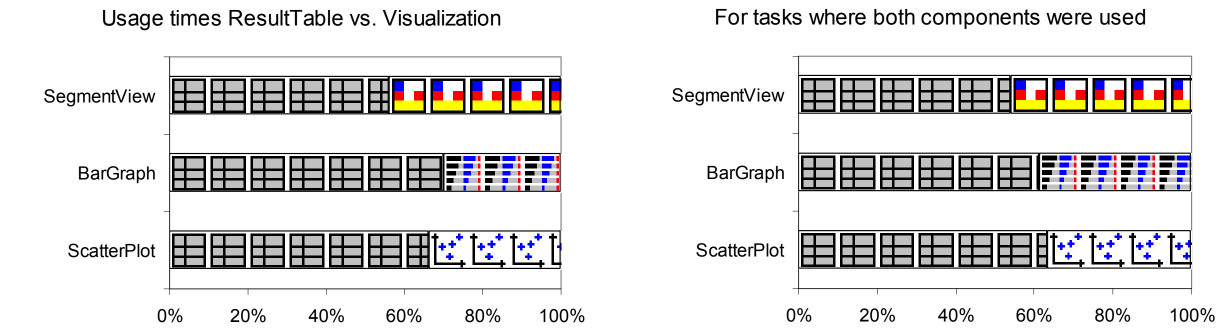


(no threshold)	ScatterPlot	BarGraph	SegmentView
Visualization	3 tasks	7 tasks	8 tasks
Visualization + ResultTable	69 tasks	54 tasks	66 tasks
ResultTable	24 tasks	35 tasks	22 tasks

(7s threshold)	ScatterPlot	BarGraph	SegmentView
Visualization	6 tasks	7 tasks	15 tasks
Visualization + ResultTable	66 tasks	54 tasks	58 tasks
ResultTable	24 tasks	35 tasks	23 tasks

Figure 142: Tasks ResultTable vs. Visualization

The left half of Figure 143 shows the summed usage times of the components for all users over all tasks in the three conditions where they had the choice. From the perspective not of the number of tasks grouped by usage decisions but of summed usage times, the figure reveals that in the Visualization plus ResultTable conditions, the users spent most of their time using the ResultTable. When they had the choice between the ScatterPlot and the ResultTable, they spent 34% of the time using the ScatterPlot and 66% the ResultTable. The values for the BarGraph are 30% / 70%, for the SegmentView 44% / 56%. Whereas the left half of Figure 143 shows the complete dataset used in Figure 142, the right half of Figure 143 shows only the usage times when the users decided to use both possibilities for a task (Visualization + ResultTable, 7s threshold). The picture is the same: users always spent less time for the Visualization than for the ResultTable: 37% for the ScatterPlot, 39% for the BarGraph, and 46% for the SegmentView.



	ScatterPlot	BarGraph	SegmentView
∑ time Visualization	181 min	151 min	230 min
∑ time ResultTable	355 min	353 min	292 min

	ScatterPlot	BarGraph	SegmentView
∑ time Visualization	153 min	131 min	167 min
∑ time ResultTable	262 min	207 min	197 min

Figure 143: Usage time ResultTable vs. Visualization

There were some minor differences in the usage behavior between Experts and Beginners, but the general trends were the same. Figure 144 shows the values from the left half of Figure 143 as detailed usage patterns per user. Individual users are coded as follows: "group"-"user number in group". The five groups are the groups described in Chapter 4.3.5.5. Every group included four beginners and four experts. The difference among the groups were the component selections for the twelve tasks. The user number in the group is a digit from one to eight, which codes the efficiency rank of the user in his group. The efficiency rank is used to present the results, because ranking enhances in several cases the readability of the figures compared to using the user's sequence number[140]. To cite two examples: "1-1" was the most efficient user in group 1, "4-8" was the least efficient user in group 4.

When examining the usage patterns in detail, it is also apparent, that two users never used anything but the ResultTable in the Visualization plus ResultTable condition (2-1 and 4-1). They neither used the ScatterPlot, the BarGraph, or the SegmentView.

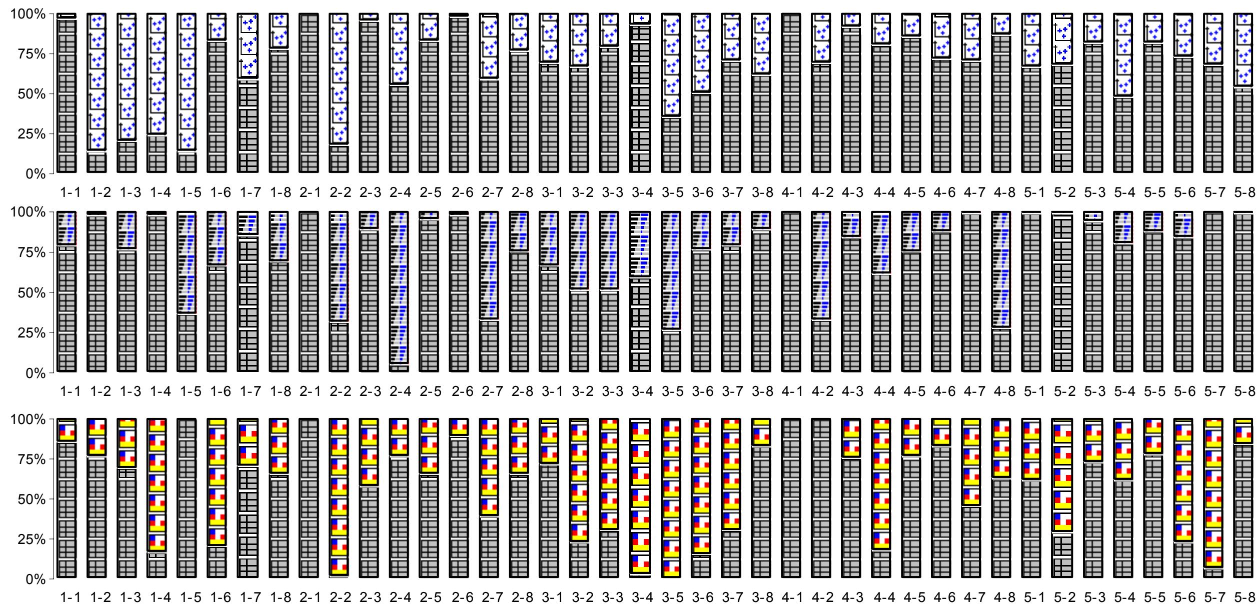


Figure 144: Usage patterns of the visualizations

[140] 1-6 Expert - 1-3 Expert - 1-2 Beginner - 1-5 Expert - 2-7 Expert - 2-1 Expert - 1-8 Beginner - 2-2 Expert - 2-6 Beginner - 3-7 Expert - 3-8 Expert - 3-1 Expert - 3-6 Beginner - 4-2 Expert - 4-5 Expert - 4-7 Expert - 5-2 Expert - 5-5 Expert - 1-4 Beginner - 4-6 Beginner - 1-1 Expert - 5-1 Beginner - 2-3 Beginner - 2-4 Beginner - 3-3 Beginner - 3-4 Expert - 2-8 Expert - 3-5 Beginner - 3-2 Beginner - 5-4 Expert - 5-3 Beginner - 4-4 Beginner - 5-6 Beginner - 4-1 Beginner - 2-5 Beginner - 4-3 Beginner - 4-8 Expert - 1-7 Beginner - 5-7 Expert - 5-8 Beginner

It may be concluded that in the Visualization plus ResultTable conditions, where the user had the choice of deciding which component to use, both components were used in the majority of cases. When analyzing usage times in these conditions, the ResultTable was the favorite component of the users. It was used in all three user interface conditions with ScatterPlot, BarGraph, and SegmentView more than 50% of the overall task time. Interpreting usage time as an indicator for expected value, the expected value for the users of the ResultTable seemed to be higher than that of the other components. Usage time of a component could be a misleading indicator for expected value, however, because it is possible that usage of the component is necessary for a certain task, despite its not being favored by the user. When combined with the results from the questionnaire, usage time may be an indicator for expected value. According to usage time ratios the ResultTable has the highest expected value, followed by the SegmentView, the BarGraph, and the ScatterPlot. The HTML-List was not included in this comparison because the usage time portions could only be calculated for the Visualization plus ResultTable conditions.

4.3.6.2. User Satisfaction

The results from the questionnaire are hereinafter presented grouped by category. The question concerning the introduction to the system showed that the introduction was clearly understandable and detailed enough. In response to the selection question “In retrospect did you find the introduction for a first operation: too short / detailed enough / confusing / understandable”[141] only one user checked „too short”, and another one “confusing”. 28 users answered “detailed enough” and 29 “understandable”. Multiple selections were possible.

4.3.6.2.1. Suitability for the task

The four questions concerning the suitability of the system for the tasks were all focused on the entire system. None asked for a specific rating of individual user interface components. For results about the confidence of the users in the helpfulness of the individual components to solve the tasks see Figure 161 on page 186.

Figure 145 shows the results for the rating question “How well are the visualizations generally adapted to the work demands - in particular to the present tasks –: very badly … very well”[142]. Only 40% of the users thought that the visualizations were well or very well adapted to the demands; 38% passably, and 23% badly.

Figure 145 and the following bubble-diagrams show the number of users who checked a certain option. The middle row with gray bubbles shows the summed values for all 40 users, the upper row shows the values for the experts, and the lower row the values for the beginners.

---

[141] German original: “8. Finden Sie die Einführung im Nachhinein für ein erstmaliges Arbeiten: zu knapp / ausführlich genug / verwirrend / verständlich”

[142] German original: “3. Wie sind allgemein die Visualisierungen auf die Anforderungen der Arbeit – insbesondere die vorliegenden Problemstellungen – zugeschnitten: sehr schlecht … sehr gut”

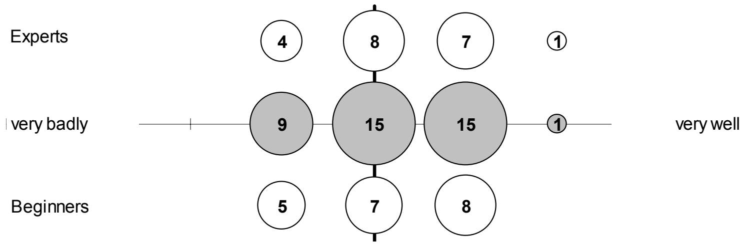


Figure 145: Suitability: Adaptation to the demands

Despite the fact that the adaptation could have been better, the majority of the users rated the visualizations as helpful. In response to the rating question "How helpful would you rate the visualizations in supporting work: not helpful at all … very helpful"[143]. 83% of the users rated the visualizations as helpful or very helpful, 13% as of medium helpfulness.

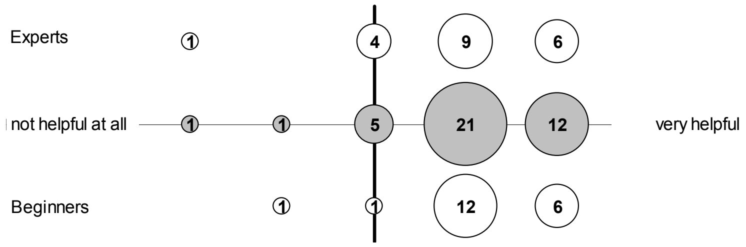


Figure 146: Suitability: Helpfulness to support work

Interestingly, of the two users who never used ScatterPlot, BarGraph, or SegmentView, one checked "not helpful at all" and the other one "very helpful". In the latter case, a comment was added that this holds for the entire system. Accordingly, for this and the other questions asking generally about the rating of the "visualization" without specifying the components, the comments may to an extent be independent from the ratings for ScatterPlot, BarGraph, and SegmentView.

In response to the alternatives question "Would you like in future to have the possibility of being supported by visualizations like those used today: do not agree – undecided – agree"[144] 88% of the users agreed.

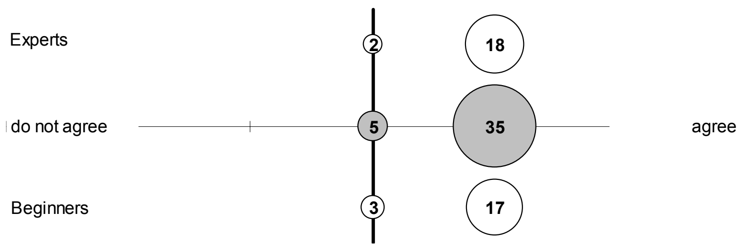


Figure 147: Suitability: Want be supported by these visualizations in the future

Asked for the joint value of the visualizations by the alternative question "Do you think that the visualizations complement each other well: do not agree – undecided – agree"[145] again a majority agreed, but the proportion of users was with 63% smaller than for the previous question.

---

[143] German original: "6. Wie hilfreich würden Sie die Visualisierungen als Arbeitsunterstützung einschätzen: gar nicht hilfreich … sehr hilfreich"

[144] German original: "26. Hätten Sie in Zukunft gerne die Möglichkeit, auf solch eine Unterstützung durch Visualisierungen wie heute zurückgreifen zu können: nicht zustimmen – unentschieden – zustimmen"

[145] German original: "29. Finden Sie, daß die Visualisierungen sich gut ergänzen: nicht zustimmen – unentschieden – zustimmen"

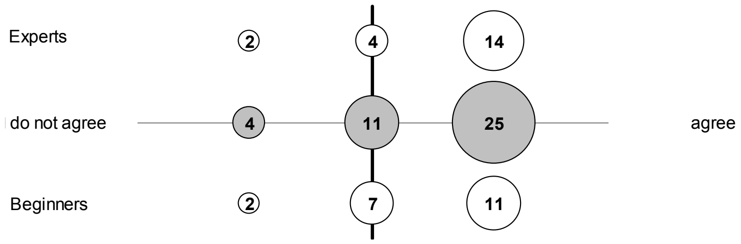


Figure 148: Suitability: Visualizations complement each other

Summarizing the results of the questions concerning the suitability of the system for the tasks, it can be said that the users perceived the visualizations as helpful and complementary each other. Asked if they would like to have this type of visualizations in future, they responded positive. The adaptation of the visualizations to the demands of work leaves something to be desired.

4.3.6.2.2. Ease of use

Of the four questions concerning "ease of use" two addressed the system in general and two offered the possibility of distinguishing among the different components. Starting with the former pair, the first question dealt with possible disorientation problems encountered when changing between the visualizations. Unfortunately, the question did not specify if the "change" addressed switches between the ResultTable and a Visualization or from one task to the next with a change of the components. Accordingly, the answers to the rating question "After a change from one to another visualization I can reorientate myself: very badly … very well"[146] are to some degree interpretable. 75% of the users reported a good or very good reorientation, 10% reported a bad or very bad reorientation.

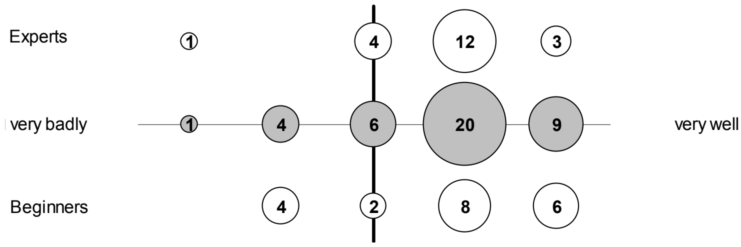


Figure 149: Reorientation after changing the visualization

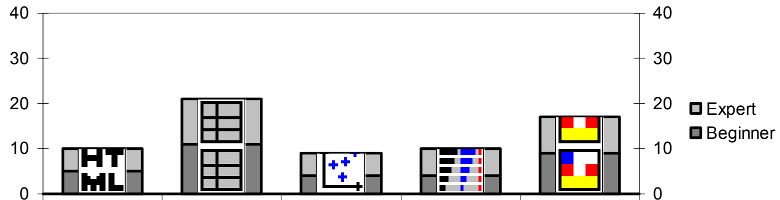
When asked "Which events forced an unnecessary interruption of your work?"[147] more than half of the users complained about the speed of the system. Reasons given included calculating times for the SegmentView, loading times for documents in the internal document browser of the INSYDER system or an external browser, and system crashes. In particular the internal browser had occasionally loading times of some seconds despite the fact that the documents had been stored locally. One user experienced problems loading documents from the ScatterPlot. Another user complained that for each new task he had to change the settings of the ResultTable back to his preferences, because they had been restored to the default upon loading a new Sphere of Interest.

[146] German original: "4. Nach einem Wechsel von einer in die andere Visualisierung finde ich mich wieder wie folgt zurecht: sehr schlecht … sehr gut"

[147] German original: "23. Welche Vorgänge haben eine unnötige Unterbrechung Ihrer Arbeit erzwungen?"

For the selection question “With which visualization(s) did you manage to work best? Can you give a reason?”[148] the ResultTable and the SegmentView received the highest values. Figure 150 shows the results. 53% of the users voted for the ResultTable, 43% for the Segment View, 25% for the HTML-List and the BarGraph each, and 23% for the ScatterPlot. Multiple selections were possible.

Figure 150 and the following bar charts show the values for selection questions that offered the possibility of selecting one or more of the five components used in the INSYDER system. For each component the middle bar shows the summed values for all 40 users; the dark gray bar shows the values for the beginners; and the light gray bar the values for the experts. The questions that asked for specific answers for the components included in most cases an additional option of “none”. The “none” values are not reported, except in the case of one question in Chapter 4.3.6.2.7 where it makes really sense.



	HTML-List	ResultTable	ScatterPlot	BarGraph	SegmentView
Expert	5	10	5	6	8
Beginner	5	11	4	4	9
Total	10	21	9	10	17

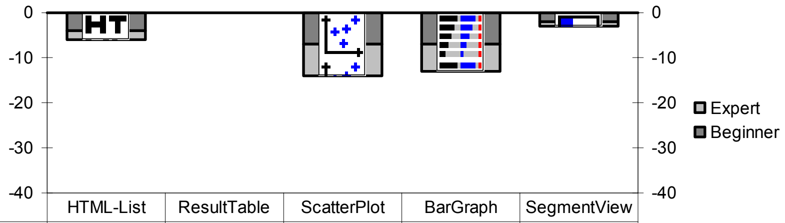
Figure 150: Components best to work with

Reasons given as to why specific components were best to work with included for the HTML-List familiarity; for the ResultTable clarity, sorting possibilities, and the similarities with the familiar HTML-List; for the ScatterPlot the mathematical notion, the configurability, and the possibility of combining two keyword rankings; for the BarGraph space economy, easy understandability and comprehensability of the ranking; for the SegmentView the comprehensability of the ranking, and the possibility of examining parts of the documents without too much reading. The relatively low value for the HTML-List is surprising, as it should have been the form of presentation most familiar to the users. When looking at the results it must be noted that every user had the possibility of utilizing the ResultTable in nine or ten out of the twelve tasks, whereas the other components were only available in two or three tasks. For details about the setting see Table 37 on page 174.

When asked which components were extraneous, the majority of the users chose the ScatterPlot and the BarGraph in response to the question “The number of offered visualizations appears to me excessive. I personally would do without the following”[149], 35% would do without the ScatterPlot, 33% without the BarGraph, 15% without the HTML-List, and 8% without the SegmentView. No one wanted to do without the ResultTable.

[148] German original: “16. Mit welcher (welchen) Visualisierung(en) kamen Sie am besten zurecht? Können Sie dafür einen Grund nennen?”

[149] German original: “9. Die Zahl der angebotenen Visualisierungen erscheint mir zu groß. Ich persönlich würde auf folgende Visualisierung(en) verzichten. / 9. Die Zahl der angebotenen Visualisierungen erscheint mir zu groß. Ich persönlich würde auf folgende verzichten.”



	HTML-List	ResultTable	ScatterPlot	BarGraph	SegmentView
Expert	-2	0	-7	-6	-1
Beginner	-4	0	-7	-7	-2
Total	-6	0	-14	-13	-3

Figure 151: Dispensable components

4.3.6.2.3. Self descriptiveness

In the category self-descriptiveness one question concerned the overall system and three its different components. For the general alternatives question "Do you think that you would still be able to work with the visualizations after a longer break: do not agree – undecided – agree"[150] 88% of the users agreed, while the rest were undecided.

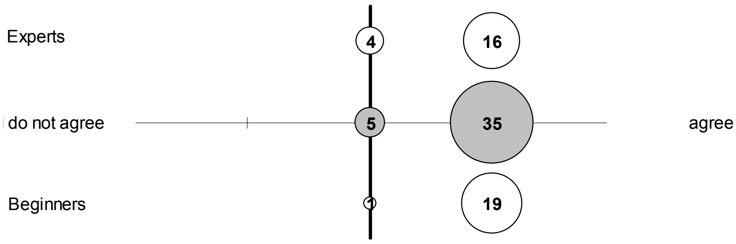
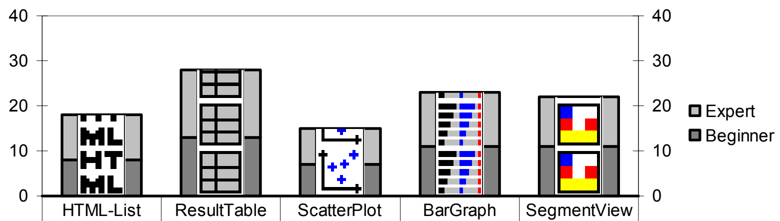


Figure 152: Able to work with the components after a longer break

When asked directly which of the components had been self-describing the ResultTable, the BarGraph and the SegmentView received the highest rankings. For the question "Which visualization(s) appear(s) to you to be most plausible and self-describing"[151] 70% of users voted for the ResultTable, 58% for the BarGraph, 55% for the SegmentView, 45% for the HTML-List, and 38% for the ScatterPlot. Surprising again is the low value for the HTML-List.



	HTML-List	ResultTable	ScatterPlot	BarGraph	SegmentView
Expert	10	15	8	12	11
Beginner	8	13	7	11	11
Total	18	28	15	23	22

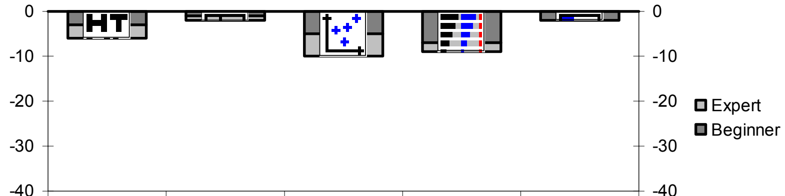
Figure 153: Self descriptiveness of the components

When asked the somewhat inverse question "The benefit of the following visualization(s) is unclear to me"[152], interestingly users' positiveness towards the BarGraph declined relative to that towards the SegmentView. Whereas the two visualizations were considered to be similarly plausi-

[150] German original: "27. Könnten Sie Ihrer Einschätzung nach auch noch nach einer längeren Pause mit den Visualisierungen umgehen: nicht zustimmen – unentschieden – zustimmen"

[151] German original: "11. Welche Visualisierung(en) schien(en) Ihnen am einleuchtendsten und ist (sind) sozusagen selbsterklärend. / 11. Welche Visualisierungen schienen Ihnen am einleuchtendsten und verstehen sich praktisch von selbst."

ble and self-describing, the potential benefit of the BarGraph was not as clear to the users as that of the SegmentView. In particular beginners had difficulty perceiving the benefit of the BarGraph. Overall 25% of users had had difficulty perceiving the benefit of the ScatterPlot, 23% of the BarGraph, 15% of the HTML-List, and 5% of either the SegmentView or the ResultTable.



	HTML-List	ResultTable	ScatterPlot	BarGraph	SegmentView
Expert	-3	-1	-5	-2	0
Beginner	-3	-1	-5	-7	-2
Total	-6	-2	-10	-9	-2

Figure 154: Benefit unclear

When asked "Which visualization(s) should offer additional information? Which explanations would have helped you there?"[153], the majority of the comments (sixteen) referred to the ScatterPlot. The ResultTable was the subject of eleven comments, the BarGraph and the SegmentView of eight each, and the HTML-List of seven. Regarding the ScatterPlot users would have liked to have explanation of the 1970-effect, which was explained on page 143 of this thesis but not in the introduction given to users. An additional feature requested for the ScatterPlot was an indication of the relevance for all combinations of dimensions, not just when the relevance was mapped to one of the axes. Users also requested an enhanced capability to handle document groups displayed as a square. Regarding the ResultTable multiple criteria sorting possibilities and the document tooltip available for ScatterPlot, BarGraph and SegmentView was missed. Regarding the BarGraph Title, Abstract and URL were considered lacking. In general, users asked for a feature that would automatically mark visited pages. The select-flag that was available in all components except the HTML-List for such a purpose obviously did not fulfill users' expectations of such a feature.

4.3.6.2.4. Suitability for learning

The four questions concerning the suitability of the system for the tasks were all focused on the whole system. When directly asked about the learnability of the system, 58% of the users were in agreement with the statement "The application is intuitively learnable; handling it requires little time and hardly any external assistance: do not agree – undecided – agree"[154]. As shown in Figure 155, experts and beginners interestingly had different opinions. Whereas the beginners largely agreed, the opinions of the experts were less sure.

[152] German original: "14. Den Nutzen folgender Visualisierung(en) kann ich nur schwer nachvollziehen. / 14. Den Nutzen folgender Visualisierungen kann ich nur schwer nachvollziehen."

[153] German original: "18. In welcher (welchen) Visualisierung(en) sollten noch Zusatzinformationen angeboten werden? Welche Erklärungen hätten Ihnen dort weitergeholfen?"

[154] German original: "24. Die Anwendung ist intuitiv erlernbar; der Umgang mit ihr erfordert wenig Zeit und kaum fremde Hilfe: nicht zustimmen – unentschieden – zustimmen"

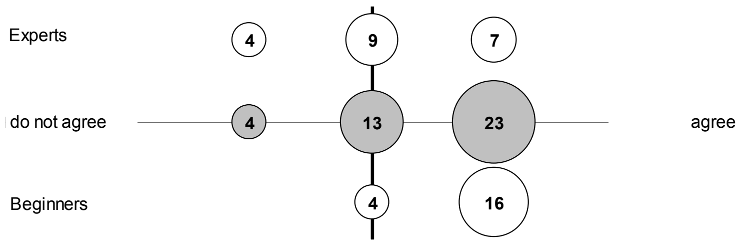


Figure 155: The application is intuitively learnable

Users were asked about learning effects with the question "In the course of the investigation, you felt a learning effect, and you think now that you are able to work with the visualizations better: do not agree – undecided – agree"[155]. Only one user did not agree, while 90% of the users agreed. (Figure 156 shows the results.)

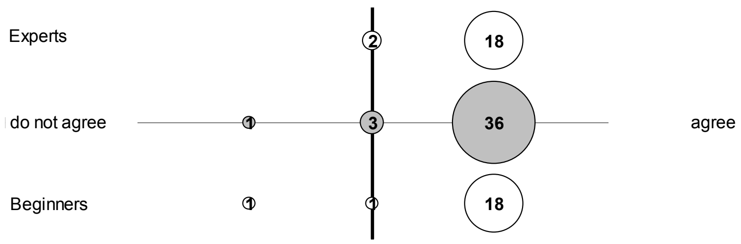


Figure 156: Felt learning effects, now able to manage the work with the components better

Turning now to how well the users thought they were able to operate the application at the end of the test, Figure 157 shows the answers for the question "How well in your opinion have you mastered this application: very badly … very well[156]". At the end of the test 63% of the users believed that they could now operate the application well or very well, only 8% thought that they did so badly.

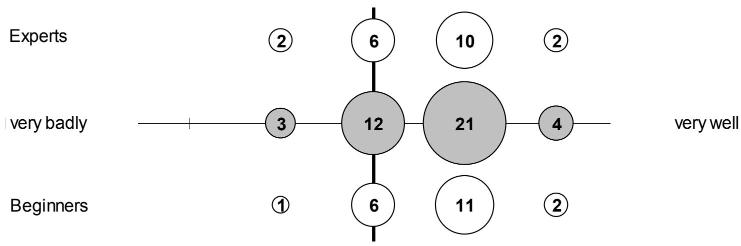


Figure 157: Ability to operate the application at the end of the test

Figure 158 shows that 40% of the users answered "yes" when asked "Were you confused when using the visualizations? yes – no. If yes, for what reason?"[157]. A reason listed several times was the missed marking of already visited documents. Other reasons included that the visualizations in general are too unfamiliar, not intuitive, or graphically overloaded. Users reported problems re-identifying documents when changing the view including sorting or when changing the component. With the ScatterPlot specifically users reported problems with document group symbols and colors. In the ResultTable, it was reported that the abstracts of the documents had hardly been readable. One user was confused because not all documents contained text. Due to the exclusively local storage of the HTML-files and the disconnection from the Internet framesets and documents

[155] German original: "28. Im Laufe der Untersuchung spürten Sie einen Lerneffekt und denken, nun besser mit den Visualisierungen umgehen zu können: nicht zustimmen – unentschieden – zustimmen"

[156] German original: "2. Wie gut beherrschen Sie Ihrer Meinung nach jetzt diese Anwendung: sehr schlecht … sehr gut"

[157] German original: "17. Waren Sie beim Arbeiten mit den Visualisierungen verwirrt? ja – nein. Wenn ja, was war der Grund?"

relying on figures frequently lack text. Another user reported confusion because it was unclear to him that the SegmentView was based only on the text of the files and did not include figures.

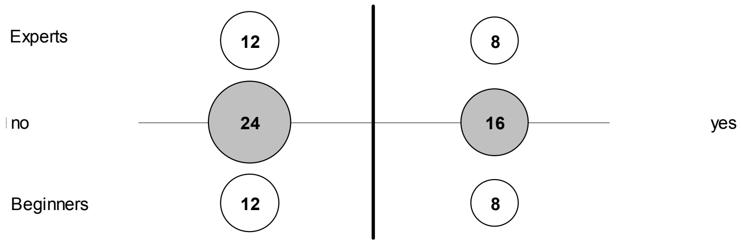


Figure 158: Confused when working with the visualizations

4.3.6.2.5. Confidence

Of the three questions covering confidence two concerned the overall system and one asked specifically after the individual components. The answers for the first alternatives question "How sure are you, that you have always proceeded correctly: very unsure … very sure"[158] showed that there was definitely a degree of uncertainty among the users as to whether they had done everything right. As shown in Figure 159, only 25% of the users were sure or very sure that they had always proceeded correctly. This uncertainty could have been caused by the tasks or by the system itself. Nevertheless, the answer suggests that additional training or a longer introduction might have influenced the results of the evaluation.

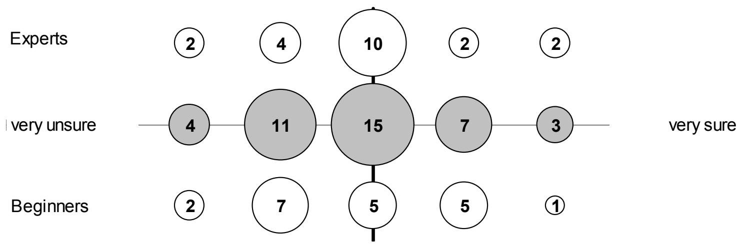


Figure 159: Confidence about having always proceeded correctly

Concerning the visualizations themselves 88% of the users agreed when asked "Do you consider the representations of search results with visualizations to be more efficient than those from other common search engines, that you know so far: do not agree – undecided – agree"[159]. (Figure 160 shows the detailed results.)

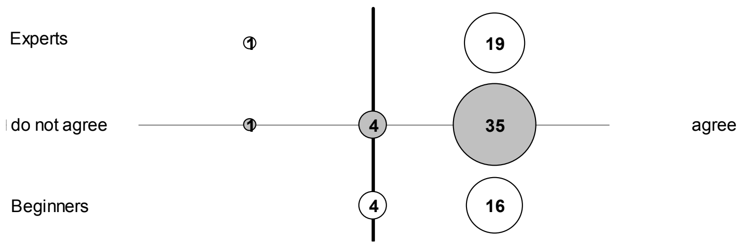
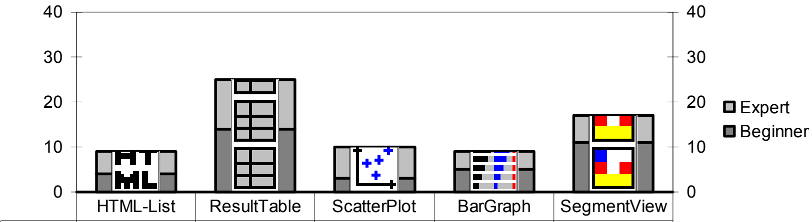


Figure 160: Greater efficiency of visualizations compared to search engines known to the users

[158] German original: "5. Wie sicher sind Sie sich, immer richtig vorgegangen zu sein: sehr unsicher … sehr sicher."

[159] German original: "30. Halten sie die Darstellung von Suchergebnissen mit Visualisierungen für leistungsfähiger als die von anderen gängigen Suchmaschinen, die Sie bisher kennengelernt haben: nicht zustimmen – unentschieden – zustimmen"

When directly asked "Which visualization(s) - in your opinion - most helped you to solve the given problem?"[160], the users were most confident with the ResultTable. Besides the ResultTable Beginners were especially confident with the SegmentView, whereas experts were also confident with the ScatterPlot. (Figure 161 shows the detailed results.) 63% of the users voted for the ResultTable, 43% for the SegmentView, 25% for the ScatterPlot, and 23% for each of the HTML-List and the BarGraph.



	HTML-List	ResultTable	ScatterPlot	BarGraph	SegmentView
Expert	5	11	7	4	6
Beginner	4	14	3	5	11
Total	9	25	10	9	17

Figure 161: Components that helped most to solve the given problem

4.3.6.2.6. Design and Layout

Of the three questions about design and layout one concerned the overall system and two its different components. In general, the design of the visualizations seemed to be fairly acceptable. In response to the question "How did you find the design of the visualizations: not at all appealing … very appealing"[161] 80% of the users checked appealing or very appealing. Nevertheless, four users answered not appealing. (Figure 162 shows the details.)

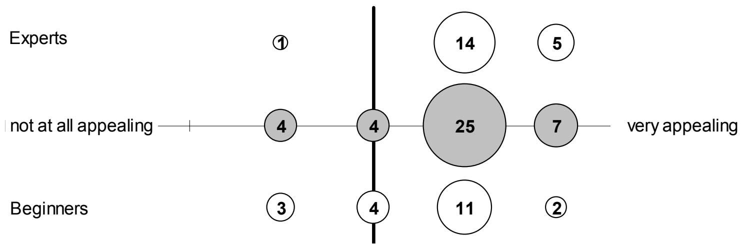


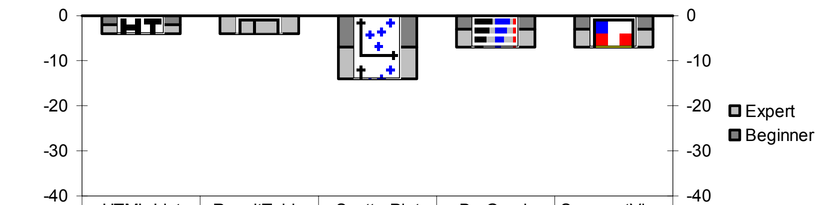
Figure 162: Design of the components

Regarding the individual components the ScatterPlot was named by 35% of the users when they were asked, "Which visualization(s) did you perceive as too complex or overloaded?"[162]. (Figure 163 shows the details.) 18% named the BarGraph and the SegmentView each and 10% the HTML-List and the ResultTable each. It is interesting to note that none of the beginners perceived the ResultTable with the Relevance Curve as too complex or overloaded, but that 20% of the experts did.

[160] German original: "13. Welche Visualisierung(en) hat (haben) Ihnen - Ihrer Meinung nach - am besten geholfen, das vorgegebene Problem zu lösen? / 13. Welche Visualisierungen hat Ihnen - Ihrer Meinung nach - am besten geholfen, das vorgegebene Problem zu lösen?"

[161] German original: "7. Wie fanden Sie die optische Gestaltung der Visualisierungen: gar nicht ansprechend … sehr ansprechend"

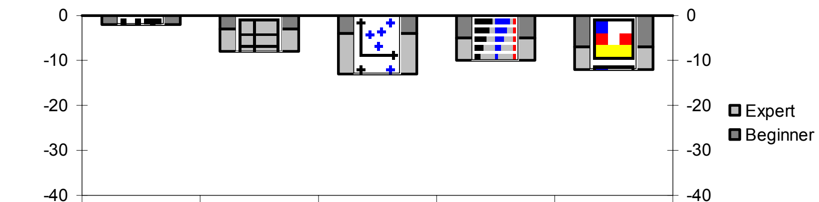
[162] German original: "10. Welche Visualisierung(en) empfanden Sie als zu unübersichtlich bzw. überladen? / 10. Welche Visualisierung empfanden Sie als zu unübersichtlich bzw. zu überladen?"



	HTML-List	ResultTable	ScatterPlot	BarGraph	SegmentView
Expert	-2	-4	-7	-4	-4
Beginner	-2	0	-7	-3	-3
Total	-4	-4	-14	-7	-7

Figure 163: Too complex or overloaded components

Figure 164 shows the answers to the question "The design of which of the following visualization(s) would you improve?"[163]. The ScatterPlot is again the leading candidate for changes, being named by 33% of the users, in particularly experts. The SegmentView got with 30% also a high value, but in this case as revealed by additional comments, at least some users had not only the visual design in mind but also the system design. The SegmentView had the longest answer times of all visualizations, because the segment values were calculated on the fly every time instead of being stored in the database like the other attributes. 25% suggested the BarGraph for design improvements, 20% the ResultTable, and 5% the HTML-List.



	HTML-List	ResultTable	ScatterPlot	BarGraph	SegmentView
Expert	0	-5	-9	-5	-5
Beginner	-2	-3	-4	-5	-7
Total	-2	-8	-13	-10	-12

Figure 164: Candidates for design improvements

4.3.6.2.7. Mood

Of the questions concerning mood all but one tried to elicit impressions generally rather than regarding specific components. Figure 165 shows the results for the question "Do you regard on the basis of your recent experiences the usage of visualizations as a waste of time? (Please consider that the present waiting periods are a result of the "tardiness" of the software functions and have less to do with the visualizations themselves): do not agree – undecided – agree"[164]. 15% of the users, or 6 users in absolute terms, thought that the usage of the visualization was a waste of time or were at least undecided.

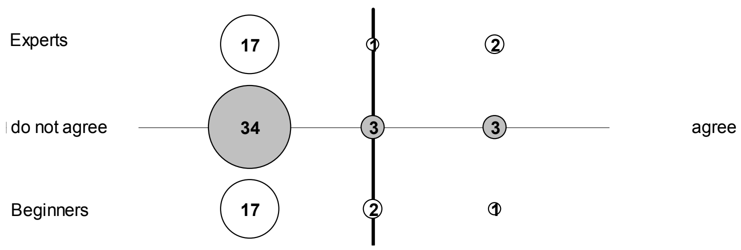
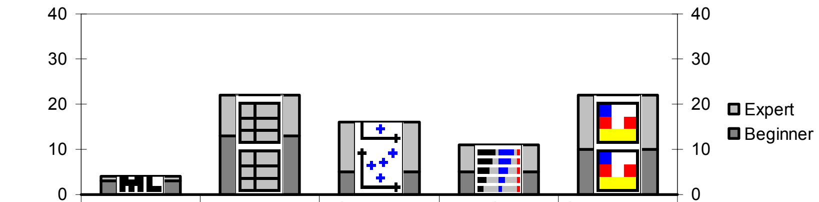


Figure 165: Using the visualizations was a waste of time

---

[163] German original "15. Das Design welcher der folgenden Visualisierung(en) würden Sie verbessern? / 15. Das Design welcher der folgenden Visualisierungen würden Sie verbessern?"

When asked whether "The usage of this/these visualization(s) was really fun"[165], the ResultTable and the SegmentView were most often named by the users. (Figure 166 shows the detailed values.) 55% of the users enjoyed working with the ResultTable or the SegmentView to a degree, 40% with the ScatterPlot (especially experts), 28% with the BarGraph, and 10% with the HTML-List. Two users checked "none" for this question. Nevertheless, one of the two had answered "undecided" and the other "do not agree" to the previous question about whether they regarded usage of the visualizations as a waste of time.



	HTML-List	ResultTable	ScatterPlot	BarGraph	SegmentView
Expert	1	9	11	6	12
Beginner	3	13	5	5	10
Total	4	22	16	11	22

Figure 166: Fun factor of the components

Some tasks were really hard work. To get an impression if work with the system positively or negatively influences the mood of the candidates, there was a relevant question in the entry questionnaire to be filled out before the test and a corresponding one in the closing questionnaire after the test. Figure 167 shows the answers to the question "How do you currently feel: very bad… very good"[166] from the entry questionnaire. The left part of Figure 168 shows the answers to the question "How would you now after the test define your mood: very bad… very good"[167] from the closing questionnaire. The right part of the same figure shows the changes. Grey bubbles in the diagonal show the number of users who gave the same answer after the test as before. Starting from the upper right corner, it can be seen that eight users answered both times "very good", eleven users "good" … The green bubbles above the diagonal show users who were in a better mode afterwards than before, the red bubbles below the diagonal show the opposite. Two users who answered before the test "good", answered after the test "very good"; two users that answered before the test "very good", answered "good" after the test; and an additional user dropped one category further. Overall, 60% of the users remained unchanged, 25% were in better mood afterwards, and 15% were in worse mood afterwards.

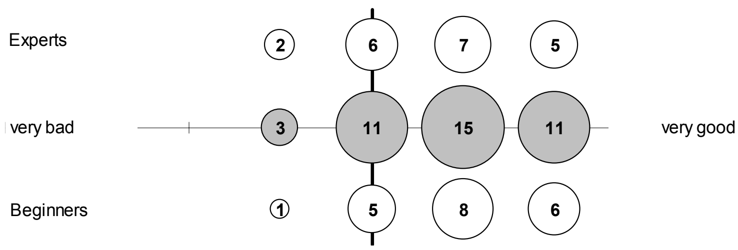


Figure 167: Mood before the test

[164] German original: "25. Halten Sie nach Ihren gerade gemachten Erfahrungen das Arbeiten mit Visualisierungen für Zeitverschwendung? (Bedenken Sie, daß die zur Zeit noch existierenden Wartezeiten durch die „Langsamkeit" der Softwarefunktionen entstehen und weniger mit den Visualisierungen selbst zu tun haben!)"

[165] German original: "12. Das Arbeiten mit dieser (diesen) Visualisierung(en) hat mir viel Spaß gemacht. / 12. Das Arbeiten mit folgender Visualisierung hat mir besonders viel Spaß gemacht."

[166] German original: "Wie fühlen Sie sich gerade: sehr schlecht … sehr gut"

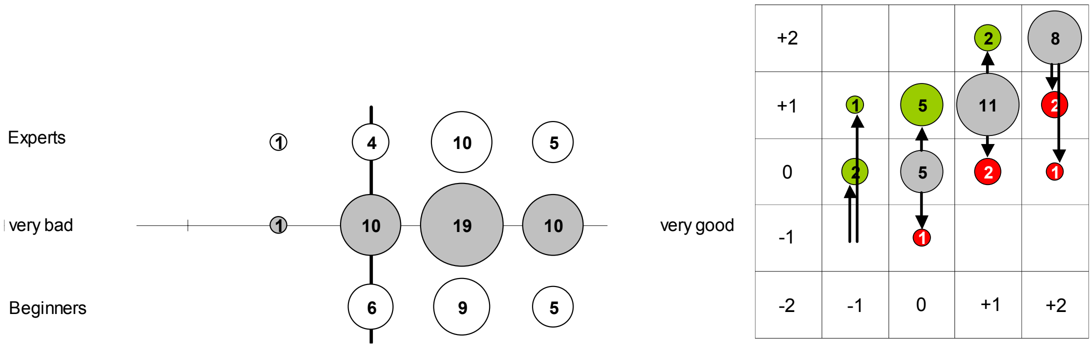


Figure 168: Mood after the test (left) and changes (right)

When the open question “What frustrated you? You can specify several points”[168] directly was posed, the users mentioned above all the low speed of the system (especially of the SegmentView), the inability to follow links from documents in the result set to other documents (which had been part of the evaluation setting), and the tasks with only one keyword in the query (especially with the 500-hit document sets). Additional sources of frustration were system crashes, the already prepared result sets, the inability to refine the search that had been deactivated for the evaluation, the inability to decide which visualization to use, the absence of overview in the HTML-List, difficult decisions as to which dimensions to select in the ScatterPlot, the usage of an external browser to view the documents (even when automatically invoked by clicking on a document in the INSYDER system), repeated visits to pages due to the missing “already visited”-functionality, difficulties in some cases to set the document select-flag, a warm-up period that had been too short to learn all functions, too much (unnecessary) information, excessively large result sets and insufficient time, and last but not least, the fact that not all tasks had been solved successfully.

4.3.6.2.8. Conformity with user expectations

The conformity with user expectations was addressed by three open questions. One question was system oriented and dealt with consistency, the other two aimed more at general user expectations for this type of visualizations. When users were asked “Were you able to uncover inconsistencies in the organization or operation of the individual visualizations? yes – no. If yes, which?”[169], only a few comments were made. Some users remarked that the functionality of the mouse buttons occasionally differed between the components. Others mentioned that the ResultTable had an integrated browser, whereas the other components used an external browser.

The questions “Which functionality(ies) did you miss in the visualizations?”[170], and “What do you additionally consider worth to be improved regarding the visualizations?”[171] elicited comments that included several points already raised in the context of other questions. In addition the following points were made. The users requested for the HTML-List and the ResultTable a string search feature; for the ScatterPlot the possibility of combining attributes for display on one axis and the

[167] German original: “1. Wie würden Sie jetzt nach der Untersuchung Ihre Stimmung bezeichnen: sehr schlecht … sehr gut”

[168] German original: “22. Was hat Sie frustriert? Sie können mehrere Punkte aufführen”

[169] German original “19. Haben Sie Inkonsistenzen in der Gestaltung oder Bedienung der einzelnen Visualisierungen entdecken können? ja – nein. Wenn ja, welche?”

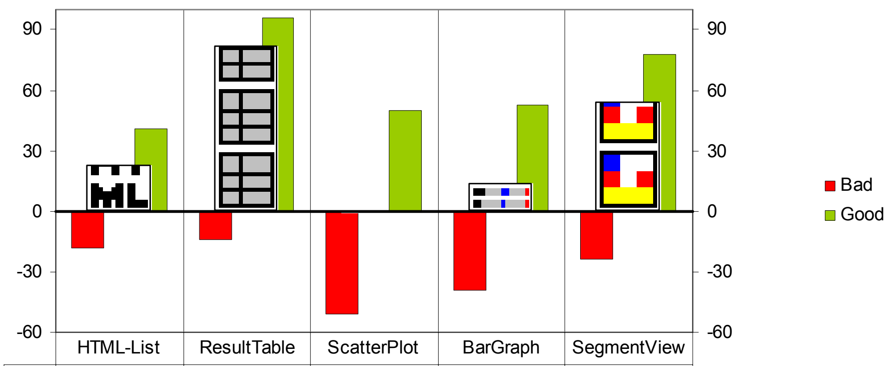
[170] German original “20. Welche Funktionalität(en) haben Sie bei den Visualisierungen vermisst?”

[171] German original “21. Was halten Sie noch in Bezug auf die Visualisierung(en) für verbesserungswürdig?”

usage of an additional third dimension; for the BarGraph more information and more vertical space between the bars; for the SegmentView a possibility of switching off the display of selected keywords. Finally, users would like in general to have a better elimination of doubles; more detailed information for the server type "European"; Boolean retrieval possibilities; the possibility of searching in the result set with a new query; user selectable weighting of keywords; keyword highlighting; sorting by more than one criterion; another look and feel; and a selection possibility for other colors.

4.3.6.2.9. Summary of the Questionnaire results

Figure 169 shows a summary of the ratings for the individual components with positive and negative comments and an overall result. The favorite component of the users was the ResultTable, followed by the SegmentView and the HTML-List. The BarGraph and especially the ScatterPlot found little favor. Figure 169 combines the component-specific values of ease of use (Figure 150: Components best to work with, Figure 151: Dispensable components), self descriptiveness (Figure 153: Self descriptiveness of the components, Figure 154: Benefit unclear), confidence (Figure 161: Components that helped most to solve the given problem), design and layout (Figure 163: Too complex or overloaded components, Figure 164: Candidates for design improvements), and last but not least, mood (Figure 166: Fun factor of the components). In an earlier paper [Reiterer, Mußler, Mann 2001], we presented a somewhat different combination of questions to form an overall impression of user satisfaction about the components. The values there are different, but the sequence of the ranking is the same: ResultTable, SegmentView, HTML-List, BarGraph, ScatterPlot.



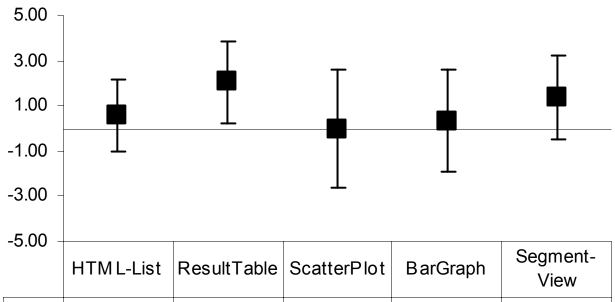
	HTML-List	ResultTable	ScatterPlot	BarGraph	SegmentView
Bad	-18	-14	-51	-39	-24
Good	41	96	50	53	78
Total	23	82	-1	14	54

Figure 169: Summarized user ratings of the components

The ranking of the components derived from the questionnaire was as expected the same as those derived from the usage times. The users' order of preference was the ResultTable followed by the SegmentView, the BarGraph, and the ScatterPlot. The HTML-List was not included in the usage time comparison because the usage time portions could only be calculated for the Visualization plus ResultTable conditions.

Turning now to the hypotheses, hypothesis H1a had been: The ResultTable and the Visualizations produce results in terms of user satisfaction that differ from the results for the HTML-List. A statistical validation of hypothesis H1a with inferential statistical methods may be done on the basis of the questionnaire results selected for Figure 169. Figure 170 shows the mean values and standard deviations for this data. Compared with the HTML-List, the ResultTable and the Segment-

View received higher mean user satisfaction values, the ScatterPlot and the BarGraph lower.



	HTML-List	ResultTable	ScatterPlot	BarGraph	Segment-View
Mean	0.58	2.05	-0.03	0.35	1.35
SD	1.58	1.80	2.63	2.27	1.87

Figure 170: Mean user satisfaction index and standard deviation

The differences in the user satisfaction indexes between the HTML-List and the values from the other visualizations were calculated and tested using a t-test for dependent samples. Table 39 shows the t-values and the corresponding probability values of the two-tailed paired t-test.

Reference	Compared Visualization	t value	Pr > |t|
HTML-List	ResultTable	-4.42	<.0001**
HTML-List	ScatterPlot	1.19	0.2430
HTML-List	BarGraph	0.46	0.6471
HTML-List	SegmentView	-1.82	0.0765

Table 39: Two-tailed paired t-test: user satisfaction index

** highly significant (p<.01)

The user satisfaction for the ResultTable is highly significant higher than that for the HTML-List on the 1% significance level. In the same direction but not significant is the difference between the SegmentView and the HTML-List (p <.1). The values for the ScatterPlot and the BarGraph that performed worse than the HMTL-List, are far from being significant. Hypothesis H1a was therefore confirmed only for the ResultTable. The ResultTable produced results in terms of user satisfaction that differ from the results for the HTML-List.

Figure 171 shows the user satisfaction differentiated by target user group. When distinguishing between experts and beginners, the ScatterPlot received a slightly “positive” overall rating from the experts, whereas the rating from the beginners was slightly “negative”. The “positive” rating of the ResultTable is more pronounced for the beginners, whereas for the HTML-List and the BarGraph it is more pronounced for the experts. The SegmentView got the same “positive” overall rating from both groups.

Experts

	HTML-List	ResultTable	ScatterPlot	BarGraph	SegmentView
Bad	-7	-10	-28	-17	-10
Good	21	45	31	28	37
Total	14	35	3	11	27

Beginners

	HTML-List	ResultTable	ScatterPlot	BarGraph	SegmentView
Bad	-11	-4	-23	-22	-14
Good	20	51	19	25	41
Total	9	47	-4	3	27

Figure 171: Summarized user ratings of the components per user group

Hypothesis H2a had been: The target user group influences how the user satisfaction will be determined by the user interface condition in comparison with the HTML-List. Please note that it is another matter to examine if there are differences in the user satisfaction with the user interface conditions between beginners and experts. In our study, we were not interested in seeing if experts have a higher user satisfaction for a certain user interface condition than beginners. Sample questions behind the hypothesis could instead be: assuming that a beginner is working with the system, which user interface condition will he prefer? Are the preferences of experts different? We do not in general look for significant differences between experts and beginners for a single user interface condition, but for significant differences inside the user groups for user interface conditions. Applying inferential statistical methods now to hypothesis H2a, Figure 172 shows the mean values and standard deviations of the data that were the basis for Figure 171. The overall pattern for all users from Figure 170 is the same for both target user groups. The ResultTable and the SegmentView received higher user satisfaction values than the HTML-List, the ScatterPlot and the BarGraph lower.

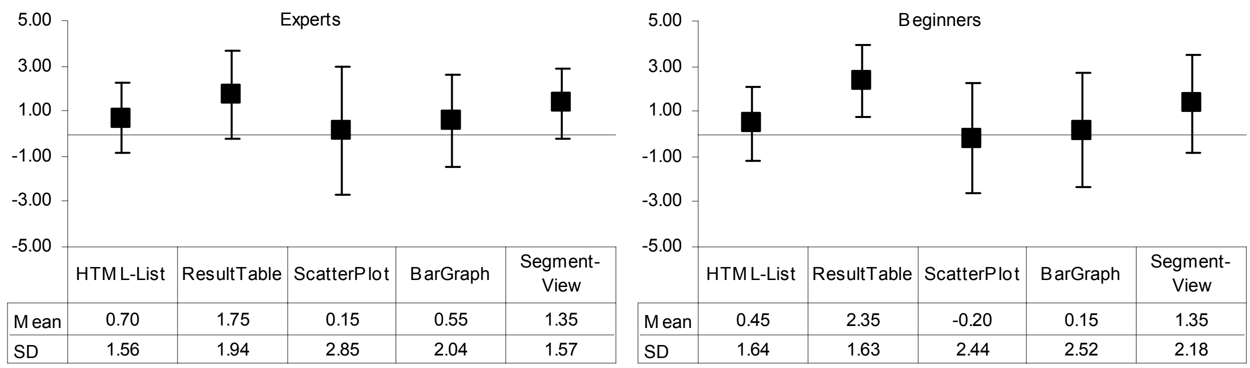


Figure 172: Mean user satisfaction index and standard deviation per user group

In order to check the hypothesis H2a for both groups the differences in the user satisfaction indexes between the HTML-List and the values from the other visualizations were calculated and tested in the same way as for hypothesis H1a but now independently for experts and beginners. (The results of the two-tailed paired t-test are shown in Table 40.)

Reference	Compared Visualization	Experts		Beginners	
		t value	Pr > |t|	t value	Pr > |t|
HTML-List	ResultTable	-1.88	0.0760	-5.36	<.0001**
HTML-List	ScatterPlot	0.69	0.4981	1.01	0.3269
HTML-List	BarGraph	0.22	0.8272	0.42	0.6810
HTML-List	SegmentView	-1.22	0.2388	-1.33	0.1990

Table 40: Two-tailed paired t-test: user satisfaction index per user group

** highly significant (p<.01)

In the beginners' group, the difference between the HTML-List and the ResultTable is highly significant. There are no significant differences for the other components. In the experts' group, all differences are in the same direction as for the beginners' group. The differences do not, however, reach the level of significance for any of the components. It can be said concerning the differences between HTML-List and ResultTable that what was highly significant for the beginners is for the experts at least a trend (p<.1). Interpreting the numbers strictly, Hypothesis H2a was validated for the ResultTable. For the ResultTable the target user group influences how the user satisfaction will be determined by the user interface condition in comparison with the HTML-List. There was a significant difference for the beginners but not for the experts. As a whole, the user satisfaction patterns of experts and beginners are so similar, however, that H2a must be rejected.

Summarizing the results from the questionnaire it can be said that the users experienced several usability problems, but that they in general welcomed the possibilities offered by the Visualizations and the ResultTable. An overall statistical analysis for selected questions revealed relative to the HTML-List highly significant higher user satisfaction values for the ResultTable. The SegmentView showed a trend in the same direction. Differences between experts and beginners were in general very low. When present in some of the individual questions, they are mostly concentrated on the ScatterPlot. In the statistical analysis the difference between HTML-List and ResultTable was significant for the beginners. For the experts it was a trend.

System speed was critical and was negatively biased, in particular by processing times for the SegmentView and loading times or timeout pauses for the internal and external document browsers. Several functions requested by the users had already been implemented in the INSYDER system by the time that the evaluation was performed, but they had been deactivated to ease or control the evaluation setting. These features were not mentioned in the introduction to avoid distracting the users. The users missed some of these features. Other features requested by the users not planned or implemented before the evaluation had interim been integrated. (For details see [Mußler 2002].) A new evaluation with more features activated may be worth undertaking. Such an evaluation might show whether the user satisfaction would be even higher in a full-featured version, or whether the additional complexity would bias the results.

In general, the focus of the user satisfaction questionnaire was on the visualizations and the review of results phase in the information seeking process. The evaluation itself was restricted to this area. User satisfaction concerning the INSYDER system's support of the whole information seeking process may also be interesting to examine.

4.3.6.3. Hard Facts

Turning now to the hard facts derived from the evaluation, some figures will presented so as to attempt an overview from two different angles of the results of the tasks. First, we will look at how the different users performed over all twelve tasks in terms of task time, effectiveness, and efficiency. Second, the same examination will be done summarizing results for questions instead of for users. The validation of the hypotheses and several additional findings will be presented in the subsequent chapters.

All box plots shown below use the legend explained in Figure 173.

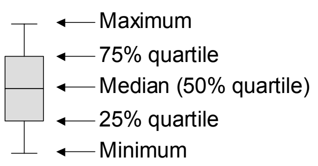


Figure 173: Box plot legend

4.3.6.3.1. Task time, effectiveness, and efficiency per user

Starting with the mean task time per user, Figure 174 shows the values generated over all twelve questions. User 2-1, 2-3, and 4-1 were the three fastest users with a mean task time of 186, 210, and 253 seconds. User 4-8 was the slowest with 442 seconds. The latter time was 138% longer than that of the fastest user and very close to the upper limit of 450 seconds. The upper limit is the total of the maximum allowed task time for each of the twelve tasks[172]. Among the three fastest users 2-1 and 4-1 were the two users mentioned in Chapter 4.3.6.1 who never used anything but the ResultTable in the Visualizations plus ResultTable condition. In general, the tasks times vary considerably.

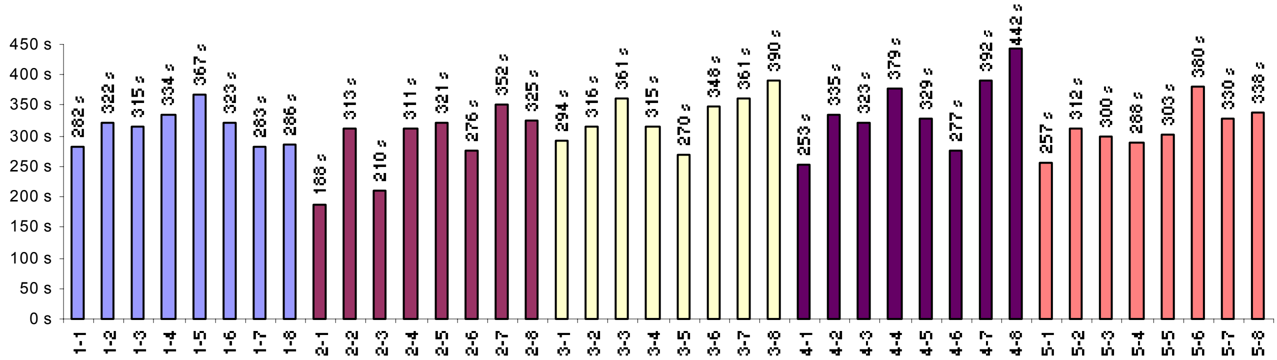


Figure 174: Mean task time per user

Figure 175 shows box plots for every user and the quartile data for his twelve tasks. Several users took only a couple of seconds for their fastest question. (Examples are user 5-7 with 8 seconds and user 4-3 with 12 seconds). Most users had at least one extended fact-finding question where they used the maximum time of 600 seconds. User 2-3 is remarkable in that he/she never took longer than 331 seconds for a single question. User 4-8 worked at least 230 seconds on each question, even took his fastest.

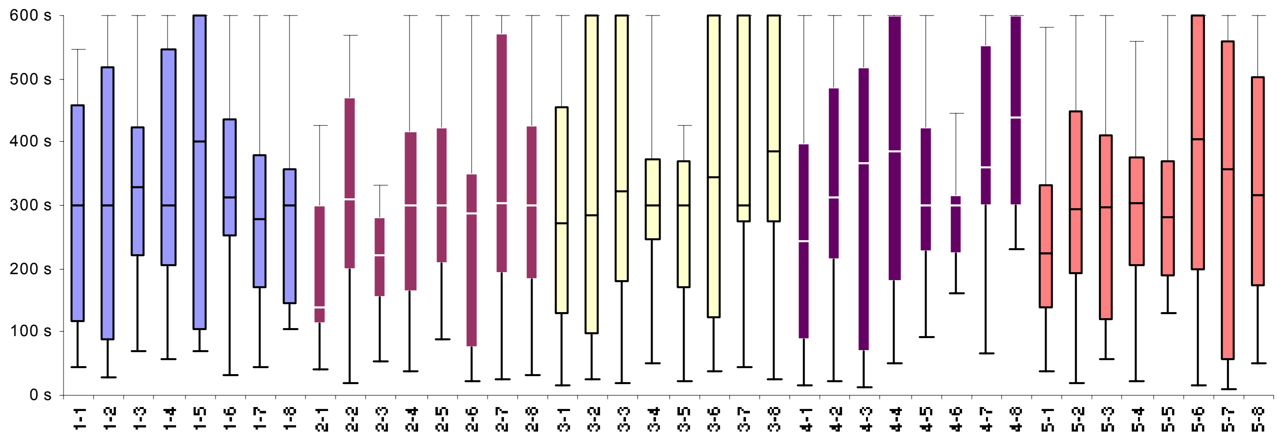


Figure 175: Box plot task time per user (min / 25%-quartile / median / 75%-quartile / max)

Turning now from task time to effectiveness, Figure 176 shows the mean effectiveness per user. The three most effective users, 5-2, 5-6, and 3-3, had average values of 80% and higher over all twelve questions. The least effective user was 4-8, who had an average of 33%. As may be remembered from the task times, user 4-8 was not only the least effective but also the slowest of all users tested. Accordingly, the values collected from this user are potential outliers. (Figure 178 on page 195 will show this graphically.)

[172] Six tasks had a limit of five minutes each and six tasks a limit of ten minutes: (6 * 300 s + 6 * 600 s)/12 = 450 s

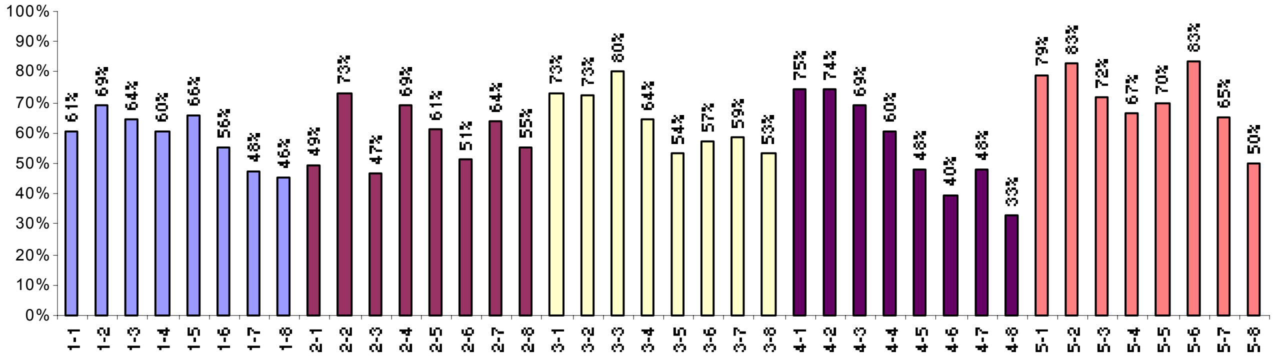


Figure 176: Mean effectiveness per user

The box plot in Figure 177 reveals that most of the users had at least one question that they were unable to answer, providing an effectiveness of 0%. Only the four users 3-3, 5-1, 5-2, and 5-6 solved all twelve tasks with, at least, some minimum success.

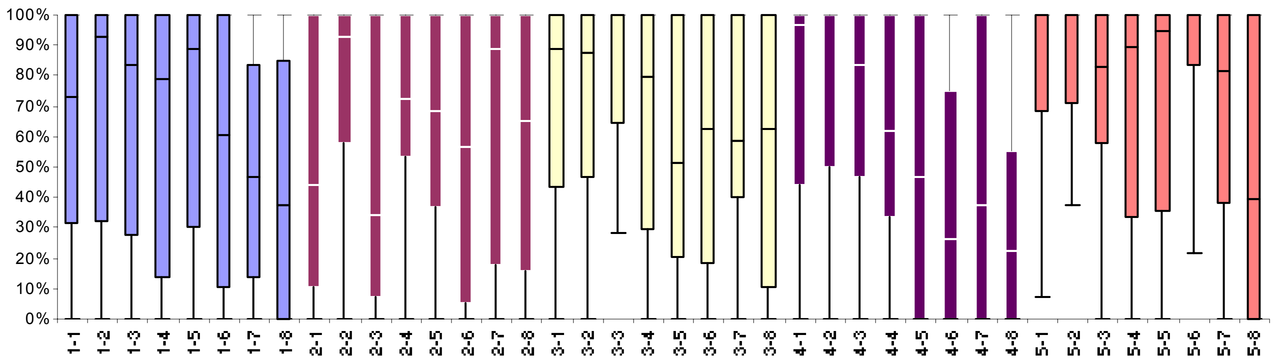


Figure 177: Box plot effectiveness per user (min / 25%-quartile / median / 75%-quartile / max)

When task time and effectiveness are combined as in Figure 178, the collected data does not clearly show a normal distribution among the users when taking the values directly and not indexing the data. Coding the status of the users in addition fails to produce a clear overall trend, such as the experts performed better than the beginners. Indeed, there are several experts among the users with poor effectiveness / task time ratios. User 4-8, who had the lowest mean effectiveness and the highest mean task time, was an expert.

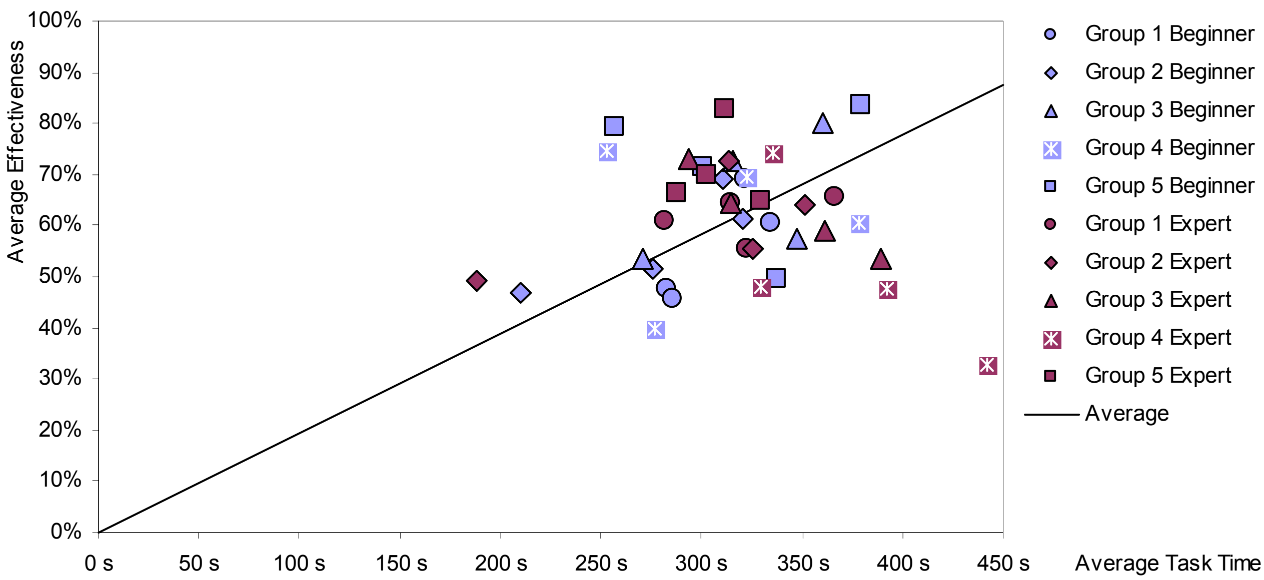


Figure 178: Average effectiveness and task time per user

Figure 179 shows the results of efficiency values by dividing mean effectiveness by mean task time in minutes. According to this calculation group 1 is the most homogenous group, while group 4 is the least. Please note that the overall efficiency values are quite different when calculated separately for each question and then averaged, rather than when calculated after the task time and effectiveness over all questions. The latter method is used in Figure 179 and in all following descriptions and figures. It is more robust against outliers in single questions than the former method.

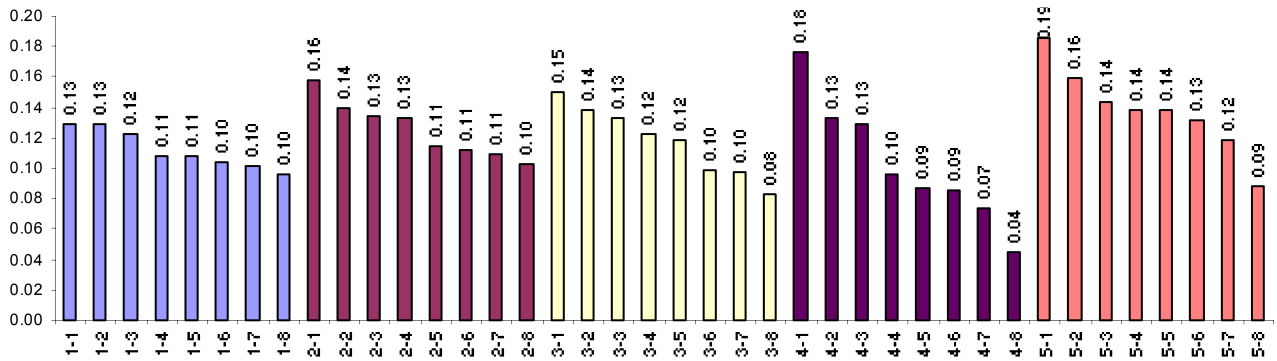


Figure 179: Mean efficiency per user

The more detailed analysis of the values in the box plot in Figure 180 reveals that there are several extreme efficiency values for some users and tasks. See, for example, the maximum value of 7.50 for user 5-7. This value occurred because the user solved one task with 100% effectiveness in 8 seconds[173].

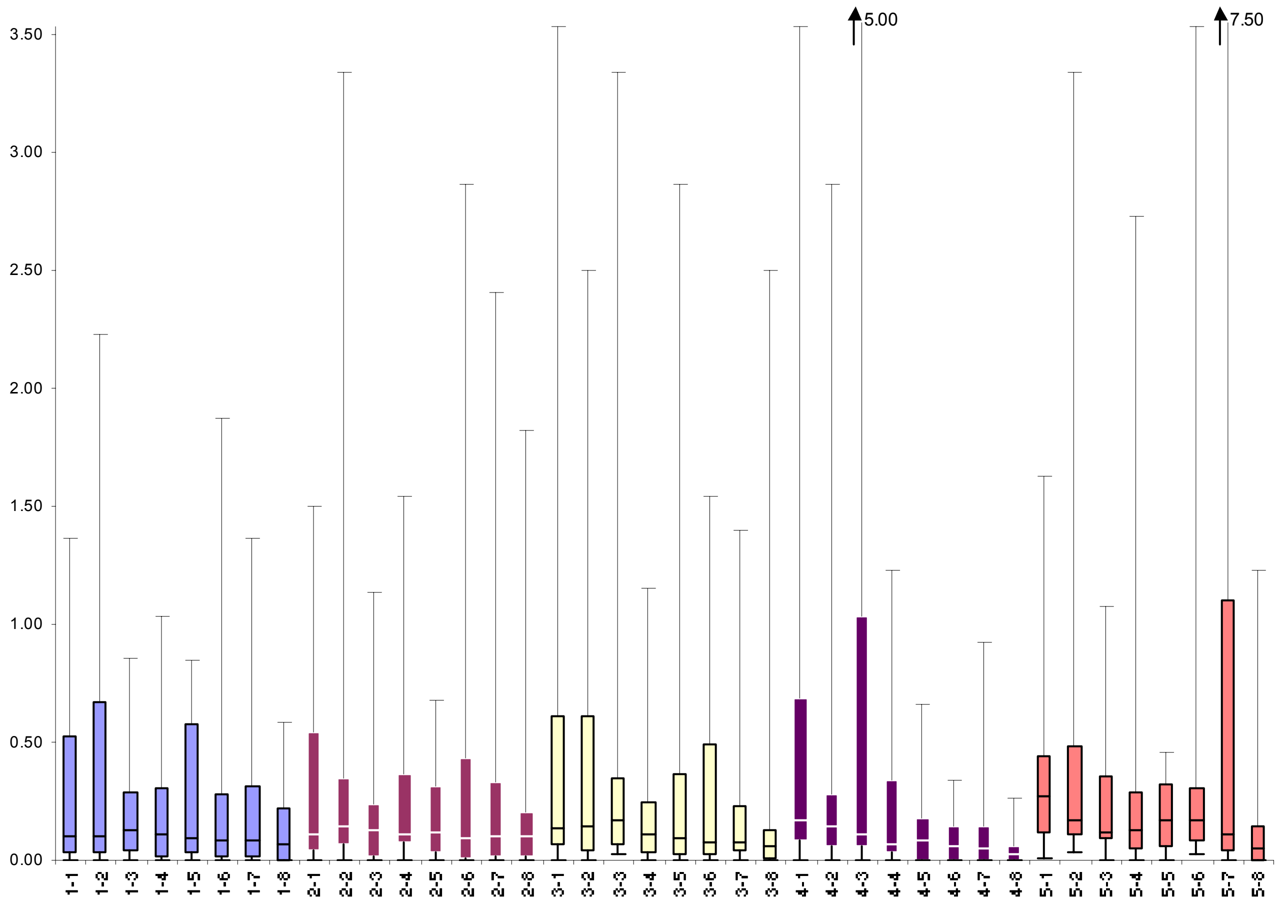


Figure 180: Box plot efficiency per user (min / 25%-quartile / median / 75%-quartile / max)

In sum, it can be stated that there was a large variation among users.

[173] 100% / 0.1333333333 min = 7.50

4.3.6.3.2. Task time, effectiveness, and efficiency per question

In all subsequent figures, questions are coded as follows: "type of task"-"number of hits"-"number of keywords" topic. To give two examples: "ex-30-1 platon" is the extended fact-finding question about Plato, using one keyword and presenting 30 hits, "sp-500-8 exxon valdez" is the specific fact-finding question about the Exxon Valdez, using eight keywords and presenting 500 hits. The questions are in most cases presented ordered by type of task, number of hits, and number of keywords so as to enhance the readability of the figures. In the discussion of learning effects, the questions will be presented in the order in which the users performed the tasks. See Table 33 on page 167 and Table 34 on page 167 for details about the tasks.

Examination of the task times per question instead of per user Figure 181 shows that the extended fact-finding tasks have much longer mean task times than the specific fact-finding tasks. It is surprising that the "Danube"-question has the longest average task time of all specific fact-finding tasks in view of the fact that the result set of 30 documents was much smaller than the ones of three 500-hit questions that were also specific fact-finding tasks. The „Danube"-question's average of 270 seconds is very close to the theoretical upper limit of 300 seconds for specific fact-finding tasks. Very similar is the result for the "moon"-question.

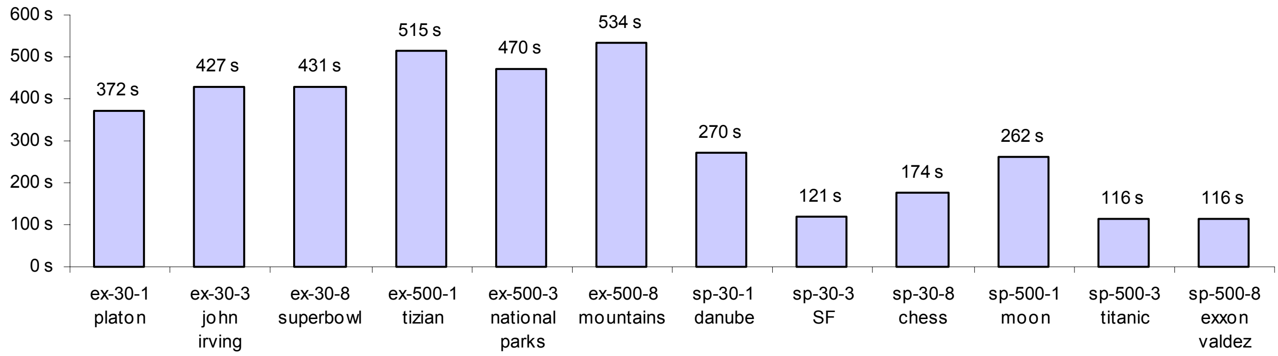


Figure 181: Mean task time per question

When studying the details in Figure 182, it becomes clear that for four of the twelve questions a ceiling effect may have as a result of allowing no more than 600 seconds in which to accomplish extended fact-finding tasks and 300 seconds to accomplish specific fact-finding tasks. Affected were the "Titian"-, the "mountains"-, the "Danube"-, and the "moon"-question. The medians for the four questions are the same as the upper limits.

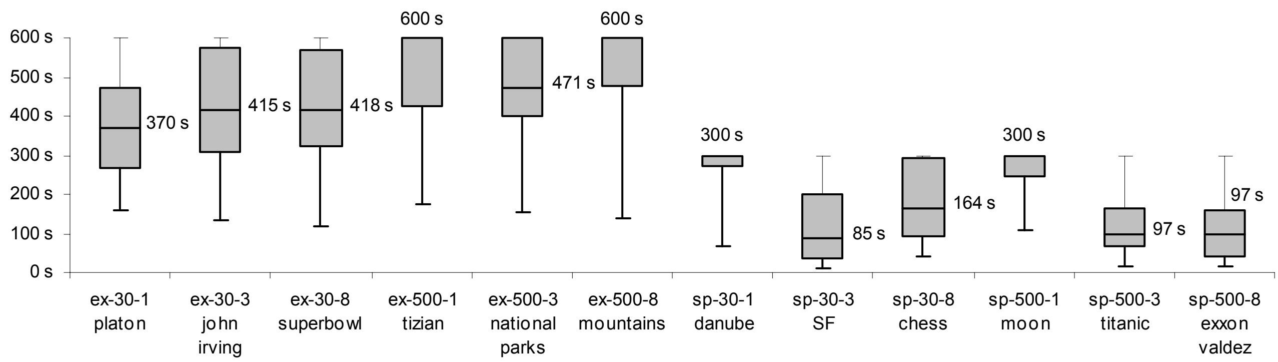


Figure 182: Box plot task time per question (min / 25%-quartile / median / 75%-quartile / max)

A graphical overview of users, questions, usage scenarios, and task times can be found in Figure 223 on page 259 in the appendix. It also provides the details behind the high mean and median task time values for the "Titian"-, the "mountains"-, the "Danube"-, and the "moon"-question.

Figure 183 illustrates that the four questions with the ceiling effect had the lowest mean effectiveness values. The ceiling effect itself should not be critical for the evaluation, because time restrictions are quite common in everyday life, and all components were tested under the same conditions. In general, specific fact-finding tasks show higher effectiveness values than the extended fact-finding tasks. A remarkable exception is the “Danube”-question, which had the lowest mean effectiveness of all tasks.

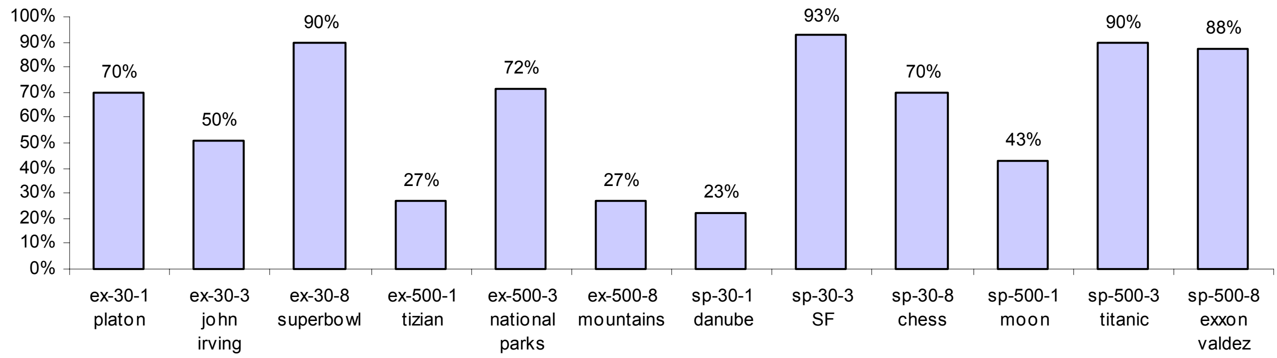


Figure 183: Mean effectiveness per question

Due to the binary nature of the specific fact-finding tasks, in which the answer is either found or not, box plots with quartiles make no sense in presenting the results for this type of question. For reasons of completeness they are nevertheless included in Figure 184. As regards the extended fact-finding tasks it can be seen that for two questions one user at least did not find any parts of the answer. These were the “Titian”- and the “mountains”-question, which were both subject to the ceiling-effect.

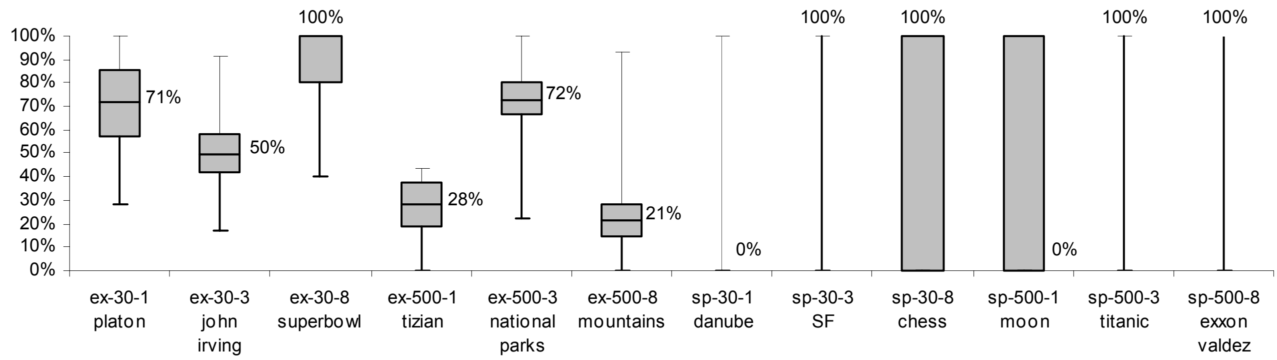


Figure 184: Box plot effectiveness per question (min / 25%-quartile / median / 75%-quartile / max)

An overview covering users, questions, usage scenarios and effectiveness can be found in Figure 224 on page 260 in the appendix. The low mean and median effectiveness values of the “Titian”-, the “mountains”-, the “Danube”- and the “moon”-question become apparent there with more details. For example, it is very interesting that the “Danube”-question was not solved by any user with the HTML-List but by 50% of the users with the SegmentView plus ResultTable. This phenomenon will be discussed in more detail in Chapter 4.3.6.3.5 beginning at page 208.

Figure 185 shows the effectiveness and task time per question in a scatterplot. As for the users, there is definite no normal distribution of the values. As expected, the four questions with the ceiling effect show the worst values. It is noteworthy, that there are three specific fact-finding questions with 30 or 500 hits and three or eight keywords that show very similar average effectiveness and task time values: “Titanic”, “Exxon Valdez”, and “SF”. The same phenomenon can be seen with the “Tizian”- and “mountains”- question among the extended fact-finding tasks.

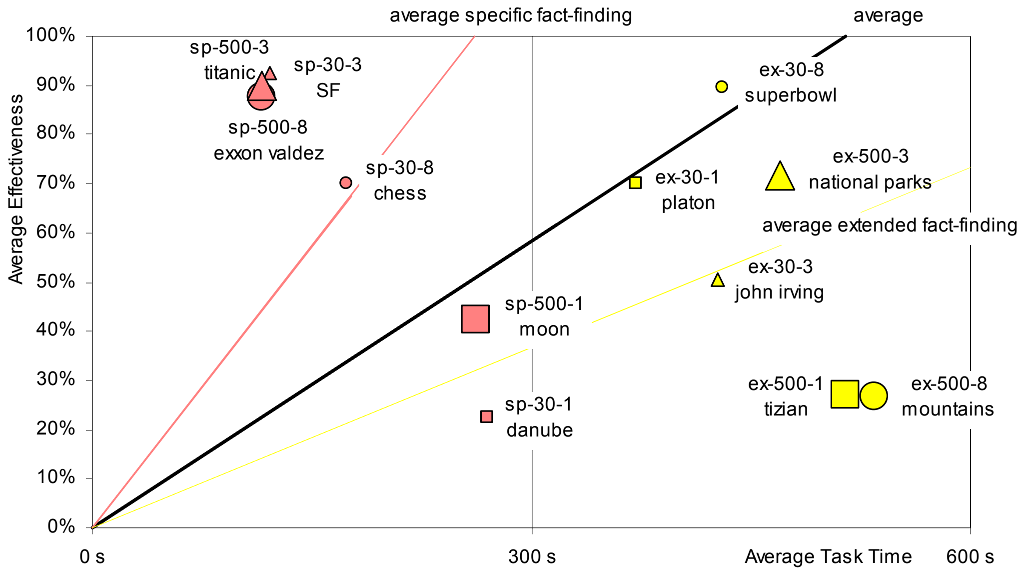


Figure 185: Average effectiveness and task time per question

Figure 186 shows the same similarities between the questions as Figure 185 but in another form, namely as calculated mean efficiency values. For the calculation of the overall average efficiency, the same principle is used as for the calculation of the average efficiency per user. Despite some similarities in the mean efficiency values, the range of efficiency values differ considerably for the different questions. Details are shown in the box plot Figure 220 on page 257 in the appendix.

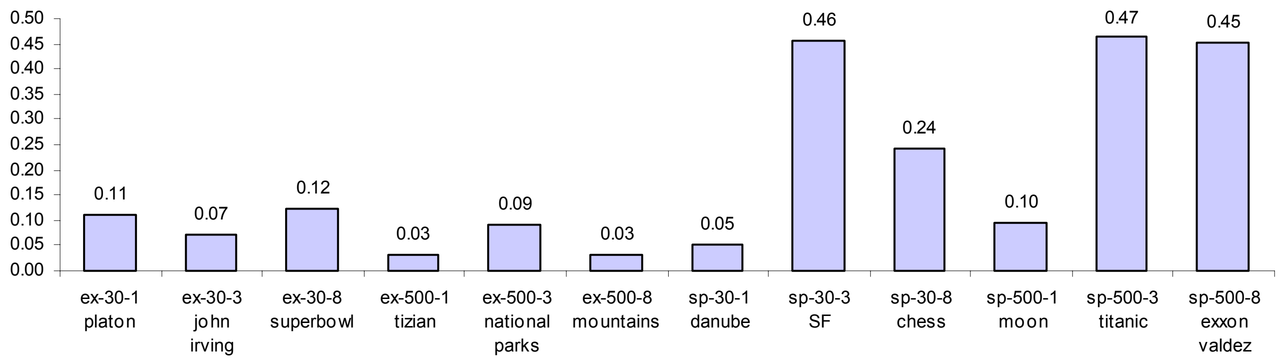


Figure 186: Mean efficiency per task

In sum, it can be stated that there was a large variation among topics. Similar effects have been in other evaluations such as [Golovchinsky 1997].

4.3.6.3.3. Task time, effectiveness, and efficiency per user interface condition

Turning now to the hypotheses, the general usability of the five different visualizations first will be examined. As regards the hard facts, the hypothesis H1b was that the ResultTable and the Visualization plus ResultTable conditions produce results in terms of effectiveness that differ from the results for the HTML-List.

In earlier publications [Mann, Reiterer 2000], [Reiterer, Mußler, Mann 2001], we reported preliminary mean effectiveness values for the five different user interface conditions. Figure 187 shows the final values. The HTML-List and the ResultTable had the highest average effectiveness values, both at around 64%. The mean effectiveness values for the ScatterPlot and the SegmentView conditions are at 62% and 61% respectively not far behind those for the HTML-List and the ResultTable. The BarGraph condition had at 57% the worst average effectiveness.

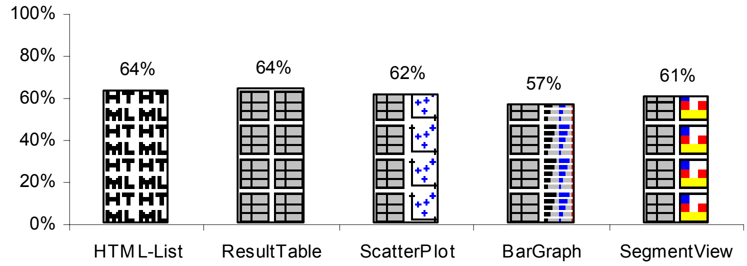


Figure 187: Mean effectiveness per user interface condition

The final effectiveness values shown above differ a little bit from the preliminary values reported earlier. In general, the final values are 1.7 percentage points lower. To be precise: 1.6 points lower for the HTML-List, 1.0 for the ResultTable, 1.6 for the ScatterPlot, 2.7 for the BarGraph, and 1.6 for the SegmentView. A careful reexamination of the evaluation results revealed that in several cases the effectiveness values had to be adjusted. The main reason was that the test subjects found some correct answers for the extended fact-finding tasks that had not been found by the evaluation preparation team in the overall set of 3,180 documents. Accordingly, the number of possible correct answers was higher than initially calculated, leading to a lower average effectiveness. The additional correct answers found by the subjects were not distributed equally among components. The results for the components were therefore not influenced equally. In addition, some results had been rated incorrect. An example was the question as to which mountains are higher than 8000 meters. One of the correct answers was “Gasherbrum I”. Four users did not write down “Gasherbrum I” but instead “Hidden Peak”, which is another name for “Gasherbrum I”. The experimenters marked this answer as wrong. When preparing the test, the experimenters knew that there are exactly 14 mountains higher than 8000 meters. They had therefore stopped looking for additional mountains in the 500-document result set after identifying the 14 names. The alternative name was found when checking the results for plausibility to prepare this thesis and reexamining the result sets for additional answers given by the users.

An additional potential problem detected in this process concerns the “chess”-question: who lost the second game in the chess match between Gary Kasparov and Deep Blue in 1997. The question asked who lost the second game. 28 users answered correctly “Kasparov”. Five users gave no answer. The remaining seven users answered “Deep Blue”. Two of them wrote “Deep Blue won”[174], which is equivalent to “Kasparov lost”. It may - though it need not to be - that the other “Deep Blue”-users had identified the correct document or document part, but that they accidentally answered the question in the opposite way from how it was asked. Because we did not record the documents used to find the answers, this matter cannot be clarified. Affected was in one case the ResultTable condition, in two cases the BarGraph condition, and in two cases the SegmentView condition. Assuming that the five subjects had meant to write “Deep Blue won / Kasparov lost” the overall mean effectiveness for the ResultTable would rise by one percentage point and of the BarGraph and the SegmentView by two percentage points each.

Instead of speculating further, the hard facts should be returned to and a statistical validation for hypothesis H1b done. H1b was: the ResultTable and the Visualization plus ResultTable conditions produce results in terms of effectiveness that are different from the results for the HTML-List. According to Figure 187, the effectiveness values for the ResultTable and the HTML-List are nearly equal, while the values for the Visualization plus ResultTable conditions are lower.

[174] The original German answer of both users was “Deep Blue gewann”.

In order to check the hypothesis with inferential statistical methods, an effectiveness index per user interface condition was calculated for each test subject. This was done by calculating first the mean effectiveness for each task over all subjects[175]. Next, every subject was given a score for every task. The score was +1 when the subjects' effectiveness exceeded the average effectiveness for the task, -1 when the subject performed worse, and 0 when the subject equaled the average. In this way, an effectiveness index was calculated for every subject and user interface condition, which varied from -3 to +3. To give an example, let us take subject 1-1 and the HTML-List. User 1-1 used the HTML-List in task 1 (Danube), task 6 (mountains), and task 11 (Titanic). The mean effectiveness values for these tasks had been 23% (Danube), 27% (mountains), and 90% (Titanic). The user had the values 0% (Danube), 43% (mountains), and 100% (Titanic). Calculated scores are therefore –1, +1, +1 leading to an effectiveness index of +1 for user 1-1 and the HTML-List.

Figure 188 shows the mean effectiveness index values and the standard deviation. In contrast to the raw data, the index data showed a normal distribution. The ResultTable and the Visualization plus ResultTable conditions all showed more or less lower mean effectiveness index values than the HTML-List.

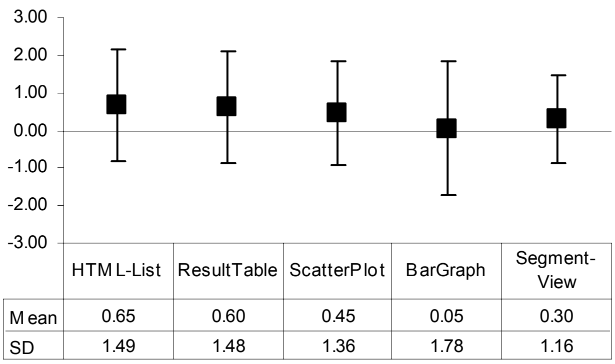


Figure 188: Mean effectiveness index and standard deviation

The differences of the effectiveness indexes of the HTML-List and the values from the other user interface conditions were calculated and tested using a t-test for dependent samples on the 5% significance level. Table 41 shows the t-values and the corresponding probability values.

Reference Condition	Tested Condition	t value	Pr > |t|
HTML-List	ResultTable	0.15	0.8824
HTML-List	ScatterPlot + ResultTable	0.80	0.4307
HTML-List	BarGraph + ResultTable	2.05	0.0471*
HTML-List	SegmentView + ResultTable	1.27	0.2128

Table 41: Two-tailed paired t-test: effectiveness indexes for the user interface conditions

* significant (p<.05)

The lower effectiveness in comparison with the HTML-List is only significant for the BarGraph condition. The differences for the effectiveness values of the other user interface conditions are not significant. H1b has therefore only been validated for the BarGraph condition. It produced results in terms of effectiveness that are significantly different from the results for the HTML-List.

The higher effectiveness values for the HTML-List could have occurred for several reasons. For example, the users could have spent more time working with the HTML-List. The left part of

[175] See Figure 183 on page 198 for the results.

Figure 189 shows the mean task times for all five different configurations. At 299 seconds, the HTML-List has the shortest average task time. Several different reasons could explain this. First, it is the most familiar representation for the users. The other components may have needed additional training times. Learning effects will be discussed in Chapter 4.3.6.3.4. Second, it could be that the HTML-List is the most efficient representation, allowing the shortest task times. Or third, working with the visualizations was so much fun that the users just played around longer with the other components.

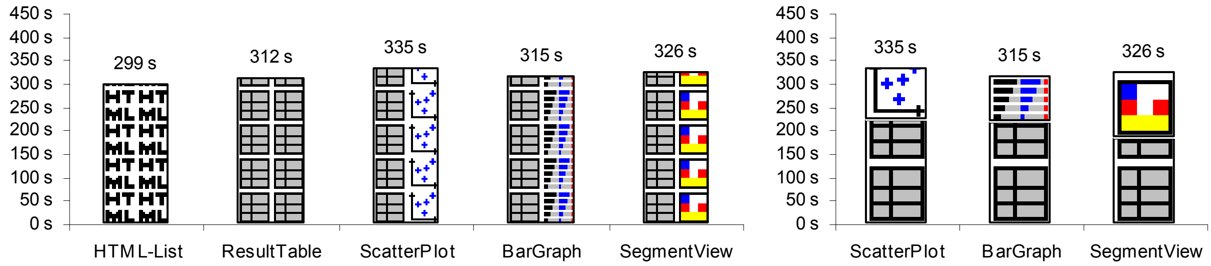
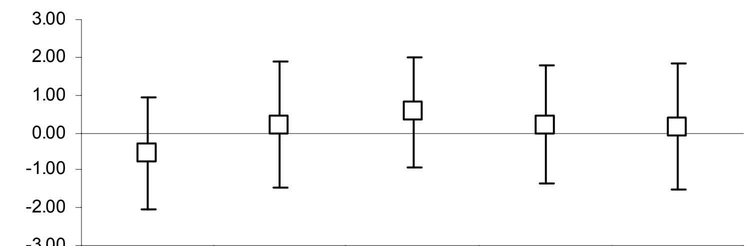


Figure 189: Mean task time per user interface condition and situation for the Visualization plus ResultTable conditions

Please remember that for the Visualization plus ResultTable conditions the subjects used the ResultTable and not the Visualization the majority of the time. The right part of Figure 189 shows the situation. The ScatterPlot was used 34% from the task time of the ScatterPlot condition, the BarGraph 30%, and the SegmentView 44%. See Chapter 4.3.6.1 on page 176 for details. The users who used only the ResultTable in these conditions will bias the results for the Visualizations. The ResultTable was used exclusively for 25% of the tasks in the ScatterPlot condition, 36% of the tasks in the BarGraph condition, and 23% of the tasks in the SegmentView condition. Theoretically the task time, effectiveness, and efficiency values could be calculated for each usage scenario. Practically this would make no sense, because the usage decisions are not equally distributed over the questions and users. The heterogeneity inside these dimensions is far too large to produce sense-making results.

Before we turn to the efficiency values and hypothesis H1c, a statistical examination of the task times will be done. As for to the effectiveness index a task time index was created and analyzed. Figure 190 shows the mean task time index values and the standard deviation. The ResultTable and the Visualization plus ResultTable conditions all showed generally higher mean task time index values than the HTML-List.



	HTML-List	ResultTable	ScatterPlot	BarGraph	SegmentView
Mean	-0.55	0.20	0.55	0.20	0.15
StdDev	1.5012815	1.6824890	1.4667249	1.5557997	1.6878145

Figure 190: Mean task time index and standard deviation

The results of the t-tests listed in Table 42 reveal that the HTML-List showed not only a significantly higher effectiveness than the BarGraph condition, but also significantly shorter task times. The task times for the HTML-List were significantly shorter than for all other user interface conditions on the 5% significance level. For the ScatterPlot condition the effect had been even highly significant on the 1% significance level.

Reference Condition	Tested Condition	t value	Pr > |t|
HTML-List	ResultTable	-2.28	0.0285*
HTML-List	ScatterPlot + ResultTable	-4.11	0.0002**
HTML-List	BarGraph + ResultTable	-2.49	0.0171*
HTML-List	SegmentView + ResultTable	-2.16	0.0373*

Table 42: Two-tailed paired t-test: task time indexes for the user interface conditions

* significant (p<.05) ** highly significant (p<.01)

We turn now to the efficiency values and hypothesis H1c: the ResultTable and the Visualization plus ResultTable conditions produce results in terms of temporal efficiency that are different from the results for the HTML-List. Figure 191 and gives a first impression of the mean efficiency values for the user interface conditions by visualizing the mean task times and the mean effectiveness values in a scatterplot.

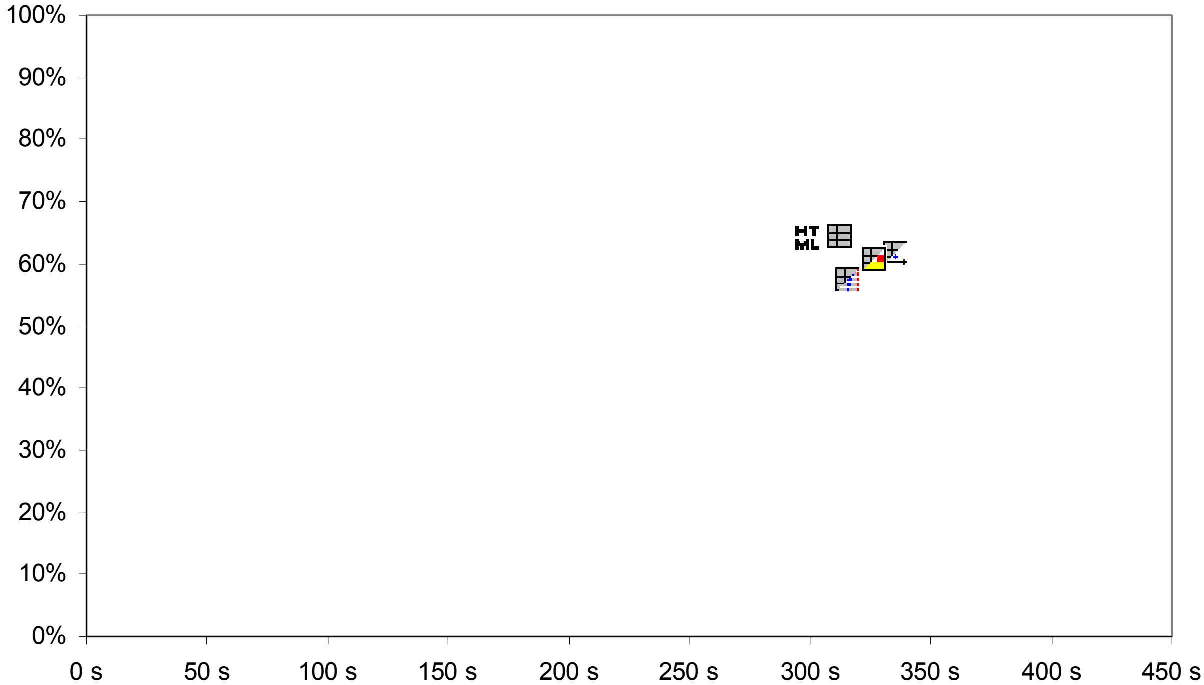
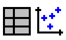
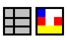


Figure 191: Mean effectiveness and mean task time per user interface condition

In the figure, , , and  represent the Visualization plus ResultTable conditions instead of , , and . Figure 192 shows the results for the different user interface conditions in the same dimensions as Figure 191 but uses the means of the indexed values instead of the means of the raw data.

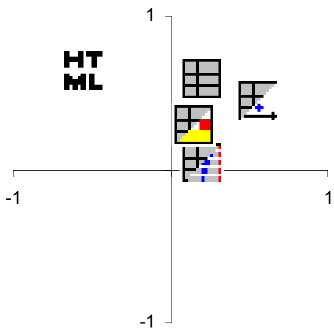


Figure 192: Mean effectiveness index (y-axis) and mean task time index (x-axis) per user interface condition

In Figure 8 of [Reiterer, Mußler, Mann 2001], efficiency values were presented that had been calculated from task times and effectiveness values similar to the ones shown in Figure 193[176]. The pattern there looks quite different from the one presented in Figure 191 or Figure 192. As described in Chapter 4.3.6.3.1, overall efficiency values differ considerably when calculated separately for each question and then averaged, rather then when calculated after averaging the task time and effectiveness over all questions. The pattern visible in the scatterplot of Figure 191 corresponds to the second method, the pattern visible in the bargraph of Figure 193 to the first method. As mentioned, the second method is more robust against outliers in single questions, than the first method. Figure 194 shows the patterns from the scatterplot of Figure 191 as bargraph. The robustness against single outliers becomes apparent when assuming that the most efficient of the 480 evaluation cells, where one user solved a task with 100% effectiveness in 8 seconds, would not have been solved by the user. In Figure 193, the value for the BarGraph would drop from 0.44 to 0.36 and in Figure 194 the value of 0.11 would remain the same.

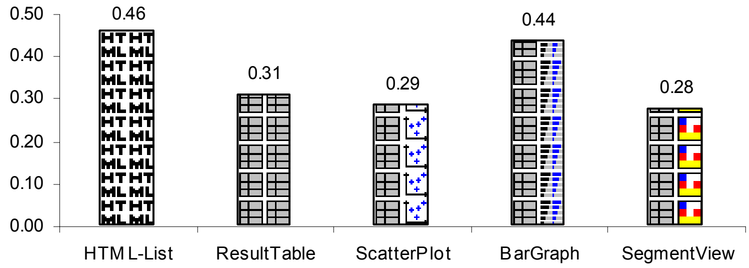


Figure 193: Mean efficiency per user interface condition (calculated by using efficiency per question)

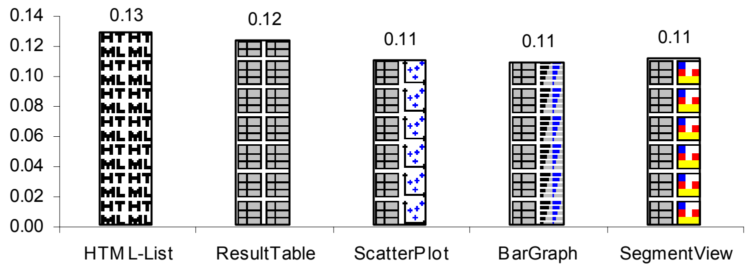


Figure 194: Mean efficiency per user interface condition (calculated by using average task time in minutes and average effectiveness)

More details can be found in the appendix, where Figure 221 on page 257 shows a box plot of the efficiency values for the different user interface conditions. The great variation in the values becomes even more apparent when studying Figure 222, showing the same values as Figure 221 but using the full scale, or Figure 226 on page 262, giving an overview covering users, questions, usage scenarios, and efficiency. In contrast to the scatterplot of Figure 191, the patterns visible in Figure 226 correspond to the patterns visible in Figure 193. The longest bar representing the ResultTable component in the BarGraph condition for user 5-7 and the "San Francisco"-question depicts the 100% effectiveness and 8 seconds task time result. The second longest bar in the same row for user 4-3 stand for 100% effectiveness and 12 seconds task time using the ResultTable component in the ScatterPlot condition. Figure 226 also reveals that the results from the extended fact-finding questions seem to be underrepresented when averaging the efficiency values for the single tasks in order to calculate the average efficiency of a user interface condition.

[176] In [Reiterer, Mußler, Mann 2001], we did not use task times in minutes with decimal fractions as in this thesis but other decimal time units. Accordingly the values are different, but the pattern is the same.

In Chapter 4.3.4.3 on page 169, temporal efficiency had been defined as effectiveness divided by task time. However, this definition of temporal efficiency is in practice difficult to apply to overview results when large between-topics and between-subjects variances are present. Another open methodical question is how to handle cases when, for example, in the “mountain”-question one user identified one mountain in one minute and then stopped, and another user identified nine mountains in ten minutes. Taking the numbers strictly, the second user is less efficient. It could have been the case, however, that it was very easy to find one answer in the result set, but much more difficult to find additional answers. Cases where subjects did not find any answer are also problematic. Dividing effectiveness by task time a subject who stopped searching after 30 seconds will have the same temporal efficiency as a subject who tried to find the answer for 10 minutes (zero in both cases).

In the analysis of the mean effectiveness and mean task times indexes, none of the user interface conditions showed higher mean effectiveness values than the HTML-List and all had significantly longer mean task time values. The temporal efficiency of the ResultTable and the Visualization plus ResultTable conditions may therefore not be better than that of the HTML-List.

Accordingly, it seems advisable to do a direct analysis of the temporal efficiency solely for tasks and subjects with comparable effectiveness values. Only in the cases of same positive effectiveness values can the efficiency be determined in an unproblematic way. Temporal efficiency is in these cases defined just by the reciprocal task time. The six specific fact-finding tasks are therefore candidates for temporal efficiency calculations. Every user who found the answer for such a task has the same effectiveness level of 100%. For the calculations the tasks were excluded, in which a remarkable number of users did not find the answers. Only the tasks were included in which the mean effectiveness for all users was above 75%. Figure 183 on page 198 shows that the “San Francisco”-, the “Titanic”- and the “Exxon Valdez”-question qualified. Table 43 - Table 45 show the mean task times and standard deviation per user interface condition for these tasks.

User Interface Condition	N	Mean	SD	Minimum	Maximum
HTML-List	8	81.50	71.39	18	213
ResultTable	8	135.13	104.34	19	298
ScatterPlot + ResultTable	6	153.50	113.57	12	300
BarGraph + ResultTable	8	72.75	77.72	8	199
SegmentView + ResultTable	7	101.86	93.33	27	300

Table 43: Task 5 (San Francisco), mean task times and standard deviation per user interface condition for all cases with 100% effectiveness

User Interface Condition	N	Mean	SD	Minimum	Maximum
HTML-List	8	40.25	29.04	17	105
ResultTable	6	113.33	59.58	27	187
ScatterPlot + ResultTable	8	132.38	90.34	17	291
BarGraph + ResultTable	6	96.33	71.17	29	223
SegmentView + ResultTable	7	93.71	47.51	33	173

Table 44: Task 9 (Exxon Valdez), mean task times and standard deviation per user interface condition for all cases with 100% effectiveness

User Interface Condition	N	Mean	SD	Minimum	Maximum
HTML-List	8	79.88	25.19	32	103
ResultTable	8	88.38	30.00	53	135
ScatterPlot + ResultTable	8	170.75	43.99	94	209
BarGraph + ResultTable	6	86.50	87.47	17	232
SegmentView + ResultTable	6	122.50	86.87	31	261

Table 45: Task 11 (Titanic), mean task times and standard deviation per user interface condition for all cases with 100% effectiveness

The values in Table 43 - Table 45 reveal that the task time and thus the temporal efficiency has a strong variance as a function of the user interface condition. In two of the three tasks, the HTML-List has the shortest mean task time and therefore the highest mean temporal efficiency. The BarGraph condition has a shorter mean task time than the HTML-List for the "San Francisco"-question alone. In order to ground these findings statistically, the task times were examined by using a univariate analysis of variance (one-way ANOVA). Table 46 contains the Degrees of Freedom (DF) of the model, the Type 1 Sums of Squares (SS), the Mean Square, the F Value, the probability value, and the coefficient of determination (R-Square), which indicates the percentage of variance explained.

Task	DF	Type I SS	Mean Square	F Value	Pr > F	R-Square
5 (San Francisco)	4	34070.56515	8517.64129	1.01	0.4189	0.111708
9 (Exxon Valdez)	4	37168.81548	9292.20387	2.33	0.0788	0.237012
11 (Titanic)	4	44775.75000	11193.93750	3.46	0.0189*	0.308690

Table 46: One-way ANOVA of the task times for tasks 5, 9, and 11

* significant (p<.05)

The differences between the user interface conditions remain under the level of significance in task 5 (San Francisco). Concerning Task 9 (Exxon Valdez) a trend may be spoken of (p<.1), and for task 11 (Titanic) the difference is significant on the 5% significance level. In order to test hypothesis H1c directly, it is necessary to carry out additional contrast calculations (multiple comparisons) apart from the ANOVA, which checks whether all mean values come from the same distribution. Contrast calculations compare the mean values in the task times for the individual user interface conditions among themselves (single comparisons). Because we are interested in differences between the HTML-List and the four other user interface conditions, four single comparisons using a Bonferroni analysis were done.

Contrast	DF	Contrast SS	Mean Square	F Value	Pr > F
HTML-LIST vs. ResultTable	1	11502.56250	11502.56250	1.36	0.2524
HTML-LIST vs. ScatterPlot + ResultTable	1	17773.71429	17773.71429	2.10	0.1571
HTML-LIST vs. BarGraph + ResultTable	1	306.25000	306.25000	0.04	0.8504
HTML-LIST vs. SegmentView + ResultTable	1	1547.14286	1547.14286	0.18	0.6719

Table 47: Task 5 (San Francisco), Bonferroni analysis user interface condition / task time

Contrast	DF	Contrast SS	Mean Square	F Value	Pr > F
HTML-LIST vs. ResultTable	1	18312.59524	18312.59524	4.59	0.0404*
HTML-LIST vs. ScatterPlot + ResultTable	1	33948.06250	33948.06250	8.51	0.0066**
HTML-LIST vs. BarGraph + ResultTable	1	10784.02381	10784.02381	2.70	0.1105
HTML-LIST vs. SegmentView + ResultTable	1	10671.47143	10671.47143	2.68	0.1123

Table 48: Task 9 (Exxon Valdez), Bonferroni analysis user interface condition / task time

* significant (p<.05) ** highly significant (p<.01)

Contrast	DF	Contrast SS	Mean Square	F Value	Pr > F
HTML-LIST vs. ResultTable	1	289.00000	289.00000	0.09	0.7670
HTML-LIST vs. ScatterPlot + ResultTable	1	33033.06250	33033.06250	10.21	0.0032**
HTML-LIST vs. BarGraph + ResultTable	1	150.48214	150.48214	0.05	0.8306
HTML-LIST vs. SegmentView + ResultTable	1	6229.33929	6229.33929	1.93	0.1751

Table 49: Task 11 (Titanic), Bonferroni analysis user interface condition / task time

** highly significant (p<.01)

There is no significant single comparison for Task 5 (San Francisco). In Task 9 (Exxon Valdez) the mean task time for the ResultTable and for the ScatterPlot plus ResultTable are significantly / highly significantly longer than the mean task time for the HTML-List. In addition, for Task 11 (Titanic) there is a significant single comparison: the mean task time for the ScatterPlot plus ResultTable is highly significantly longer than the mean task time for the HTML-List

In summary, we can state that hypothesis H1c is valid for some user interface conditions in two of the tasks. The temporal efficiency of at least one of the user interface conditions differs significantly from the HTML-List. In these cases, the HTML-List had a superior temporal efficiency. It must be noted, however, that only three of the twelve tasks were analyzed, and for these three tasks only subjects who solved the tasks had been taken into account.

4.3.6.3.4. Learning effects

So far the results per question have been presented in a logical order grouped by different independent variables. Learning effects have not been the focus of the evaluation. Nevertheless, learning effects could be an issue if they are present for the different user interface conditions. The between-topics and between-users variances make it difficult to detect learning patterns. When studying the mean effectiveness values per question, there is also no clear trend. Figure 195 shows the results for the questions in the order in which the users solved the tasks. Some interesting facts can be discovered. For example, for the "Danube"-question, which had to be solved as the first task, none of the users in the HTML-condition, but 50% of the users in the SegmentView condition, solved it. The figure shows increasing effectiveness starting at the 6th question, but the experimental setting does not allow us to determine whether this is a learning effect or arose merely by chance out of the ordering of the questions.

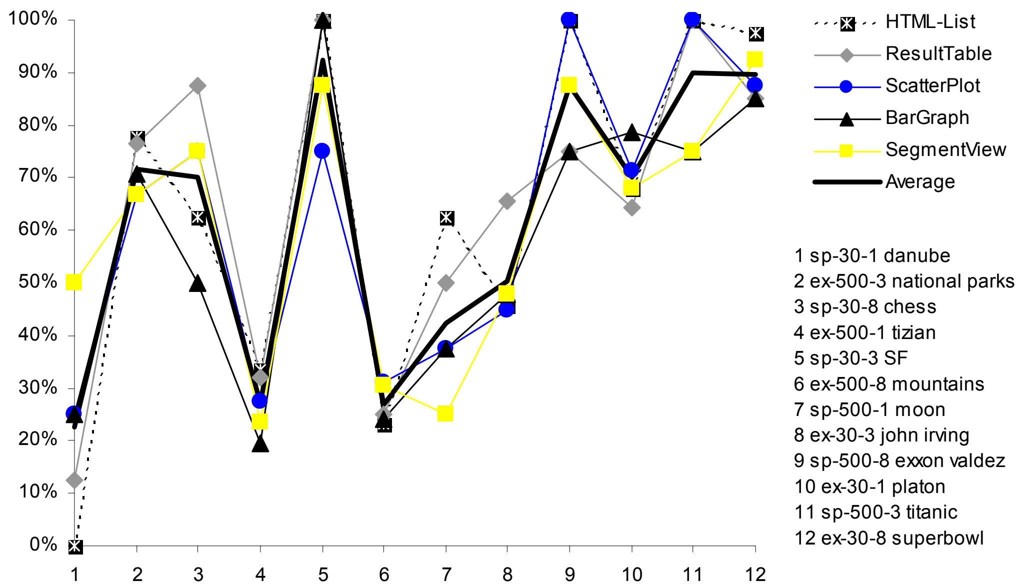


Figure 195: Mean effectiveness per user interface condition in the original sequence of the questions

Figure 196 shows the mean task times for all twelve questions in the order in which the users solved the tasks. To improve readability specific fact-finding and extended fact-finding questions are separated. In general, there is no clear trend for the overall average task time per question. There is also no clear pattern, when each user interface condition is compared with the overall average.

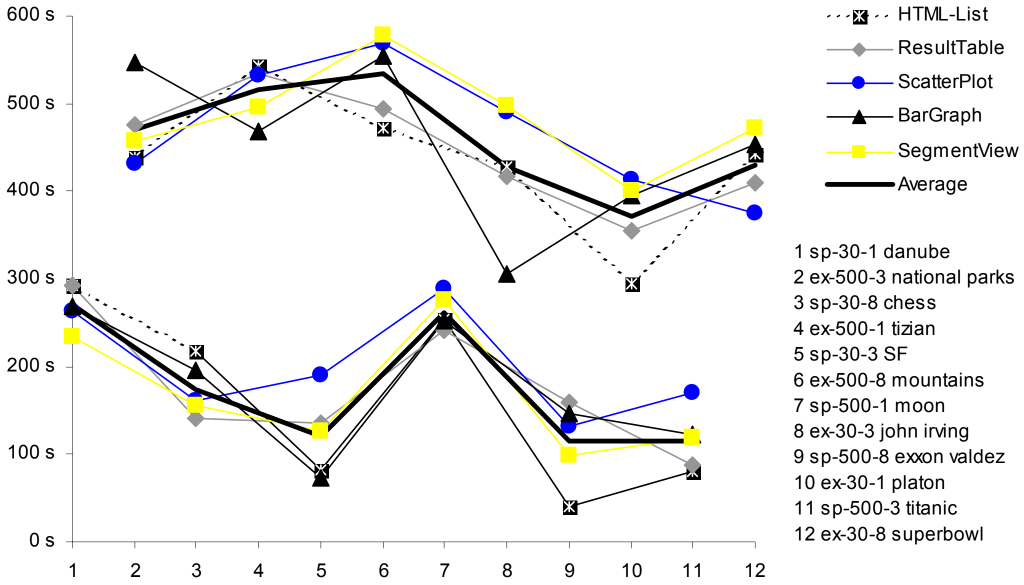


Figure 196: Mean task time per user interface condition in the original sequence of the questions

The test setting using the same twelve tasks in the same order for all users could theoretically influence the results, because the setting is not 100% randomized. On the basis of the figures shown in this chapter, it appears that the results of the test were not influenced.

4.3.6.3.5. The "Danube"-question

As mentioned above, the "Danube"-question showed very interesting results. On one hand it was the first task to be solved by the users after the warm-up exercise. It could be therefore assumed that the users were not very familiar with the type of task and the INSYDER system. On the other, the specific fact-finding condition and only 30 hits in the result set should allowed the task to be solved in the foreseen maximum of five minutes. Nevertheless, only nine of the 40 subjects or 23% succeeded in solving the task. The answer could be found in only one of the 30 documents of the result set[177]. It was on rank 22 with a relevance value of 72% for the keyword "danube". The title of the document was "Rectors Linked by the Danube – The Danube Rectors' Conference". The high average effectiveness of the SegmentView condition is very surprising, especially compared with that of the HTML-List, where none of the users found the answer. Figure 197 shows the relative average effectiveness of the user interface conditions per question. If all user interface conditions had the same mean effectiveness per question, the value for each condition should be 20%. The "Danube"-question can be recognized clearly as a special case.

---

[177] The information in the document reads "*Between its source in the Black Forest and its mouth on the Black Sea, the Danube traverses 1,170 miles* ...". The real length of the Danube river is about 1,770 miles. The distance as the crow flies from the Black Forest to the Black Sea may be somewhat farther than 1,170 miles. In other words, the answer that can be found in the document is in fact incorrect. Because it was an answer that could be found in the document set, it was treated as a correct answer. It could have happened that a subject knew the river's real length by chance, and that he therefore dismissed the answer 1,170 miles. It is extremely improbable that this actually happened, as none of the users made such a comment in the used think-aloud technique.

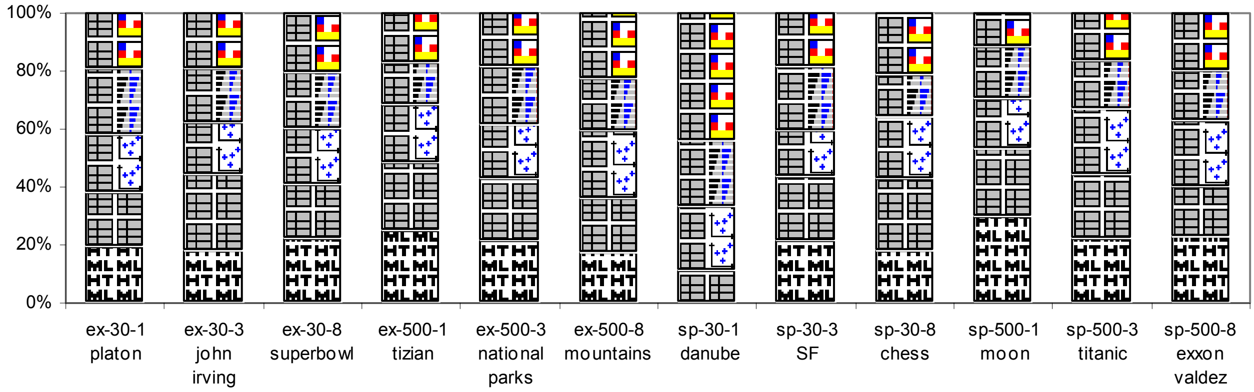


Figure 197: Relative mean effectiveness of the user interface conditions per question

To understand why the SegmentView condition was so effective compared to the other user interface conditions it is important to look at the views of the result set for the different components. The left part of Figure 136 on page 164 shows the BarGraph presentation of the "Danube" result set and the left part of Figure 139 on page 164 its ScatterPlot presentation. Figure 198 shows the relevant HTML-List and the ResultTable views. In the latter, the document with the answer is selected and displayed in the integrated browser. The left part of Figure 199 shows the SegmentView of the result set. Most of the documents are very short, and only one document exceeded 160 words. Not only was the result set small with 30 documents, but the documents in the result set were also relatively short. Even more surprising is that only nine from 40 users succeeded in solving the task in five minutes. The right part of Figure 199 shows that the 1,170 miles answer was shown in the tooltip of the target document. The tooltip was the same for ScatterPlot, BarGraph, and SegmentView. The 1,170 miles answer was also visible in the HTML-Result list, but required scrolling right. It was not visible, by just scrolling up and down the HTML-List.

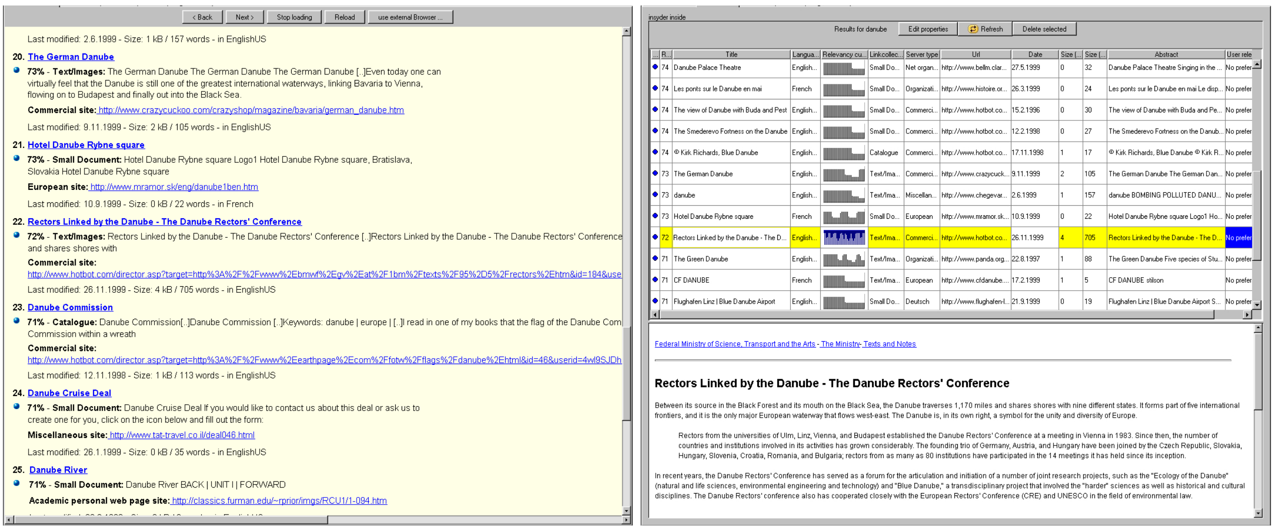


Figure 198: HTML-List and ResultTable of the "Danube"-question result set

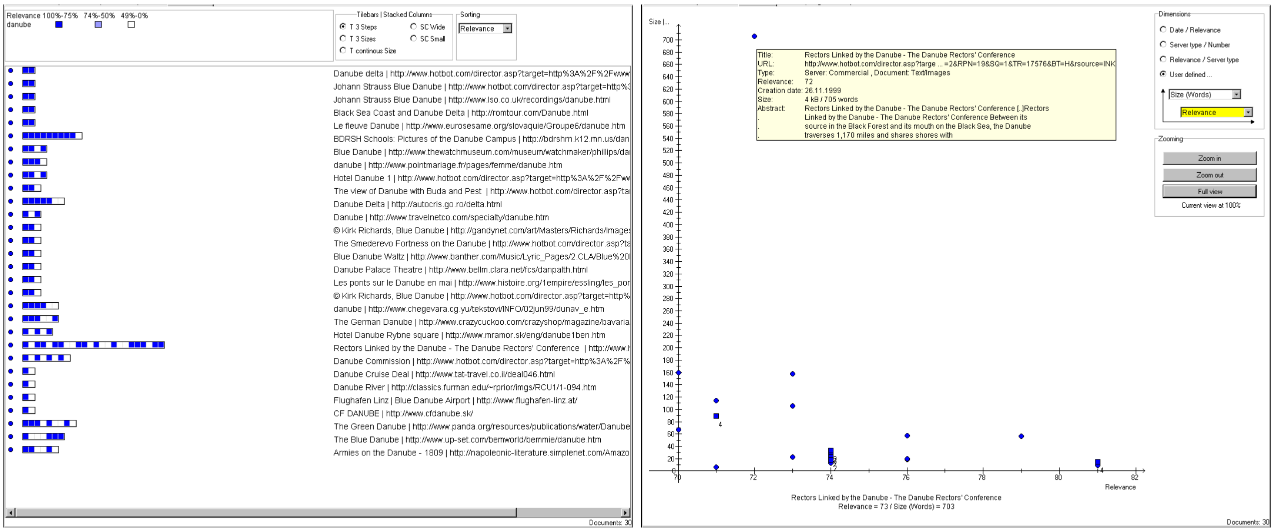


Figure 199: SegmentView and user configured ScatterPlot of the "Danube"-question result set

The SegmentView may have helped to examine the documents in detail and therefore been the reason for the subjects' high success rate in this user interface condition. On the other hand this example shows that the success of a user interface condition may also have been influenced by factors not controlled in the experiment. The target document was with more than 700 words by far the longest document in this result set. It was therefore very prominent in the SegmentView. This might have encouraged users to examine this document more carefully than the other documents. The salient size of the document may also have been visible in a user-configured ScatterPlot like the one in the right part of Figure 199. It shows the relevance on the x-axis and the size of the documents in words on the y-axis. The default ScatterPlot shown in the left part of Figure 139 on page 164 did not expose the document size. After this demonstration of how sensitive such test settings can be, we return now to the verification of the hypotheses.

4.3.6.3.6. Influence of the target user group

Hypothesis H2b had been: the target user group influences how the effectiveness will be determined by the user interface condition in comparison with the HTML-List. Like for the user satisfaction we are not interested in significant differences between experts and beginners[178] but in significant differences inside each group. For a statistical validation of H2b the effectiveness indexes were calculated in a way similar to that for hypothesis 1. Instead of comparing the effectiveness of a user for a certain task with the mean for this task, the comparison was done with the mean of the user group (beginners or experts) for this task. A similar method was used for all subsequent hypotheses concerning type of task, number of documents presented, and number of keywords used and shown.

Figure 200 shows the mean effectiveness index and standard deviation per user interface condition and user group. Inside the experts group, all user interface conditions show lower mean effectiveness index values than the HTML-List. This is consistent with the results from the analysis on the overall level. Inside the beginners group, most of the user interface conditions also performed worse than the HTML-List, except the ResultTable, which has a higher mean effectiveness index than the HTML-List.

---

[178] Such a comparison is included in the appendix on page 263.

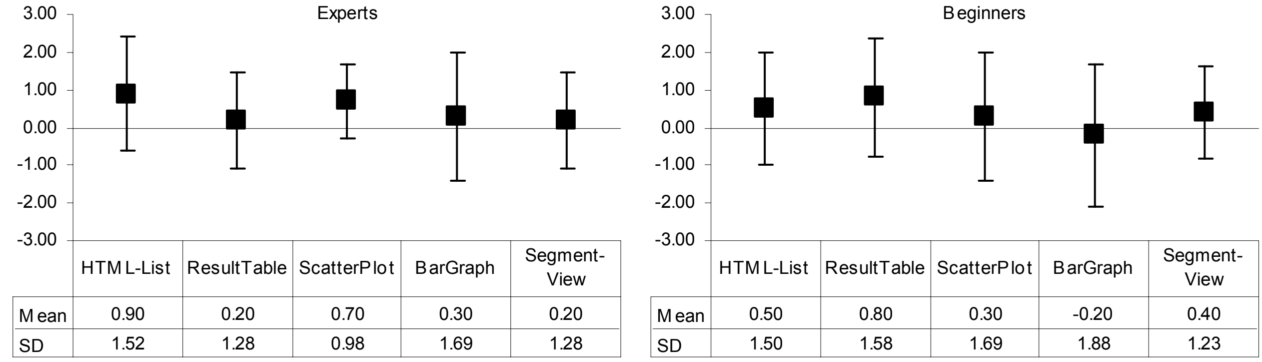


Figure 200: Mean effectiveness index and standard deviation per user group

Table 50 shows the t-values and the probability values of the corresponding t-tests for dependent samples. Despite the fact that experts and beginners show some differences in effectiveness patterns between the user interface conditions, inside the user groups, only one comparison shows significant differences on a 5% significance level. The effectiveness of the SegmentView condition for the experts is significantly lower than that of the HTML-List. Accordingly, hypothesis H2b was only validated for the SegmentView. Only for the SegmentView does the target user group influence how effectiveness will be determined by the user interface condition in comparison with the HTML-List.

Reference Condition	Tested Condition	Experts		Beginners	
		t value	Pr > |t|	t value	Pr > |t|
HTML-List	ResultTable	1.41	0.1759	-0.66	0.5163
HTML-List	ScatterPlot + ResultTable	0.66	0.5190	0.54	0.5922
HTML-List	BarGraph + ResultTable	1.45	0.1625	1.41	0.1759
HTML-List	SegmentView + ResultTable	2.10	0.0493*	0.23	0.8205

Table 50: Two-tailed paired t-test: effectiveness index per user group

* significant (p<.05)

As explained in Chapter 4.3.1, task time and efficiency were not included in hypotheses on the detailed level. The observed results will nevertheless be reported afterwards, including an inferential statistic analysis of the task times. Figure 201 shows the mean task time indexes and standard deviations per user interface condition and user group. As on the global level, in both groups the HTML-List shows the shortest mean task time indexes of all user interface conditions.

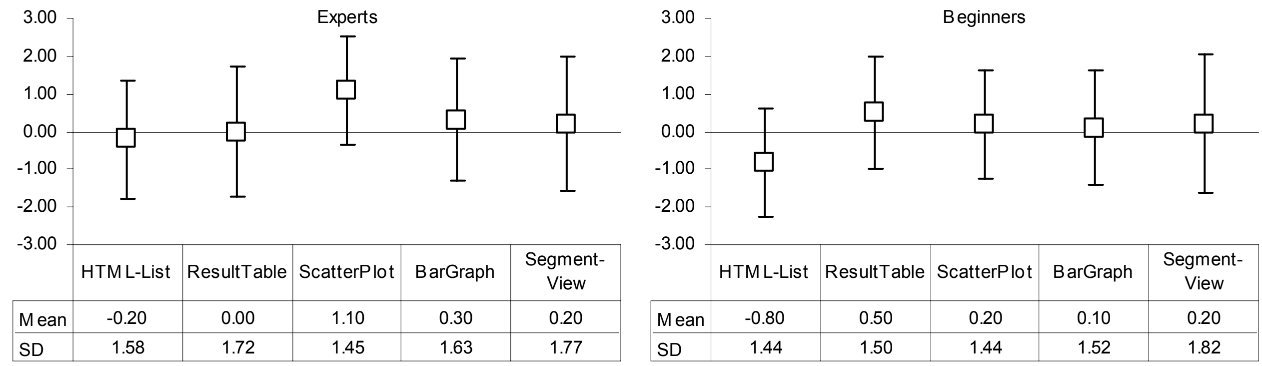


Figure 201: Mean task time index and standard deviation per user group

Table 51 shows the t-values and the probability values of the corresponding t-tests for dependent samples. For the experts the task times for the ScatterPlot condition are highly significantly longer than those for the HTML-List. For the beginners the task times for the ScatterPlot and the Bar-

Graph conditions are significantly longer than those for the HTML-List. The task times of the ResultTable are highly significantly longer than those for the HTML-List.

Reference Condition	Tested Condition	Experts		Beginners	
		t value	Pr > |t|	t value	Pr > |t|
HTML-List	ResultTable	-0.45	0.6581	-2.94	0.0084**
HTML-List	ScatterPlot + ResultTable	-3.58	0.0020**	-2.30	0.0331*
HTML-List	BarGraph + ResultTable	-1.07	0.2981	-2.44	0.0248*
HTML-List	SegmentView + ResultTable	-0.75	0.4639	-2.08	0.0515

Table 51: Two-tailed paired t-test: task time index per user group

* significant (p<.05) ** highly significant (p<.01)

Omitting a detailed analysis of the efficiency values for the reasons listed above, Figure 202 shows the results for the effectiveness and the task time in one visualization, giving some impressions concerning efficiency.

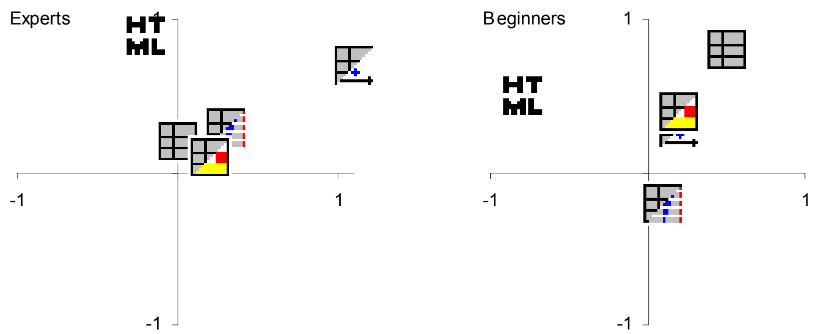
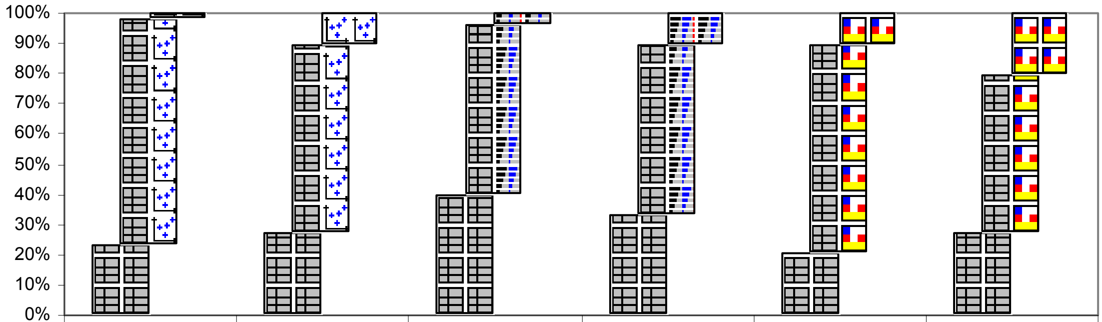


Figure 202: Mean effectiveness index and mean task time index per target user group

To summarize the facts:

- For the experts (left figure)
  - Highly significant difference of the effectiveness index between HTML-List and SegmentView condition (vertical axis)
  - Highly significant difference of the task time index between HTML-List and ScatterPlot condition (horizontal axis)
- For the beginners (right figure)
  - Highly significant difference of the task time index between HTML-List and ResultTable (horizontal axis)
  - Significant difference of the task time index between HTML-List and ScatterPlot condition (horizontal axis)
  - Significant difference of the task time index between HTML-List and BarGraph condition (horizontal axis)

Please remember that all results in the Visualization plus ResultTable conditions are biased by the usage decisions of the subjects. When studying Figure 203, it can be seen, for example, that the beginners in 40% of the cases in the BarGraph condition used only the ResultTable to solve the task. Unfortunately, our evaluation design does not allow us to compare the results of the usage scenarios of a user interface condition.

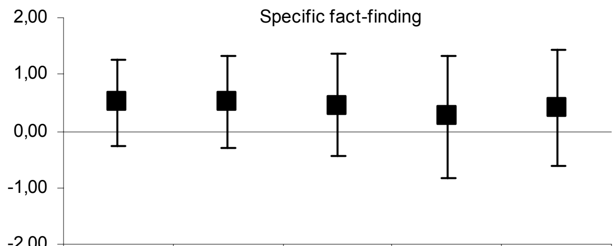


	ScatterPlot		BarGraph		SegmentView	
	Beginner	Expert	Beginner	Expert	Beginner	Expert
Visualization	1 tasks	5 tasks	2 tasks	5 tasks	5 tasks	10 tasks
Visualization + ResultTable	36 tasks	30 tasks	27 tasks	27 tasks	33 tasks	25 tasks
ResultTable	11 tasks	13 tasks	19 tasks	16 tasks	10 tasks	13 tasks

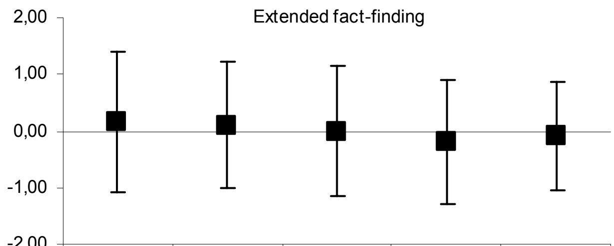
Figure 203: Usage of the components in the Visualization plus ResultTable conditions per user group

4.3.6.3.7. Influence of the task type

Hypothesis H3 had been: the task type influences how the effectiveness will be determined by the user interface condition in comparison with the HTML-List. Figure 204 shows the mean effectiveness index and standard deviation per user interface condition and task type for specific fact-finding and extended fact-finding tasks. In both cases, the HTML-List shows the highest mean effectiveness index. In the case of the specific fact-finding tasks, the ResultTable has the same high mean effectiveness index value as the HTML-List.



	HTML-List	ResultTable	ScatterPlot	BarGraph	Segment-View
Mean	0,50	0,50	0,45	0,25	0,40
SD	0,75	0,82	0,90	1,08	1,03



	HTML-List	ResultTable	ScatterPlot	BarGraph	Segment-View
Mean	0,15	0,10	0,00	-0,20	-0,10
SD	1,23	1,10	1,15	1,09	0,96

Figure 204: Mean effectiveness index and standard deviation per task type

Specific fact-finding and extended fact-finding tasks show quite similar effectiveness index patterns. None of the differences inside the task groups are significant, when tested using the t-test. Table 52 shows the results. Only for the extended fact-finding tasks is there a trend ($p<.1$) for the difference between HTML-List and BarGraph conditions. The difference between HTML-List and BarGraph condition for specific fact-finding tasks is in the same direction. On the overall level this difference was significant[179]. In general, hypothesis H3 must be rejected. The type of task did not show any influence on how the effectiveness would be determined by the user interface condition in comparison with the HTML-List.

[179] See Table 41 on page 201

Reference Condition	Tested Condition	Specific fact-finding		Extended fact-finding	
		t value	Pr > |t|	t value	Pr > |t|
HTML-List	ResultTable	0.00	1.0000	0.18	0.8578
HTML-List	ScatterPlot + ResultTable	0.27	0.7854	0.65	0.5196
HTML-List	BarGraph + ResultTable	1.26	0.2155	1.74	0.0897
HTML-List	SegmentView + ResultTable	0.57	0.5703	1.00	0.3235

Table 52: Two-tailed paired t-test: effectiveness index per task type

Figure 205 shows the mean task time indexes and standard deviations per user interface condition and task type for specific fact-finding and extended fact-finding tasks. In both cases, the HTML-List shows the lowest mean task time index.

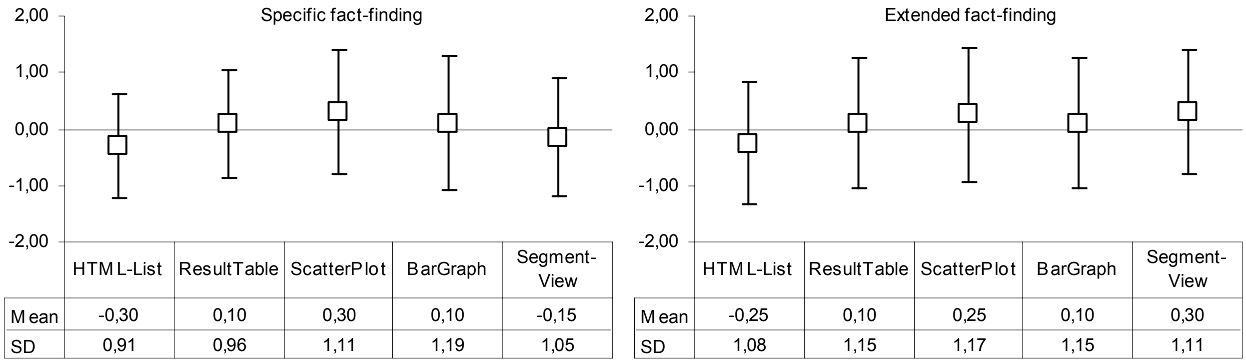


Figure 205: Mean task time index for extended fact-finding tasks and standard deviation

Table 53 shows the t-values and the probability values of the corresponding t-tests for dependent samples. For both types of tasks there is a highly significant difference in the task time index between HMTL-List and ScatterPlot condition. A significant difference can be found for extended fact-finding tasks between HTML-List and SegmentView.

Reference Condition	Tested Condition	Specific fact-finding		Extended fact-finding	
		t value	Pr > |t|	t value	Pr > |t|
HTML-List	ResultTable	-1.89	0.0657	-1.64	0.1092
HTML-List	ScatterPlot + ResultTable	-2.76	0.0087**	-2.91	0.0059**
HTML-List	BarGraph + ResultTable	-1.63	0.1105	-1.80	0.0799
HTML-List	SegmentView + ResultTable	-0.67	0.5092	-2.49	0.0170*

Table 53: Two-tailed paired t-test: task time index per task type

* significant (p<.05) ** highly significant (p<.01)

Figure 206 shows again the results for the effectiveness and the task time in one visualization, giving some impressions concerning efficiency.

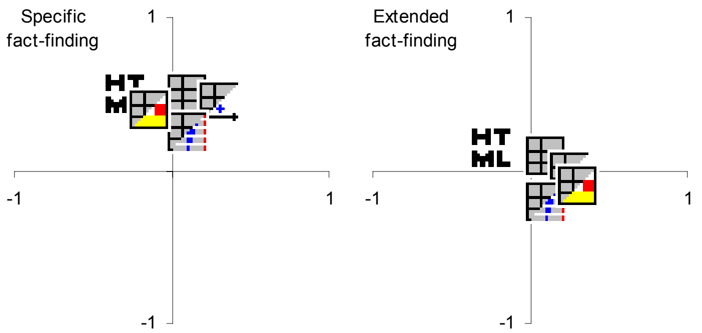
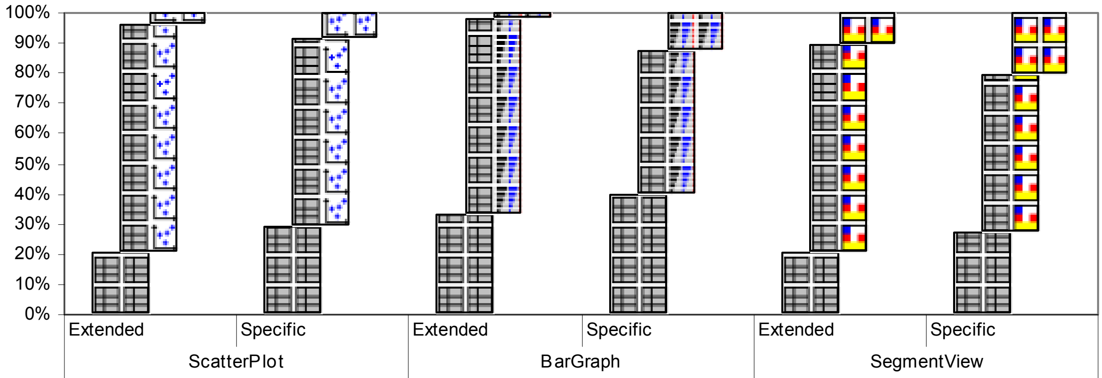


Figure 206: Mean effectiveness index and mean task time index per task type

To summarize the facts:

- For the specific fact-finding tasks (left figure)
  - Highly significant difference of the task time index between HTML-List and ScatterPlot condition (horizontal axis)
- For the extended fact-finding tasks (right figure)
  - Highly significant difference of the task time index between HTML-List and ScatterPlot condition (horizontal axis)
  - Significant difference of the task time index between HTML-List and SegmentView condition (horizontal axis)

The usage patterns in Figure 207 indicate that for specific fact-finding tasks the users more often used just one of the components in the Visualization plus ResultTable conditions, whereas for extended fact-finding tasks the users more often used both available components. When looking at the task times, this is not surprising. The longer the users take to solve a task, the more likely it becomes that they will use the second available component in a condition as well.

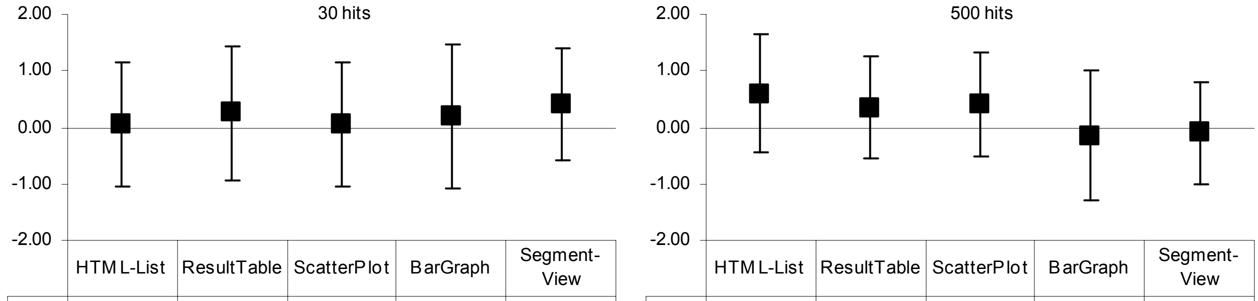


	ScatterPlot Extended	ScatterPlot Specific	BarGraph Extended	BarGraph Specific	SegmentView Extended	SegmentView Specific
Visualization	2 tasks	4 tasks	1 tasks	6 tasks	5 tasks	10 tasks
Visualization +ResultTable	36 tasks	30 tasks	31 tasks	23 tasks	33 tasks	25 tasks
ResultTable	10 tasks	14 tasks	16 tasks	19 tasks	10 tasks	13 tasks

Figure 207: Usage of the components in the Visualization plus ResultTable conditions per task type

4.3.6.3.8. *Influence of the size of the document set*

Hypothesis H4 was: the number of documents presented influences how the effectiveness will be determined by the user interface condition in comparison with the HTML-List. Figure 208 shows the mean effectiveness index and standard deviation per user interface condition and size of the document set for 30 hits and 500 hits tasks. In the case of the 30 hits tasks, all user interface conditions show the same or a higher mean effectiveness index than the HTML-List. The SegmentView condition has the highest mean effectiveness index value in this task group. In the case of the 500 hits tasks, all other user interface conditions show a lower mean effectiveness index than the HTML-List.



30 hits:

	HTML-List	ResultTable	ScatterPlot	BarGraph	Segment-View
Mean	0.05	0.25	0.05	0.20	0.40
SD	1.11	1.17	1.11	1.26	0.98

500 hits:

	HTML-List	ResultTable	ScatterPlot	BarGraph	Segment-View
Mean	0.60	0.35	0.40	-0.15	-0.10
SD	1.03	0.89	0.93	1.14	0.90

Figure 208: Mean effectiveness index and standard deviation per size of the document set

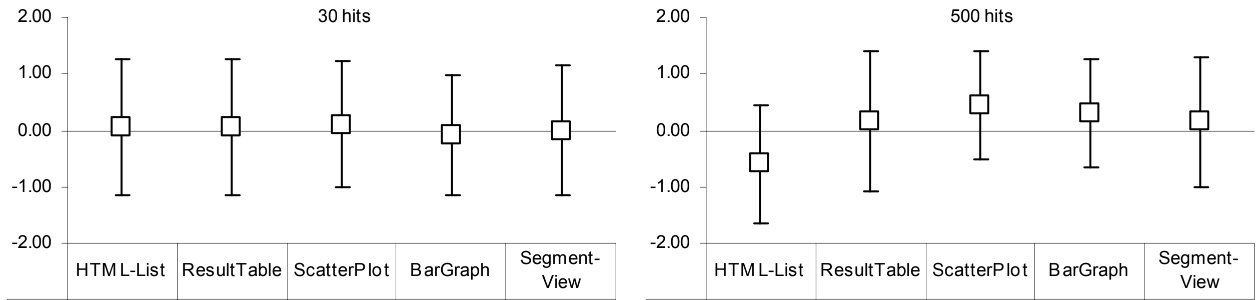
Table 54 shows the t-values and the probability values of the corresponding t-tests for dependent samples. For the 500 hits tasks the effectiveness of the BarGraph condition and the SegmentView condition is highly significantly lower than that of the HTML-List. All other constellations show no significant differences. Hypothesis H4 was therefore validated at least for the BarGraph and the SegmentView condition.

Reference Condition	Tested Condition	30 hits		500 hits	
		t value	Pr > |t|	t value	Pr > |t|
HTML-List	ResultTable	-0.75	0.4567	1.04	0.3030
HTML-List	ScatterPlot + ResultTable	0.00	1.0000	1.27	0.2099
HTML-List	BarGraph + ResultTable	-0.72	0.4738	3.20	0.0027**
HTML-List	SegmentView + ResultTable	-1.34	0.1894	3.34	0.0018**

Table 54: Two-tailed paired t-test: effectiveness index per size of the document set

** highly significant (p<.01)

Figure 209 shows the mean task time indexes and standard deviations per user interface condition and size of the document set for 30 hits and 500 hits tasks. Inside the 30 hits tasks group, differences are small. For the 500 hits tasks the HTML-List shows the lowest mean task time index of all user interface conditions.



30 hits:

	HTML-List	ResultTable	ScatterPlot	BarGraph	Segment-View
Mean	0.05	0.05	0.10	-0.10	0.00
SD	1.20	1.20	1.10	1.06	1.15

500 hits:

	HTML-List	ResultTable	ScatterPlot	BarGraph	Segment-View
Mean	-0.60	0.15	0.45	0.30	0.15
SD	1.03	1.23	0.96	0.97	1.14

Figure 209: Mean task time index and standard deviation per size of the document set

Table 55 shows the t-values and the probability values of the corresponding t-tests for dependent samples. Inside the 30 hits task group, there are no significant differences of the mean task times between the HTML-List and the other user interface conditions. Inside the 500 hits task group, all the differences between the HTML-List and the other user interface conditions are highly significant.

Reference Condition	Tested Condition	30 hits		500 hits	
		t value	Pr > |t|	t value	Pr > |t|
HTML-List	ResultTable	0.00	1.0000	-3.06	0.0040**
HTML-List	ScatterPlot + ResultTable	-0.25	0.8062	-5.55	<.0001**
HTML-List	BarGraph + ResultTable	0.65	0.5196	-3.80	0.0005**
HTML-List	SegmentView + ResultTable	0.19	0.8502	-3.06	0.0040**

Table 55: Two-tailed paired t-test: task time index per size of the document set

** highly significant (p<.01)

Figure 210 shows again the results for the effectiveness and the task time in one visualization, giving some impressions concerning efficiency.

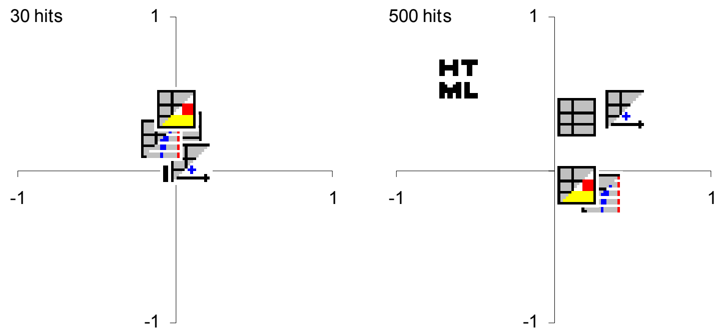


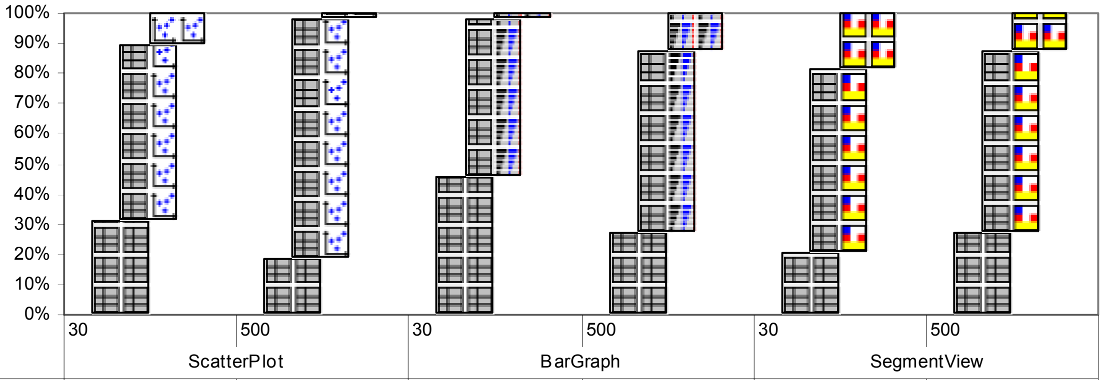
Figure 210: Mean effectiveness index and mean task time index per size of the document set

To summarize the facts:

- For the 30 hits tasks (left figure)
  - No significant differences of the effectiveness indexes or task time indexes
- For the 500 hits tasks (right figure)
  - Highly significant difference of the effectiveness index between HTML-List and BarGraph condition (vertical axis)
  - Highly significant difference of the effectiveness index between HTML-List and SegmentView condition (vertical axis)
  - Highly significant difference of the task time between HTML-List and all other user interface conditions (horizontal axis)

Differences among the user interface conditions seemed more pronounced for larger result sets.

Figure 211 shows that the users in the Visualization plus ResultTable conditions used both components more often when the results sets were larger. For the SegmentView condition the value is the same for both sizes of the result set.



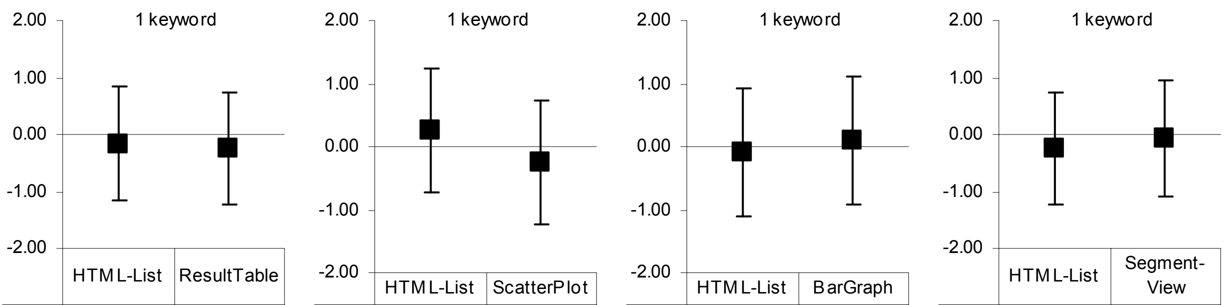
	ScatterPlot 30	ScatterPlot 500	BarGraph 30	BarGraph 500	SegmentView 30	SegmentView 500
Visualization	5 tasks	1 tasks	1 tasks	6 tasks	9 tasks	6 tasks
Visualization +ResultTable	28 tasks	38 tasks	25 tasks	29 tasks	29 tasks	29 tasks
ResultTable	15 tasks	9 tasks	22 tasks	13 tasks	10 tasks	13 tasks

Figure 211: Usage of the components in the Visualization plus ResultTable conditions per number of hits

4.3.6.3.9. Influence of the number of keywords

Hypothesis H5 was: the number of query keywords used and shown influences how the effectiveness will be determined by the user interface condition in comparison with the HTML-List. Figure 212 - Figure 214 show the mean effectiveness index and standard deviation per user interface condition and number of keywords for one, three, and eight keywords. The values from 40 users were available for within-subjects comparisons of the user interface conditions for the task type and the size of the document set. For the number of keywords it was only 24 users, as not all users could get all combinations for three different numbers of keywords. For every comparison between the user interface conditions therefore a subset of only 24 users was selected. This necessitates a separate calculation of the mean values and standard deviations for the adequate 24 users set of the HTML-List condition for each comparison.

In the case of the one-keyword tasks, BarGraph and SegmentView condition show higher mean effectiveness index values than the HTML-List; the ResultTable and the ScatterPlot condition show lower values. The three-keywords tasks show the same pattern, except for the SegmentView condition, whose mean effectiveness index is equal to that of the HTML-List. The eight-keywords tasks show the opposite pattern. The ResultTable and the ScatterPlot condition show higher mean effectiveness values than the HTML-List; the BarGraph and the SegmentView condition show lower values. This pattern means that for the eight-keywords tasks the visualizations that show the relevancies of each single keyword seem to drop down.



	HTML-List	ResultTable
Mean	-0.17	-0.25
SD	1.01	0.99

	HTML-List	ScatterPlot
Mean	0.25	-0.25
SD	0.99	0.99

	HTML-List	BarGraph
Mean	-0.08	0.08
SD	1.02	1.02

	HTML-List	Segment-View
Mean	-0.25	-0.08
SD	0.99	1.02

Figure 212: Mean effectiveness index for one-keyword tasks and standard deviation

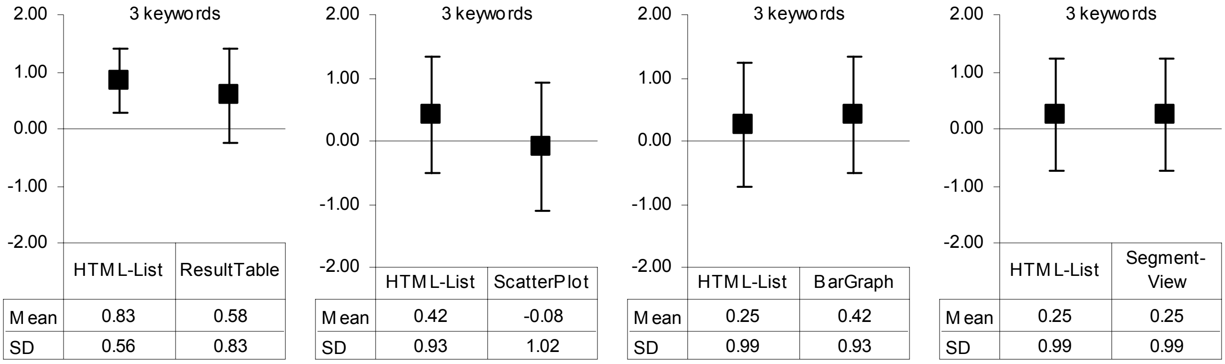


Figure 213: Mean effectiveness index for three-keywords tasks and standard deviation

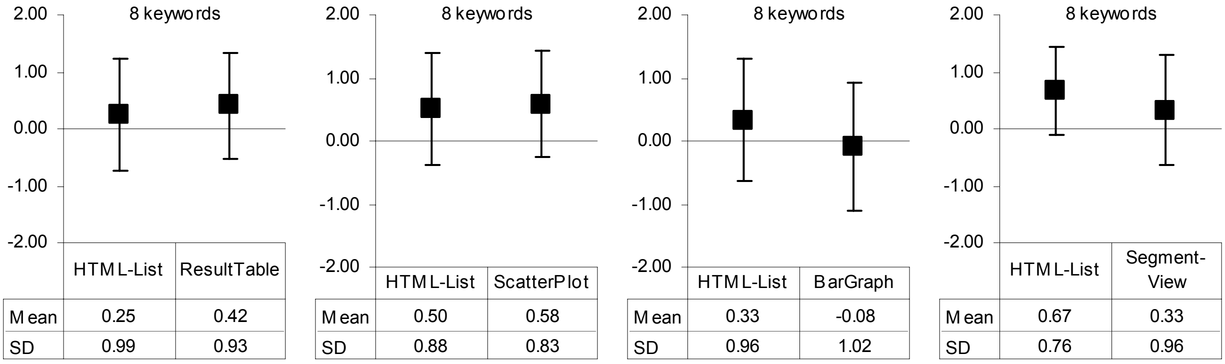


Figure 214: Mean effectiveness index for eight-keywords tasks and standard deviation

Table 56 shows the t-values and the probability values of the corresponding t-tests for dependent samples. For none of the comparisons are there significant differences of the effectiveness between the user interface conditions. Hypothesis H5 was therefore not validated. There may either be no significant differences or the size of the samples for this test was too small to get results on a significant level.

Reference Condition	Tested Condition	1 keyword		3 keywords		8 keywords	
		t value	Pr > |t|	t value	Pr > |t|	t value	Pr > |t|
HTML-List	ResultTable	0.30	0.7701	1.14	0.2656	-0.70	0.4912
HTML-List	ScatterPlot + ResultTable	1.81	0.0830	1.54	0.1366	-0.37	0.7140
HTML-List	BarGraph + ResultTable	-0.62	0.5385	-0.57	0.5748	1.55	0.1345
HTML-List	SegmentView + ResultTable	-0.62	0.5385	0.00	1.0000	1.45	0.1617

Table 56: Two-tailed paired t-test: effectiveness index per number of keywords

Figure 215 - Figure 217 show the mean task time indexes and standard deviations per user interface condition and number of keywords for one, three, and eight keywords. For most of the comparisons the HTML-List shows a lower mean task time index than the other user interface conditions. Exceptions can be found for the BarGraph, which has the same mean task time index as the HTML-List for the one-keyword tasks and a lower task time index for the three-keywords tasks. For the eight-keywords tasks the ResultTable has the same mean task time index as the HTML-List.

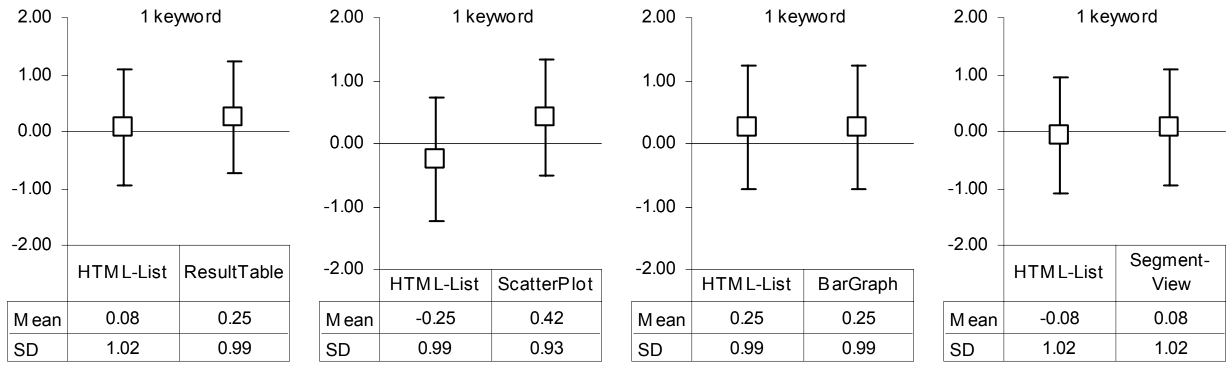


Figure 215: Mean task time index for one-keyword tasks and standard deviation

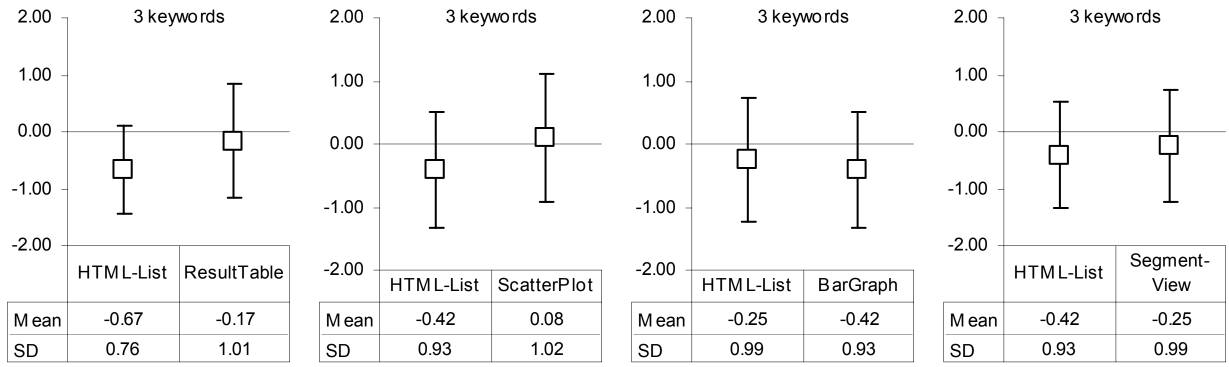


Figure 216: Mean task time index for three-keywords tasks and standard deviation

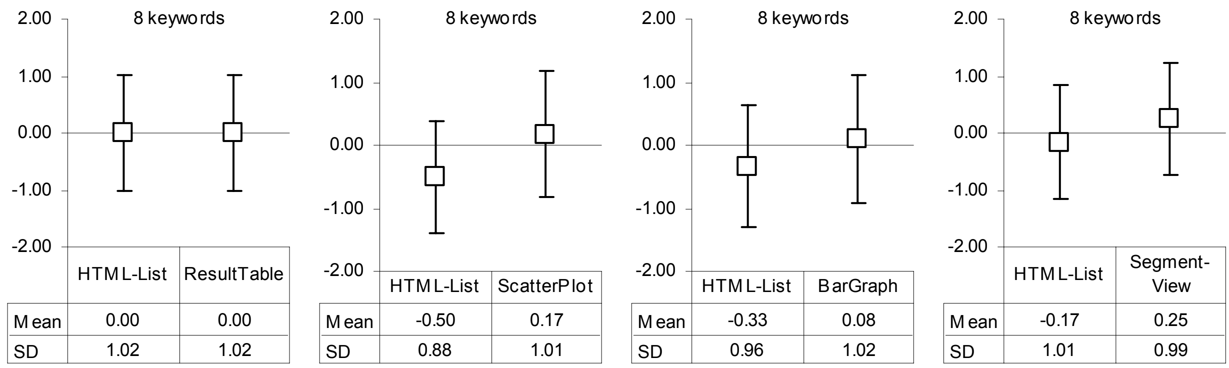


Figure 217: Mean task time index for eight-keywords tasks and standard deviation

Table 57 shows the t-values and the probability values of the corresponding t-tests for dependent samples. Inside the 3-keywords tasks group there are no significant differences of the mean task times between the HTML-List and the other user interface conditions. Inside the one-keyword and the eight-keywords task groups only the difference between the HTML-List and the ScatterPlot condition is significant. The missing significance of the differences of the task time could have been caused by the smaller size of the samples used to test the effect of the number of keywords.

Reference Condition	Tested Condition	1 keyword		3 keywords		8 keywords	
		t value	Pr > |t|	t value	Pr > |t|	t value	Pr > |t|
HTML-List	ResultTable	-0.57	0.5748	-1.81	0.0830	0.00	1.0000
HTML-List	ScatterPlot + ResultTable	-2.56	0.0174*	-1.66	0.1102	-2.14	0.0428*
HTML-List	BarGraph + ResultTable	0.00	1.0000	0.62	0.5385	-1.55	0.1345
HTML-List	SegmentView + ResultTable	-0.57	0.5748	-0.81	0.4259	-1.55	0.1345

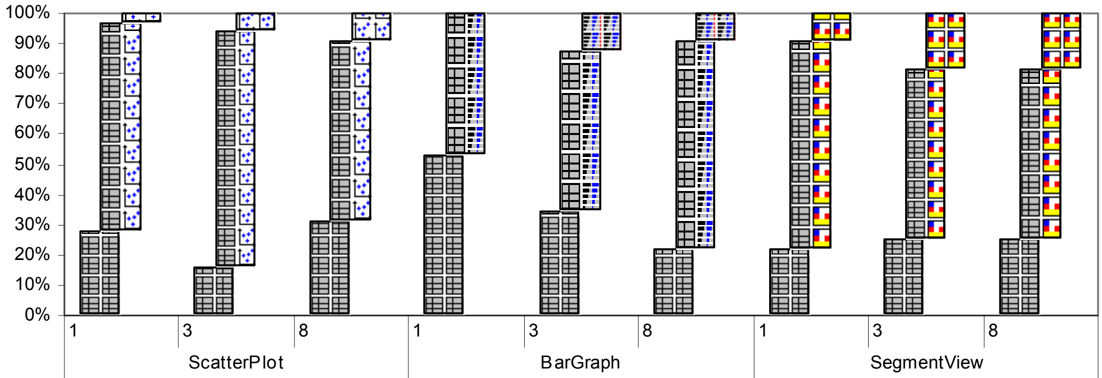
Table 57: Two-tailed paired t-test: task time index per number of keywords

* significant (p<.05)

Figures showing the results for the effectiveness and the task time in one visualization are omitted because the values cannot be combined in one graph per number of keywords condition. A summary of the facts can be performed anyhow:

- For the one-keyword tasks
    - Significant difference of the task time index between HTML-List and ScatterPlot condition
- For the three-keyword tasks
    - No significant differences of the effectiveness indexes or task time indexes
- For the eight-keywords task
    - Significant difference of the task time index between HTML-List and ScatterPlot condition

When looking at the usage patterns in Figure 218 one point stands out. None of the subjects used the BarGraph without the ResultTable to answer a one-keyword question. When studying Figure 136 on page 164 and Figure 137 on page 164 this is not astonishing. With a single keyword the insights provided by the BarGraph visualization are really questionable.



	ScatterPlot 1	ScatterPlot 3	ScatterPlot 8	BarGraph 1	BarGraph 3	BarGraph 8	SegmentView 1	SegmentView 3	SegmentView 8
Visualization	1 tasks	2 tasks	3 tasks	0 tasks	4 tasks	3 tasks	3 tasks	6 tasks	6 tasks
Visualization +ResultTable	22 tasks	25 tasks	19 tasks	15 tasks	17 tasks	22 tasks	22 tasks	18 tasks	18 tasks
ResultTable	9 tasks	5 tasks	10 tasks	17 tasks	11 tasks	7 tasks	7 tasks	8 tasks	8 tasks

Figure 218: Usage of the components in the Visualization plus ResultTable conditions per number of keywords

4.3.6.4. Summary of the hard facts results

When summarizing the hard facts results the temporal efficiency of the different components in the different situations may be a good indicator. As discussed above, handling the temporal efficiency in the evaluation setting used is not easy. Therefore using an pragmatic approach the overview will concentrate on effectiveness and task time as isolated items. When summarizing the results the effectiveness will be weighted a little bit more than the task time. Task times were restricted to five or ten minutes per question in the experiment. So there are no extraordinarily long task times that are be far away from real-life situations. The effectiveness values found in the experiment are therefore the important indicator for the real life usability of the different user interface conditions.

In general, the ResultTable and the Visualization plus ResultTable conditions all showed more or less lower mean effectiveness index values and at the same time higher mean task time index values than the HTML-List. The lower effectiveness in comparison with the HTML-List is only significant for the BarGraph condition. The differences for the effectiveness values of the other user interface conditions are not significant. The task time index differences in comparison with the HTML-List were significant for all user interface conditions. For the ScatterPlot condition, the effect was highly significant. Despite the work invested in the project, the implemented and tested visualizations of search results from the World Wide Web all performed more or less worse than the traditional HTML-List in terms of effectiveness and task time.

None of the user interface conditions showed higher mean effectiveness values than the HTML-List and all had significantly higher mean task time values. The temporal efficiency of the Result-Table and the Visualization plus ResultTable conditions may therefore not be better than that of the HTML-List. A detailed analysis of the temporal efficiency for three selected questions revealed that in all except one case, the ResultTable and the Visualization plus ResultTable conditions showed a lower mean temporal efficiency than the HTML-List. The exception was that in one of the three questions, the BarGraph condition performed better than the HTML-List. A statistical validation showed that most of the differences found were not statistically significant at this level of analysis. Significant differences were only found for the ResultTable in one question. Highly significant differences were found for the ScatterPlot condition in two of the three questions. Despite the missing statistical significance at this detailed level of analysis, the results for these tasks support the overall impression that the temporal efficiency of the tested visualizations of search results is worse than that of the traditional HTML-List.

Within the experts group, all user interface conditions show lower mean effectiveness index values than the HTML-List. This is consistent with the results from the analysis at the overall level. Within the beginners group most of the user interface conditions also performed worse than the HTML-List, except the ResultTable that got a higher mean effectiveness index than the HTML-List. This special result concerning the beginners and the BarGraph condition is not statistically significant. The only statistically relevant difference in comparison with the HTML-List was found for the SegmentView condition where the experts performed significantly less effectively than with the HTML-List. As was the case at the global level, in both groups the HTML-List shows the shortest mean task time indexes of all user interface conditions. The levels of significance of these findings are different at the target user group level from the ones found on the global level. In summary, there are some minor differences present for the effectiveness and the task time patterns of the experts and the beginners in the experiment. Besides the interpretation that the usability patterns of the visualizations are the same for experts and beginners, there could also be another reason for this effect: the groups tested were not really different concerning their level of expertise for the tasks tested. The “experts” group was characterized by having received formal training at least by attending a Faculty of Information Science Information Retrieval course. The “beginners” group had not had this formal training. Web search experience was not used as a criterion to separate experts and beginners. So it was the case that in the experts group, people had received some formal IR training but had little Web search experience. On the other hand, in the beginners group some of the subjects did not have formal IR training but years of Web search experience. In addition, the experiment itself was concentrated on one single phase of the complete information-seeking process: the review of results. It is really questionable if the distinction between “experts” and “beginners” in the evaluation setting was really valid for the tasks tested.

When looking at the two different task types tested for specific as well as for extended fact-finding tasks, the HTML-List shows the highest mean effectiveness and the lowest mean task time index values. In the case of the specific fact-finding tasks, the ResultTable has the same high mean effectiveness index value as the HTML-List. In none of the effectiveness index comparisons are the differences significant. The levels of significance for the task time differences on the task type level are somewhat different from the ones found on the global level. In general, it can be said, that there were effectiveness and task time differences between specific and extended fact tasks, most of them non-significant, but the patterns for the users interface conditions were not different for the two different types of tasks. So none of the visualizations turned out to be especially good or bad for either of the two types of fact-finding tasks.

Concerning the size of the result set presented in the case of the 30 hits tasks, all user interface conditions show the same or a higher mean effectiveness index than the HTML-List. This is the complete opposite of the overall trend. In the case of the 500 hits, all other user interface conditions show a lower mean effectiveness index than the HTML-List. This is consistent with the overall trend. When studying the task times inside the 30 hits tasks group differences are small. For the 500 hits tasks the HTML-List shows the lowest mean task time index of all user interface conditions. When studying the significance for the 30 hits tasks, none of the differences in the effectiveness index or task time index comparisons with the HTML-List are significant or even reach the level of a trend ($p<.1$). For the 500 hits tasks BarGraph and SegmentView conditions show highly significant lower effectiveness index values than the HTML-List. All user interface conditions show highly significant longer task time index values than the HTML-List. What does this mean? The first interpretation could be that there are no significant differences between the user interface conditions for small result sets. Another interpretation could be that document sets with 30 hits are too small to find differences between the user interface conditions in terms of effectiveness and task time. Remembering from Chapter 2 that most people do not go beyond the first ten, twenty, or thirty hits in a result set, both interpretations lead to the insight that it does not matter how the results are presented, at least when the size of a result set is clipped to 30 hits. The form of presentation will not affect effectiveness and task time significantly. For large result sets, the form of presentation plays a more important role. In the case of our 500 hits result sets BarGraph and SegmentView conditions, both performed highly significant worse than the HTML-List in terms of effectiveness and task time. For the ResultTable and the ScatterPlot condition the same effect was confirmed for the task time.

When studying the effects of the number of keywords, in the case of the one-keyword tasks, BarGraph and SegmentView conditions show higher mean effectiveness indexes than the HTML-List. The ResultTable and the ScatterPlot condition show lower values. The three-keywords tasks show the same pattern, except for the SegmentView condition where the mean effectiveness index is equal with that of the HTML-List. The eight-keyword tasks show the opposite pattern. The ResultTable and the ScatterPlot condition show higher mean effectiveness values; the BarGraph and the SegmentView conditions show lower mean effectiveness values than the HTML-List. This means that for the eight-keywords tasks, the visualizations that show the relevancies of each single keyword seem to drop down in their effectiveness values. Unfortunately, for none of the comparisons the differences of the effectiveness between the user interface conditions reach the level of significance. Therefore the drop down remains an impression. There may really be no significant differences or the size of the samples for this test were too small to get results on a significant level. For most of the comparisons the HTML-List shows a lower mean task time index than the other user interface conditions of the comparison. Some exceptions can be found, but in general the trend is the same as at the overall level. Most of the differences do not reach the level of significance. The missing significance of the differences of the task time could have been caused by the smaller size of the samples used to test the effect of the number of keywords.

Besides the insight concerning the distinction between experts and beginners, a number of other points showed up what could have been made better in the evaluation. Taking, for example, the “chess”-question the uncertainty concerning the “Kasparov lost / Deep Blue won” answers could have been alleviated if the evaluation setting had allowed a more detailed analysis of the subjects’ behavior after the test. This could have been the initially planned ScreenCam recording or a log file recording which documents have been viewed. Such a log file would also allow the examination of some additional measures like, for example, effort as number of viewed files. Another point which could be discussed is the number of variables used. The idea of getting results that are valid for a larger part of real-world situations clearly collides with the possibilities of interpreting the results and the lucidity of the findings. This situation is further complicated by the sensitivity of measuring effectiveness and task time for the special characteristics of each individual task used. The “Danube”-question is a good example of this sensitivity. Nevertheless, the results of the evaluation are an important step on the way to more insights about the usability of visualizations of search results from the World Wide Web.

5. Summary and Outlook

In the introduction to this thesis, problems encountered when searching the Web have been outlined. In particular questions concerning information overload and selection have been mentioned. The usage of Information Visualization techniques has been proposed as a possible solution to solve some of these problems.

The main chapter about information seeking has been arranged in two sections. Following a short presentation of the differences between searching the Web and classical Information Retrieval, ideas found in the literature concerning how to structure the information seeking process have been presented. In the second part, some results relating to how users search the Web have been discussed. The presentation of structuring possibilities for information seeking processes is based mainly on models developed in the context of classical Information Retrieval. Specialties of the processes when searching the World Wide Web are presented. The discussion of models has been arranged into three levels of detail: a) high-level goals, tasks, and strategies b) functions, phases, and steps of searching, and c) low-level tasks, goals, and interface actions. With the task actions model, the four-phase framework of information seeking, and the TTT data type by task taxonomy, three models from Shneiderman have been selected for these three different levels. The discussion included mainly aspects from level a) and b). Therefore the TTT data type by task taxonomy played only a subordinate role in the following chapters. Nevertheless, level c) has been included in order to complete the overview. The theoretical discussion of the information seeking process has been followed by a presentation of empirical results about people's real search behavior while searching the Web. In the chapter describing how people search the Web, the results of the following four studies in which log files from large search engines had been analyzed have been presented. They are the Excite study by [Jansen, Spink, Bateman et al. 1998], the AltaVista study by [Silverstein, Henzinger, Marais et al. 1999], the 1998 Fireball study by [Hölscher 1998], and the 1999 Fireball study by [Röttgers 1999]. The most important results have been that the average length of a query is around two keywords (with an increasing tendency) and that in the majority of cases people do not go beyond the first page of ten results. The section closed with some results about differences in Web search behavior between user groups.

After a short presentation of the concepts of Information Visualization the next chapter began with the introduction of the reference model for visualization by [Card, Mackinlay, Shneiderman 1999]. The authors' idea was to describe the process of mapping raw data through data tables and visual structures to the views finally presented to the user at the screen. The model has then been used in the following chapters to structure overviews about techniques, to classify special aspects, or to describe the mapping process in the INSYDER system. The presentation of the state-of-the-art of Information Visualization was discussed at length in the thesis. The overview focused on the representation of search results and discussed the topic from several points of view. The overview began with aspects of metaphors which should normally facilitate access to systems especially for new users. This was followed by a paragraph describing on an abstract level techniques that are used in the context of Information Visualization. Subsequently numerous ideas concerning the visualization of search results have been presented. Therefore, wherever possible, a standard example of a Web search has been used. The component-oriented presentation has been structured into the visualization of queries or query attributes, the visualization of document attributes, and the visualization of interdocument similarities. For the visualization of interdocument connections other works have been referred to. A structured listing of the systems mentioned closed the discussion of visualization possibilities from different points of view. This has been followed by a discussion of multiple coordinated views and the question of when and how to use these concepts. The chapter about Information Visualization ended with the presentation of some results from empirical evaluations of the usefulness of selected visualizations, and a summary of the factors that influence the usefulness of visualizations in form of the "5T-Environment".

The empirical part of the work started with a description of the INSYDER project in which the software was developed used for the evaluation of different visualizations of search results. The functions of the system in general have been described, as well as its software architecture, the functions of the individual software modules, the prototype-based development process, and the first formative evaluations during the project. This has been followed by a detailed discussion of the implemented visualizations and the concrete mapping process from the raw data to the final visual structures and views. Problems which occurred in the context of this process have been discussed as well as a number of visualizations which for different reasons were not implemented in the final software version. The discussion of the evaluation performed began with a description of the hypotheses and variables, as well as the test procedure. Effectiveness, efficiency, expected value, and user satisfaction for the user interface conditions HTML-list, ResultTable, ScatterPlot plus ResultTable, BarGraph plus ResultTable, and SegmentView plus ResultTable have been all examined. The tests were performed with 40 users and twelve tasks each. The evaluation took place at the University of Konstanz in Spring 2000. The independent variables used were the user interface condition, the target user group (beginner / expert), the number of keywords (1 / 3 / 8), the number of documents presented as result set (30 / 500), and the type of task (specific fact-finding / extended find finding).

When studying the expected value of a component, it can be said that in the Visualization plus ResultTable conditions, where the user had the possibility to decide which component to use, in the majority of cases both components were used. When analyzing usage times in these conditions, the ResultTable was the favorite component of the users. It was used in all three user interface conditions with ScatterPlot, BarGraph, and SegmentView for more than 50% of the overall task time. Interpreting usage time as an indicator of expected value, the expected value of the ResultTable seemed to be higher than that of the other components for the users. Usage time of a component could be a misleading indicator for expected value, because it is possible that the usage of the component is necessary for a certain task, despite its not being favored by the user. Combined with the results from the questionnaire, usage time may be an indicator for expected value. In terms of usage time ratios, the ResultTable had the highest expected value, followed by the SegmentView, the BarGraph, and the ScatterPlot. The HTML-List was not included in the usage time comparison because time portions could only be calculated for the Visualization plus ResultTable conditions.

Summarizing the results from the questionnaire, it can be said that the users had a number of usability problems but in general seemed to have welcomed the possibilities offered by the Visualizations and by the ResultTable. A statistical analysis on the overall level for selected questions from the questionnaire revealed highly significant better user satisfaction values for the ResultTable in a comparison with the HTML-List. The SegmentView showed a trend in the same direction. Differences between experts and beginners were in general very low. When present, in some of the single questions they are mostly concentrated on the ScatterPlot. In the statistical analysis of the group of selected questions, the difference between HTML-List and ResultTable was significant for the beginners. For the experts it was a trend.

System speed was critical and was biased negative especially by processing times for the SegmentView and loading times or timeout pauses of the internal and external document browsers. A number of missing functions requested by the users had already been implemented in the INSYDER system by the time when the evaluation was performed, but had been deactivated to ease or control the evaluation setting. These features were not mentioned in the introduction to avoid distractions of the users. The users missed some of these features. Other features requested by the users not planned or implemented before the evaluation were integrated in the interim. For details see [Mußler 2002]. A new evaluation with more features activated may be interesting. Such an evaluation may show if the user satisfaction would be even higher in a full-featured version, or if the additional complexity would bias the results.

In general the focus of the user satisfaction questionnaire was centered on the visualizations and the review of results phase of the information seeking process. The evaluation itself was restricted to this area. User satisfaction concerning the support of the INSYDER system for the whole information seeking process may also be an interesting point to examine.

When summarizing the hard facts the temporal efficiency of the different components in the different situations may be a good indicator. For a number of reasons, temporal efficiency values were difficult to handle. Effectiveness was therefore the most important variable discussed. Task times were also reported for every condition. In general the ResultTable and the Visualization plus ResultTable conditions all showed more or less lower mean effectiveness index values and at the same time higher mean task time index values than the HTML-List. The lower effectiveness in comparison with the HTML-List is only significant for the BarGraph condition. The differences for the effectiveness values of the other user interface conditions are not significant. The task time index differences in comparison with the HTML-List were significant or highly significant for all user interface conditions. Despite the work invested in the project, the implemented and tested visualizations of search results from the World Wide Web all performed more or less worse than the traditional HTML-List in terms of effectiveness and task time. The temporal efficiency of the ResultTable and the Visualization plus ResultTable conditions may therefore not be better than that of the HTML-List. Despite a missing statistical significance, a detailed analysis of the temporal efficiency for the three selected specific fact-finding tasks seem to support this impression. The success of the HTML-List compared with the other user interface conditions may be an effect of experience. People are used to this form of presentation of search results, and our evaluation setting did not allow examination of the effect of training.

When studying experts and beginners as the two different target user groups used in both cases effectiveness index and task time index patterns are similar to those at the overall level. One exception occurred in the beginners group, where the ResultTable condition got a higher mean effectiveness index than the HTML-List, but the difference was not statistically significant. There may be really no differences in effectiveness and task time patterns between the two target user groups, but a post-test theoretical discussion revealed that it is really questionable whether the distinction between "experts" and "beginners" in the evaluation setting was really valid for the tasks tested.

Concerning the specific and extended fact-finding tasks used, none of the visualizations turned out to be especially good or bad for one of the two types of fact-finding tasks in terms of effectiveness or task time.

For small result sets with 30 hits, no significant differences between the user interface conditions were found. Remembering that most people do not go beyond the first ten, twenty, or thirty hits in a result set, the lesson learned is that for small result sets it does not really matter how the results are presented, at least when the size of a result set is clipped to 30 hits. The form of presentation does not affect effectiveness and task time significantly. For large result sets the form of presentation plays a more important role. In the case of the 500 hits result sets, BarGraph and SegmentView condition both performed highly significant worse than the HTML-List in terms of effectiveness and task time. For the ResultTable and the ScatterPlot condition, the same effect was confirmed for the task time.

When studying the effects of the number of keywords used and displayed, there is a contrast between one-keyword or three-keywords queries on one side and eight keyword queries on the other side. For the eight-keywords queries, the visualizations that show the relevancies of each single keyword seem to drop down in their effectiveness values. The BarGraph and the SegmentView conditions are particularly affected. Unfortunately, in none of the comparisons the differences of the effectiveness between the user interface conditions reach the level of significance. Therefore the drop down may just be an impression. There may really be no significant differences, or the samples' size for this test was too small to get results on a significant level.

Besides the insight concerning the distinction between experts and beginners, a number of other points highlighted what could have been made better in the evaluation. Taking for example the "chess"-question the uncertainty concerning the "Kasparov lost / Deep Blue won" answers could have been alleviated if the evaluation setting had allowed a more detailed analysis of the subjects' behavior after the test. This could have been the initially planned ScreenCam recording or a log file recording which documents have been viewed. Such a log file would also allow the examination of some additional measures like, for example, effort as number of viewed files. Another point which can be discussed is the number of variables used. The idea of getting results that are valid for a larger part of real-world situations clearly collides with the possibilities of interpreting the results and the lucidity of the findings. This situation is further complicated by the sensitivity of measuring effectiveness and task time for the special characteristics of each individual task used. The "Danube"-question is a good example for this sensitivity.

Despite all statistical validations, the discussion of a number of special effects showed how difficult it is to create a test setting to fairly compare different visualization components. Orientation on initiatives like the TREC interactive track may help to avoid settings where such effects will be too dominant. Nevertheless, even the most carefully planned evaluation setting for visualization components may lead to "wrong" results, because there are just too many factors biasing the success of a certain visualization for a certain user in a certain situation. Nevertheless, more and more evaluations undertaken will help to set up a framework for testing visualizations and finally shaping a picture of where and when to offer which possibilities to the user. Hopefully the contributions of this thesis and the results of the evaluation are an important step on the way to more insights about the usability of visualizations of search results from the World Wide Web.

The idea to use the principles for visual information seeking for searching the Web and using other presentation forms than the traditional HTML-List has been partially successful, at least in the way that users seem to be more satisfied when working with the system. Most of the users made use of the synchronized multiple visual views, and saw in them a nice enabling technology to find the most relevant documents in the search result. But there is room for improvements.

Possible future steps will be discussed grouped in further evaluations of the existing components and possible improvements of the components and the overall system. Starting with further evaluations of the existing components, the following questions may be of interest:

- Are there any performance differences between the five different SegmentView variants?
- Does additional training or more experience with the components influence their effectiveness and efficiency?
- Are there any performance differences in the Visualization plus ResultTable condition, between subjects that used both components or just one? Which one?
- Are the users able to decide which Visualization will help them best in certain situations?
- How much does the success of a visualization component depend on the underlying ranking component?

As mentioned above five different variants of the SegmentView were implemented: three TileBar versions and two StackedColumn versions, all showing the document data at segment level. Due to resource restrictions and so as not to blow up the evaluation setting, we did not compare the different versions. The users were free to choose which versions they used. All five were available in the SegmentView configuration. Mostly people just used the default one. Will there be differences between them in terms of effectiveness, efficiency, expected value, or user satisfaction?

In our evaluation the HTML-List performed best in terms of effectiveness, task time, and efficiency. As stated above, this may be an effect of experience, because people are used to this form of presentation of Web search results. It would be an interesting test to see if the performance of the other visualizations improves, when the factor of experience or training is modified by having people use the system over a longer period of time or offering a noticeably greater amount of training to a certain group of users. [Sebrechts, Vasilakis, Miller et al. 1999] found a decrease in response time for new visualizations during their experiment even without special training.

In the Visualization plus ResultTable conditions, the users were free to use the Visualization, the ResultTable, or both. Our evaluation setting had the goal of finding the added value of the Visualization in addition to the ResultTable. Unfortunately the setting did not allow to compare the performance values between different user choices. It may be interesting to compare Visualization alone with ResultTable alone and Visualization plus ResultTable condition.

When studying the user decisions regarding when to use which component or combination of components in the Visualization plus ResultTable conditions, in a number of cases the users made decisions which seem to be logical to the researcher. An example is not to use the BarGraph for queries with only one keyword. Altogether the freedom of choice was very low for the users. The five different components were never available all at one time for one task. In two out of five cases the users had no choice, and in the remaining cases there was always just a choice between the ResultTable and only one additional Visualization. Are users able to decide (in advance) which component will help them best? Will performance decrease because of lost time when making wrong decisions?

As we have seen in things like impossible keyword highlighting or the phantom hits in the SegmentView caused by the adjacency ranking of the following segments, there are dependencies between the ranking module or its output and the possibilities of the visualization component. The question is due of to what extent the success of a certain component will be influenced by these dependencies?

Turning now to the possible improvements of the components and the overall system, two ideas in particular will be discussed:

- The Integration of Table Lens, ResultTable, HTML-List, BarGraph, and SegmentView into one component.
- Improvements for the ScatterPlot component.

The users requested a number of features for the BarGraph and the SegmentView already implemented in the ResultTable. On the other hand they requested that the Document Tooltip from these two components also should be available in the ResultTable. Even with the existing ResultTable the user satisfaction values for the ResultTable were significantly better than for the HTML-List. The ResultTable performed nearly as well as the HTML-List in terms of effectiveness and task time. Differences may for example have been caused by the fact that the document extract was not fully visible in the ResultTable, and line breaks had not been used to enhance the readability of the text. Both had been the case with regard to the HTML-List. In addition, the HTML-List had common HTML-navigation elements. All this could also be implemented in the ResultTable. As we have seen on page 110, for the Table Lens there are already components existing integrating smoothly a result table and a bar graph. Additionally, the ResultTable of the INSYDER system already contains a simple predecessor of the SegmentView in form of the Relevance Curve. With the integration of the DocumentVector into the ScatterPlot, we have already shown that the integration of visual structures into one component is possible, when offering the user easy mechanisms for changing views. So the proposed "SuperTable" must integrate a multiple focus Table Lens, the HTML-List, a BarGraph and the SegmentView with TileBars and StackedColumn in a way that allows easy manipulation of the table. This could be done for example by including a set of radio buttons like those already implemented for the ScatterPlot. Instead of predefined combinations of dimensions and axes, there would be different predefined zoom-levels. Starting for example with a BarGraph view in a first zoom level, revealing document details in a one-line-per-document-mode for the whole table in a second level, a several-line-per-document-mode in a third level, and document information revealing segment details in a fourth level. In addition, tooltips or lens mechanisms can allow easy inspection of details for single documents. Power users can use multiple focus possibilities for comparisons. The SuperTable as an integrated component will be more complex than the former single components. With adequate design a subject who is starting to use the system may not be aware of this complexity, and therefore would not be distracted. The integration will minimize context-switching effort and is able to allow a smooth learning curve from beginner mode to power user mode. The evaluation of such a highly integrated component may be demanding, but expectations of possible improvements concerning effectiveness, efficiency, and user satisfaction are high.

Of the five components implemented in the INSYDER system, four were mentally integrated above into the SuperTable, leaving the ScatterPlot. Whereas the SuperTable basically has a one-dimensional layout, the ScatterPlot has a two-dimensional one. It cannot be integrated into the SuperTable. Nevertheless, there are also possibilities of improving the ScatterPlot. Some examples include packing more information into the display, for example, by using partially filled circles in the ScatterPlot, as done by [Fishkin, Stone 1995] with their real-valued queries. If this is done the overall relevance of a document will always be visible, even when the dimension is not mapped to one of the axes. With the callout lens[180] from the same authors, an exploder for document groups can be integrated allowing to fulfill one of the requests formulated by the users. The callout lens can be used to explode "clumps" of icons or document group symbols. With this and other enhancements, the ScatterPlot could serve its functions by giving the user a quick overview of all search results, and offering the user a good variety of controls (e.g. defining own views, zooming, selecting) to reduce the amount of hits to a smaller group of interesting documents. These documents can then be selected by the mouse and analyzed in more detail in the SuperTable. In Chapter 3.4, the possible advantages of multiple coordinated views have been discussed. An enhanced ScatterPlot with additional lens mechanisms tightly coupled with a SuperTable and a document browser may be a good implementation.

Figure 219 shows as an extension of Figure 132 on page 157 which components were integrated into the INSYDER system and its components in a first step, and what can be integrated in the proposed next step into the ScatterPlot and the SuperTable.

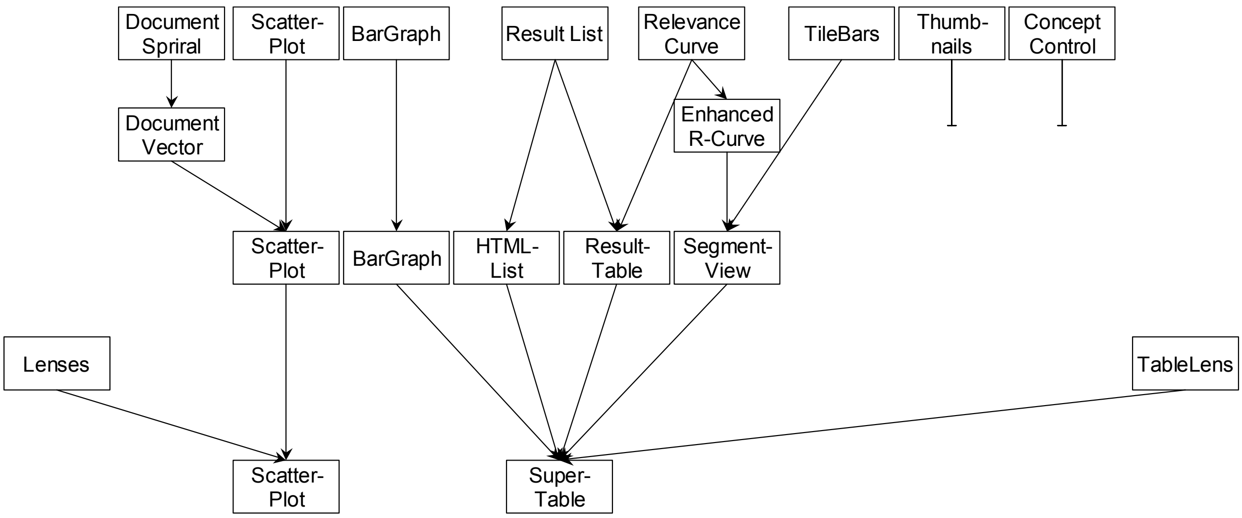


Figure 219: Further integration of the components and of additional ideas

Back from ideas to reality, it can be said that the creation, implementation, and evaluation of visualization ideas for search results from the World Wide Web is just at the beginning. It is difficult and there is a lot of work to do. Let's do it.

[180] See page 68 for a short description

6. References

[Aguilar 1967] Aguilar, Francis J.: *Scanning the Business Environment.* New York, NY (Macmillan Co.) 1967.

[Ahlberg, Shneiderman 1994] Ahlberg, Christopher; Shneiderman, Ben: *Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays.* In: Adelson, B.; Dumais, S.; Olson, J. S. (Eds.): *CHI 1994: Conference Proceedings Human Factors in Computing Systems. Conference: Boston, MA, April 24-28 1994.* New York (ACM Press) 1994. p. 313-317.

[Ahlberg, Wistrand 1995] Ahlberg, Christopher; Wistrand, Erik: *IVEE: An Information Visualization and Exploration Environment.* In: Gershon, Nahum; Eick, Stephen G. (Eds.): *Proceedings of IEEE Information Visualization 1995. Conference: Atlanta, GA, October 30-31 1995.* Los Alamitos, CA (IEEE Computer Soc. Press) 1995. p. 66-73.

[Allan, Leouski, Swan 1997] Allan, James; Leouski, Anton V.; Swan, Russell C.: *Interactive cluster visualization for information retrieval.* Technical report IR-116, Department of Computer Science, University of Massachusetts, Amherst, 1997

[Allen 2000] Allen, Bryce: *Individual Differences and the Conundrums of User-Centered Design: Two Experiments.* In: *Journal of the American Society for Information Science (JASIS)*, 51 (2000) 6, p. 508-520.

[Allen, Allen 1993] Allen, Bryce; Allen, Gilian: *Cognitive Abilities of Academic Librarians and Their Patrons.* In: *College & Research Libraries*, 54 (1993) 1, p. 67-73.

[Alonso, Baeza-Yates 1998] Alonso, Omar; Baeza-Yates, Ricardo A.: *A model for visualizing large answers in WWW retrieval.* In: *Proceedings SCCC'98. 18th International Conference of the Chilean Society of Computer Science. Conference: Antofogasta, Chile, November 9-14 1998.* Los Alamitos, CA (IEEE Comput. Soc.) 1998. p. 2-7.

[Amento, Hill, Terveen et al. 1999] Amento, Brian S.; Hill, William C.; Terveen, Loren G.: *An Empirical Evaluation of User Interfaces for Topic Management of Web Sites.* In: Williams, Marian G.; Altom, Mark W.; Ehrlich, Kate et al. (Eds.): *CHI 1999: Conference Proceedings Human Factors in Computing Systems. Conference: Pittsburgh, PA, May 17-20 1999.* New York (ACM Press) 1999. p. 552-559.

[Andrews 1995] Andrews, Keith: *Visualising Cyberspace: Information Visualisation in the Harmony Internet Browser.* In: Gershon, Nahum; Eick, Stephen G. (Eds.): *Proceedings of IEEE Information Visualization 1995. Conference: Atlanta, GA, October 30-31 1995.* Los Alamitos, CA (IEEE Computer Soc. Press) 1995. p. 97-104.

[Andrienko, Andrienko 1997] Andrienko, Gennady L.; Andrienko, Natalia V.: *Knowledge-Based Support for Visual Exploration of Spatial Data.* In: *Extended Abstracts of ACM CHI'97 Conference* New York 1997. p. 16-17.

[Anick, Brennan, Flynn et al. 1990] Anick, Peter G.; Brennan, Jeffrey D.; Flynn, Rex A.: *A Direct Manipulation Interface for Boolean Information Retrieval via Natural Language Query.* In: Vidick, Jean-Luc (Eds.): *SIGIR 1990: Proceedings of the 13th International Conference on Research and Development in Information Retrieval. Conference: Brussels, Belgium, September 5-7 1990.* New York (ACM Press) 1990. S. 135-150.

[Ankerst, Keim, Kriegel 1996] Ankerst, Michael; Keim, Daniel A.; Kriegel, Hans-Peter: *Circle Segments: A Technique for Visually Exploring Large Multidimensional Data Sets.* In: Yagel, Roni; Nielson, Gregory M. (Eds.): *Proceedings of IEEE Visualization 96 Conference: San Francisco, CA, October 27 - November 1, 1996.* Los Alamitos, CA (IEEE Computer Soc. Press) 1996. Hot Topic Session. http://www.dbs.informatik.uni-muenchen.de/dbs/projekt/papers/Vis96.ps [1999-01-12]

[Asahi, Turo, Shneiderman 1995] Asahi, Toshiyuki; Turo, David; Shneiderman, Ben: *Using treemaps to visualize the analytic hierachy process* In: *Information Systems Research*, 6 (1995) 4, p. 357-375.

[Au, Carey, Sewraz et al. 2000] Au, Peter; Carey, Matthew; Sewraz, Sahlini et al.: *New Paradigms in Information Visualization.* In: Belkin, Nicholas J.; Ingwersen, Peter; Leong, Mun-Kew (Eds.): *SIGIR 2000: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Conference: Athens, Greece, July 24-28 2000.* New York (ACM Press) 2000. S. 307-309.

[Ayers, Stasko 1995] Ayers, Eric Z.; Stasko, John T.: *Using Graphic History in Browsing the World Wide Web.* In: *WWW 4: Fourth International World Wide Web Conference. Conference: Boston, MA, December 11-14 1995.* http://www.w3.org/Conferences/WWW4/Papers2/270/ [1999-08-12]

[Baeza-Yates 1996] Baeza-Yates, Ricardo A.: *Visualizing large answers in text databases.* In: *AVI 1996: International Workshop on Advanced Visual Interfaces. Conference: Gubio, Italy, May 27-29 1996.* New York (ACM Press) 1996. p. 101-107.

[Baeza-Yates, Ribeiro-Neto 1999] Baeza-Yates, Ricardo A.; Ribeiro-Neto, Berthier (Eds.): *Modern Information Retrieval.* Reading, MA (Addison Wesley Longman) 1999.

[Baldonado, Woodruff, Kuchinsky 2000] Baldonado, Michelle Q.; Woodruff, Allison; Kuchinsky, Allan: *Guidelines for Using Multiple Views in Information Visualization.* In: Di Gesù, Vito; Levialdi, Stefano; Tarantino, Laura (Eds.): *AVI 2000: International Workshop on Advanced Visual Interfaces. Conference: Palermo, Italy, May 23-26 2000.* New York (ACM Press) 2000. p. 110-119.

[Bartram 1997] Bartram, Lyn: *Can motion increase user interface bandwidth in complex systems?* In: *Proceedings of IEEE Conference on Systems, Man and Cybernetics 1997* p. 1686-1691.

[Bates 1989] Bates, Marcia J.: *The Design of Browsing and Berrypicking Techniques for the Online Search Interface.* In: *Online Review*, 13 (1989) 5, p. 407-424.

[Beard, Walker 1990] Beard, David V.; Walker, John Q.: *Navigational techniques to improve the display of large two-dimensional space.* In: *Behavior and Information Technology*, 9 (1990) 6, p. 451-466.

[Bederson, Hollan 1994] Bederson, Benjamin B.; Hollan, James D.: *PAD++: A zooming graphical user interface for exploring alternate interface physics* In: *UIST 94: 7th ACM Symposium on User Interface Software and Technology* New York (ACM press) 1994. p. 17-27.

[Bederson, Hollan, Perlin et al. 1996] Bederson, Benjamin B.; Hollan, James D.; Perlin, Ken et al.: *Pad++: A zoomable graphical sketchpad for exploring alternate interface physics.* In: *Journal of Visual Languages and Computing*, 7 (1996) 1, p. 3-31.

[Bederson, Hollan, Stewart et al. 1998] Bederson, Benjamin B.; Hollan, James D.; Stewart, Jason et al.: *A Zooming Web Browser.* In: Forsythe, Chris; Grose, Eric; Ratner, Julie (Eds.): *Human Factors and Web Development.* Mahwah, NJ (Lawrence Erlbaum) 1998. p. 255-266.

[Bekavac 1999] Bekavac, Bernard: *Hypertextgerechte Suche und Orientierung im WWW. Ein Ansatz unter Berücksichtigung hypertextspezifischer Struktur- und Kontextinformation.* Universität Konstanz, Fachgruppe Informatik und Informationswissenschaft, Dissertation 1999

[Belkin 1980] Belkin, Nicholas J.: *Anomalous states of knowledge as a basis for information retrieval.* In: *Canadian Journal of Information Science*, 5 (1980), p. 133-143.

[Belkin, Cool, Stein et al. 1995] Belkin, Nicholas J.; Cool, Colleen; Stein, Adelheit et al.: *Cases, scripts, and information-seeking strategies: on the design of interactive information retrieval systems.* In: *Expert Systems with Applications*, 9 (1995) 3, p. 379-395.

[Belkin, Croft 1992] Belkin, Nicholas J.; Croft, W. Bruce: *Information filtering and information retrieval: Two sides of the same coin?* In: *Communications of the ACM*, 35 (1992) 12, p. 29-38.

[Belkin, Marchetti, Cool 1993] Belkin, Nicholas J.; Marchetti, Pier G.; Cool, Colleen: *BRAQUE: Design of an Interface to Support User Interaction in Information Retrieval.* In: *Information Processing & Management*, 29 (1993) 3, p. 325-344.

[Belkin, Oddy, Brooks 1982] Belkin, Nicholas J.; Oddy, Robert N.; Brooks, Helen M.: *ASK for information retrieval: Parts I.* In: *Journal of Documentation*, 38 (1982) 2, p. 61-71.

[Belkin, Oddy, Brooks 1982a] Belkin, Nicholas J.; Oddy, Robert N.; Brooks, Helen M.: *ASK for information retrieval: Parts II.* In: *Journal of Documentation*, 38 (1982) 3, p. 145-164.

[Benford, Snowdon, Greenhalgh et al. 1995] Benford, Steve D.; Snowdon, Dave; Greenhalgh, Chris et al.: *VR-VIBE: a virtual environment for co-operative information retrieval.* In: *Computer Graphics Forum*, 14 (1995) 3, p. 349-360.

[Berenci, Carpineto, Giannini 1998] Berenci, Ezio; Carpineto, Claudio; Giannini, Vittorio: *Improving the Effectiveness of WEB Search Engines Using Selectable Views of Retrieval Results.* In: *Journal of Universal Computer Science*, 4 (1998) 9, p. 737-747.

[Bertin 1977] Bertin, Jacques: *Graphics and Graphic Information Processing.* Berlin (De Gruyter) 1977.

[Bertin 1982] Bertin, Jacques: *Graphische Darstellungen. Graphische Verabeitung von Informationen.* Berlin, New York (De Gruyter) 1982.

[Bier, Stone, Fishkin et al. 1994] Bier, Eric A.; Stone, Maureen C.; Fishkin, Ken et al.: *A Taxonomy of See-Through Tools.* In: Adelson, B.; Dumais, S.; Olson, J. S. (Eds.): *CHI 1994: Conference Proceedings Human Factors in Computing Systems. Conference: Boston, MA, April 24-28 1994.* New York (ACM Press) 1994. p. 358-364.

[Borgman 1989] Borgman, Christine L.: *All users of information systems are not created equal: An exploration into individual differences.* In: *Information Processing & Management*, 25 (1989) 3, p. 237-252.

[Bourdoncle 1997] Bourdoncle, François: *LiveTopics: Recherche Visuelle d'Information sur l'Internet.* In: *Dossiers de l'Audiovisuel, La Documentation Française*, 74 (1997) Juillet-Aout, p. 36-38.

[Bourdoncle 1999] Bourdoncle, François: *Panorama et perspectives des outils de recherche d'Information textuelle sur Internet.* In: *IDT 1999: Salon Professionel de l'Information Electronique et de L'Internet Conference: Paris, France, June 10 1999.* http://www.idt.fr/idt/pages_fra/actes/actes99/sommaire.html [2001-01-29]

[Bourdoncle, Bertin 2000] Bourdoncle, François; Bertin, Patrice: *Recherches d'Aiguilles dans une Botte de Liens.* In: *La Recherche*, 328 (2000) Février. http://www.larecherche.fr/data/328/03280661.html [2001-01-28]

[Brown, Shillner 1995] Brown, Marc H.; Shillner, Robert A.: *A New Paradigm for Browsing the Web. Short Papers.* In: Katz, Irvin R.; Mack, Robert L.; Marks, Linn et al. (Eds.): *CHI 1995: Conference Proceedings Human Factors in Computing Systems. Conference: Denver, CO, May 7-11 1995.* New York (ACM Press) 1995. p. 320-321.

[Bryan, Gershman 2000] Bryan, Dough; Gershman, Anatole: *The Aquarium: A Novel User Interface Metaphor for Large, Online Stores.* In: Tjoa, A Min; Wagner, Roland R.; Al-Zobaidie, Ala (Eds.): *Proceedings 11th International Workshop on Database and Expert Systems Applications. Conference: Greenwich, London, United Kingdom, September 4-8 2000.* Los Alamitos, CA (IEEE Computer Society) 2000. p. 601-607.

[Bürdek, Eibl, Krause 1999] Bürdek, Bernhard E.; Eibl, Maximilian; Krause, Jürgen: *Visualisation in Document Retrieval: An Example of the Integration of Software Ergonomics and an Aesthetic Quality in Design.* In: Paton, Ray; Neilson, Irene (Eds.): *Proceedings of the VRI'98 - Visual Representations and Interpretations. Conference: Liverpool, United Kingdom, September 22-24 1998.* London (Springer Verlag) 1999. p. 387-396.

[Byrd 1999] Byrd, Donald: *A Scrollbar-based Visualization for Document Navigation.* In: Fox, Edward A. et al. (Eds.): *Proceedings of the fourth ACM conference on digital libraries.* New York (ACM Press) 1999. p. 122-129.

[Card, Mackinlay, Shneiderman 1999] Card, Stuart K.; Mackinlay, Jock D.; Shneiderman, Ben (Eds.): *Readings in Information Visualization. Using Vision to Think.* San Francisco, CA (Morgan Kaufmann) 1999.

[Card, Robertson, York 1996] Card, Stuart K.; Robertson, George G.; York, William: *The WebBook and the Web Forager: An Information Workspace in the World Wide Web.* In: Tauber, M. J.; Bellotti, V.; Jeffries, R. et al. (Eds.): *CHI 1996: Conference Proceedings Human Factors in Computing Systems. Conference: Vancouver, BC, Canada, April 13-18 1998.* New York (ACM Press) 1996. p. 111-117.

[Carey 2000] Carey, Matthew: *An Information Navigator Suite.* http://km.doc.ic.ac.uk/pr-m.carey-2000/mc99-report.pdf [2001-02-03]

[Carey, Kriwaczek, Rüger 2000] Carey, Matthew; Kriwaczek, Frank; Rüger, Stefan M.: *A Visualization Interface for Document Searching and Browsing.* In: *Proceedings of NPIVM 2000 Conference: Washington, DC, November 10-11 2000.* New York (ACM Press) 2000. http://rowan.doc.ic.ac.uk/www-pub/vidsb-npivm2000.pdf [2001-03-04]

[Carrière, Kazman 1995] Carrière, Jeromy; Kazman, Rick: *Interacting with huge hierarchies: Beyond cone trees.* In: Gershon, Nahum; Eick, Stephen G. (Eds.): *Proceedings of IEEE Information Visualization 1995. Conference: Atlanta, GA, October 30-31 1995.* Los Alamitos, CA (IEEE Computer Soc. Press) 1995. p. 74-81.

[Carrière, Kazman 1997] Carrière, Jeromy; Kazman, Rick: *WebQuery: Searching and Visualizing the Web through Connectivity.* In: *WWW 6: Sixth International World Wide Web Conference. Conference: Santa Clara, CA, April 7 - 11 1997.* http://decweb.ethz.ch/WWW6/Technical/Paper096/Paper96.html [1999-01-11]

[Carroll, Mack, Kellogg 1988] Carroll, John M.; Mack, Robert L.; Kellogg, Wendy A.: *Interface Metaphors and User interface Design.* In: Helander, Martin (Ed.): *Handbook of Human-Computer Interaction* Amsterdam (North-Holland) 1988. p. 67-85.

[Casner 1991] Casner, Stephen M.: *A Task-Analytic Approach to the Automated Design of Graphic Presentations.* In: *ACM Transactions on Graphics*, 10 (1991) 2, p. 111-151.

[Chalmers 1993] Chalmers, Matthew: *Using a Landscape to represent a corpus of documents.* In: Frank, A.; Caspari, I. (Eds.): *Spatial Information Theory. Proceedings of COSIT'93, European Conference on Spatial Information Theory. Elba, Italy, September 1993* Berlin (Springer) 1993. p. 377-390.

[Chalmers 1995] Chalmers, Matthew: *Design Perspectives in Visualising Complex Information.* In: Spaccapietra, Stefano; Jain, Ramesh (Eds.): *Proc 3rd IFIP Visual Databases Conference (VDB.3).* London (Chapman & Hall) 1995. p. 103-111.

[Chalmers 1996] Chalmers, Matthew: *A linear iteration time layout algorithm for visualising high-dimensional data.* In: Yagel, Roni; Nielson, Gregory M. (Eds.): *Proceedings of IEEE Visualization 1996. Conference: San Francisco, CA, 1996 October 27 - November 1* Los Alamitos, CA (IEEE Computer Soc. Press) 1996. p. 127-132.

[Chalmers, Chitson 1992] Chalmers, Matthew; Chitson, Paul: *Bead: Explorations in information visualization.* In: Belkin, Nicholas J.; Ingwersen, Peter; Pejtersen, Annelise M. (Eds.): *SIGIR 1992: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Conference: Copenhagen, Denmark, June 21-24 1992.* New York (ACM Press) 1992. p. 330-337.

[Chase, D'Amore, Gershon et al. 1998] Chase, Penny; D'Amore, Ray; Gershon, Nahum et al.: *Semantic Visualization.* In: Pustejovsky, James; Maybury, Mark (Eds.): *Proceedings of the COLING-ACL '98 Workshop on Content Visualization and Intermedia Representation, Montreal, August 1998.* p. 52-62.

[Chen, Czerwinski 1997] Chen, Chaomei; Czerwinski, Mary P.: *Spatial Ability and Visual Navigation: An Empirical Study.* In: *The New Review for Hypertext and Multimedia*, 3 (1997), p. 67-90.

[Chen, Schuffels, Orwig 1996] Chen, Hsinchun; Schuffels, Chris; Orwig, Richard E.: *Internet categorization and search: a self-organizing approach.* In: *Journal of visual communication and image representation*, 7 (1996) 1, p. 88-102.

[Chen, Yu 2000] Chen, Chaomei; Yue, Yu: *Empirical studies of information visualization: a meta-analysis.* In: *International Journal of Human-Computer Studies*, 53 (2000) 5, p. 851-866.

[Chi 2000] Chi, Ed H.: *A Taxonomy of Visualization Techniques Using the Data State Reference Model.* In: Roth, Steven F.; Keim, Daniel A. (Eds.): *Proceedings of IEEE Information Visualization 2000. Conference: Salt Lake City, UT, October 9-10 2000.* Los Alamitos, CA (IEEE Computer Soc. Press) 2000. p. 69 -75.

[Chi, Riedl 1998] Chi, Ed. H.; Riedl, John T.: *An Operator Interaction Framework for Visualization Systems.* In: Karat, Clare-Marie; Karat, John; Horrocks, Ian (Eds.): *CHI 1998: Conference Proceedings Human Factors in Computing Systems. Conference: Los Angeles, CA, April 21-23 1998.* New York (Addison-Wesley) 1998. p. 63-70.

[Chimera 1992] Chimera, Richard: *Value Bars: An Information Visualization and Navigation Tool for Multi-attribute Listings.* University of Maryland. Department of Computer Science technical report CS-TR-2773

[Choo, Detlor, Turnbull 1998] Choo, Chun Wei; Detlor, Brian; Turnbull, Don: *A Behavioral Model of Information Seeking on the Web - Preliminary Results of a Study of How Managers and IT Specialists use the Web.* In: Preston, Cecilia M. (Ed.): *ASIS 1998: Proceedings of the 61st Annual Meeting (Vol. 35). Conference: Pittsburgh, PA, October 24-29 1998.* Medford, NY (Information Today) 1998. http://choo.fis.utoronto.ca/FIS/ResPub/asis98/ [1999-07-23]

[Choo, Detlor, Turnbull 1999] Choo, Chun Wei; Detlor, Brian; Turnbull, Don: *Information Seeking on the Web—An Integrated Model of Browsing and Searching. - preprint -* In: *ASIS 1999: Proceedings of the 62 nd Annual Meeting (Vol. 36). Conference: Washington DC, USA, 1999 October 31 - November 4* Medford, NY (Information Today) 1999. –preprint - http://choo.fis.utoronto.ca/fis/respub/asis99/default.html [1999-07-23]

[Church, Helfman 1993] Church, Kenneth W.; Helfman; Jonathan I.: *Dotplot: A Program for Exploring Self-Similarity in Millions of Lines of Text and Code.* In: *Journal of Computational and Graphical Statistics*, 2 (1993) 2, p. 153-174.

[Cockburn, Greenberg 1999] Cockburn, Andy; Greenberg, Saul: *Beyond the ‚Back' Button: Issues of Page Representation and Organisation in Graphical Web Navigation Tools.* Research Report 99-640-03, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada. Paper plus 6.5 minute video. Replaced by [Cockburn, Greenburg, McKenzie et al. 1999], [Cockburn, Greenberg 1999a]

[Cockburn, Greenberg 1999a] Cockburn, Andy; Greenberg, Saul: *Issues of Page Representation and Organisation in Web Browser's Revisitation Tools.* http://www.cpsc.ucalgary.ca/grouplab/papers/1999/99-IssuesRevisitationTools.Ozchi/issues-revisitation-tools.pdf [1998-08-10]

[Cockburn, Greenburg, McKenzie et al. 1999] Cockburn, Andy; Greenberg, Saul; McKenzie, Bruce et al.: *WebView: A Graphical Aid for Revisiting Web Pages.* http://www.cpsc.ucalgary.ca/grouplab/papers/1999/99-WebView.Ozchi/webview.pdf [1999-08-10]

[Combs, Bederson 1999] Combs, Tammara T.; Bederson, Benjamin B.: *Does zooming improve image browsing?* In: Fox, Edward A. et al. (Eds.): *Proceedings of the fourth ACM conference on Digital libraries.* New York (ACM Press) 1999. p. 130-137.

[Croft, Cook, Wilder 1995] Croft, W. Bruce; Cook, Robert; Wilder, Dean: *Providing Government Information on the Internet: Experiences with THOMAS.* In: *Digital Libraries 1995: The Second Annual Conference on the Theory and Practice of Digital Libraries. Conference: Austin, TX, June 11-1 1995.* http://www.csdl.tamu.edu/DL95/ [2000-10-20] p. 19-24.

[Cugini 2000] Cugini, John V.: *Presenting Search Results: Design, Visualization and Evaluation.* In: *Workshop: Information Doors - Where Information Search and Hypertext Link (held in conjunction with the ACM Hypertext and Digital Libraries conferences) Conference: San Antonio, TX, May 30 2000.* http://www.ics.mq.edu.au/~einat/info_doors/program/Cugini.html [2000-07-27]

[Cugini, Laskowski, Piatko 1998] Cugini, John V.; Laskowski, Sharon L.; Piatko, Christine: *Document Clustering in Concept Space: The NIST Information Retrieval Visualization Engine (NIRVE).* http://zing.ncsl.nist.gov/~cugini/uicd/cc-paper.html [1998-09-10]

[Cugini, Laskowski, Sebrechts 2000] Cugini, John V.; Laskowski, Sharon L.; Sebrechts, Marc M.: *Design of 3-D Visualization of Search Results: Evolution and Evaluation.* In: *Proceedings of IST/SPIE's 12th Annual International Symposium: Electronic Imaging 2000: Visual Data Exploration and Analysis (SPIE 2000) Conference: San Jose, CA, 23-28 January 2000* p. 23-28.

[Cugini, Piatko, Laskowski 1997] Cugini, John V.; Piatko, Christine; Laskowski, Sharon L.: *Interactive 3D Visualization for Document Retrieval.* http://zing.ncsl.nist.gov/~cugini/uicd/viz.html [1998-09-10]

[Cutting, Karger, Pedersen et al. 1992] Cutting, Douglass R.; Karger, David R.; Pedersen, Jan O. et al.: *Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections.* In: Belkin, Nicholas J.; Ingwersen, Peter; Pejtersen, Annelise M. (Eds.): *SIGIR 1992: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Conference: Copenhagen, Denmark, June 21-24 1992.* New York (ACM Press) 1992. p. 318-329.

[Czerwinski, Dumais, Robertson et al. 1999] Czerwinski, Mary P.; Dumais, Susan T.; Robertson, George G. et al.: *Visualizing Implicit Queries for Information Management and Retrieval.* In: Williams, Marian G.; Altom, Mark W.; Ehrlich, Kate et al. (Eds.): *CHI 1999: Conference Proceedings Human Factors in Computing Systems. Conference: Pittsburgh, PA, May 17-20 1999.* New York (ACM Press) 1999. p. 560-567.

[Dempsey, Vreeland, Sumner et al. 2000] Dempsey, Bert J.; Vreeland, Robert C.; Sumner, Robert G. et al.: *Design and empirical evaluation of search software for legal professionals on the WWW.* In: *Information Processing & Management*, 36 (2000) 2, p. 253-273.

[Dieberger 1994] Dieberger, Andreas: *Navigation in Spatial Information Environments: User Interface Design Issues for Hypertext and VR Systems.* In: *ECHT 94: European Conference on Hypertext Technology. Conference: September 19-23 1994, Edinburgh, Scotland.* http://www.mindspring.com/~juggle5/Archive/ECHT94.Poster.handout.html [2001-01-03]

[Dieberger, Frank 1998] Dieberger, Andreas; Frank, Andrew U.: *A city metaphor for supporting navigation in complex information spaces.* In: *Journal of Visual Languages and Computing*, 9 (1998) 6, p. 597-622.

[Dieberger, Russell 2001] Dieberger, Andreas; Russell, Daniel M.: *Context Lenses - Document Visualization and Navigation Tools for Rapid Access to Detail.* In: Hirose, Michitaka (Ed.): *Human-Computer Interaction - INTERACT'01. Proceedings. Conference: Tokyo, Japan, 2001 July 9-13* Amsterdam (IOS Press) 2001. S. 545-552.

[Domik, Gutkauf 1994] Domik, Gitta O.; Gutkauf, Bernd: *User modeling for adaptive visualization systems.* In: Bergeron, Daniel; Kaufman, Arie E. (Eds.): *Proceedings of IEEE Visualization 1994. Conference: Washington, DC, October 17-21 1994.* Los Alamitos, CA (IEEE Computer Soc. Press) 1994. p. 217-223.

[Eastman 1999] Eastman, Caroline M.: *30,000 Hits May Be Better Than 300: Precision Anomalies in Internet Searches.* In: Hearst, Marti A. (Ed.): *SIGIR 1999: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval Conference: Berkeley, CA, August 15-19 1999.* New York (ACM Press) 1999. p. 313-314.

[Egan 1988] Egan, Dennis E.: *Individual differences in human-computer interaction.* In: Helander, Martin (Ed.): *Handbook of Human-Computer Interaction* Amsterdam (North-Holland) 1988. p. 543-568.

[Egan, Remde, Gomez et al. 1989] Egan, Dennis E.; Remde, Joel R.; Gomez, Louis M.: *Formative Design-Evaluation of SuperBook.* In: *ACM Transactions on Information Systems*, 7 (1989) 1, p. 30-57.

[Eibl 1999] Eibl, Maximilian: *Visualisierung der Recherche in Dokument-Retrievalsystemen: Methoden und Lösungen.* In: *Nachrichten für Dokumentation*, 50 (1999) 2, p. 79-86.

[Eibl 1999a] Eibl, Maximilian: *Unterstützung der Recherche im Dokumentenretrieval durch den Einsatz von Visualisierung.* In: Heuer, Andreas (Ed.): *ADI 1999: Agenten, Datenbanken und Information Retrieval. Conference: Rostock-Warnemünde, Germany, September 30 - November 1 2000.* http://wwwdb.informatik.uni-rostock.de/adi99/paper/eibl.ps.gz [2001-01-28]

[Eick 1994] Eick, Stephen G.: *Graphically displaying text.* In: *Journal of Computational and Graphical Statistics*, 3 (1994) 2, p. 127-142.

[Eick 2000] Eick, Stephen G.: *Visual Discovery and Analysis.* In: *IEEE Transactions on Visualization and Computer Graphics*, 6 (2000) 1, p. 44-58.

[Eick, Steffen, Sumner 1992] Eick, Stephen G.; Steffen, Joseph L.; Sumner, Eric E.: *SeeSoft: A tool for visualizing line oriented-software statistics* In: *IEEE Transactions on Software Engineering*, 18 (1992) 11, p. 957-968.

[Eick, Wills 1995] Eick, Stephen G.; Wills, Graham J.: *High interaction graphics.* In: *European Journal of Operational Research*, 81 (1995) 3, p. 445-459.

[Ellis 1989] Ellis, David: *A behavioural approach to information retrieval system design.* In: *Journal of Documentation*, 45 (1989) 3, p. 171-212.

[Elzer, Krohn 1997] Elzer, Peter F.; Krohn, Uwe: *Visualisierung zur Unterstützung der Suche in komplexen Datenbeständen.* In: Fuhr, Norbert; Dittrich, Gisbert; Tochtermann, Klaus (Eds.): *Hypertext - Information Retrieval - Multimedia 97: Theorien, Modelle und Implementierungen integrierter elektronischer Informationssysteme. Conference: Dortmund, Germany, October 29 - November 2 1997.* Konstanz (Universitätsverlag Konstanz) 1997. p. 27-38.

[Fairchild, Poltrock, Furnas 1988] Fairchild, Kim M.; Poltrock, Steven E.; Furnas, George W.: *SemNet: Three-dimensional Representations of Large Knowledge Bases.* In: Guindon, Raymonde (Ed.): *Cognitive Science and its Applications for Human-Computer Interaction.* Hillsdale, NJ (Lawrence Erlbaum) 1988. p. 201-233.

[Fishkin, Stone 1995] Fishkin, Ken; Stone, Maureen C.: *Enhanced Dynamic Queries via Movable Filters.* In: Katz, Irvin R.; Mack, Robert L.; Marks, Linn et al. (Eds.): *CHI 1995: Conference Proceedings Human Factors in Computing Systems. Conference: Denver, CO, May 7-11 1995.* New York (ACM Press) 1995. p. 23-29.

[Fowler, Fowler, Wilson 1991] Fowler, Richard H.; Fowler, Wendy A.; Wilson, Bradley A.: *Integrating Query, Thesaurus, and Documents through a Common Visual Representation.* In: Bookstein, Abraham; Chiaramella, Yves; Salton, Gerard et al. (Eds.): *SIGIR 1991: Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Conference: Chicago, IL, October 13-16 1991.* New York (ACM Press) 1991. p. 142-151.

[Fowler, Wilson, Fowler 1992] Fowler, Richard H.; Wilson, Bradley A.; Fowler, Wendy A.: *Information Navigator: An information system using associative networks for display and retrieval.* Department of Computer Science, Technical Report NAG9-551, #92-1.

[Furnas 1981] Furnas, George W.: *The FISHEYE View: A New Look at Structured Files.* Murray Hill, NJ (AT&T Bell Laboratories) 1981.

[Furnas 1986] Furnas, George W.: *Generalized fisheye views* In: Mantei, M.; Orbeton, P. (Eds.): *CHI 1986: Conference Proceedings Human Factors in Computing Systems. Conference: Boston, MA, April 13-17 1986.* New York (ACM Press) 1986. p. 16-23.

[Furnas, Bederson 1995] Furnas, George W.; Bederson, Benjamin B.: *Space-Scale Diagrams: Understand Multiscale Interfaces.* In: Katz, Irvin R.; Mack, Robert L.; Marks, Linn et al. (Eds.): *CHI 1995: Conference Proceedings Human Factors in Computing Systems. Conference: Denver, CO, May 7-11 1995.* New York (ACM Press) 1995. p. 234-241.

[Gershon, LeVasseur, Winstead et al. 1995] Gershon, Nahum; LeVasseur, Joshua; Winstead, Joel et al.: *Visualizing Internet Resources.* In: Gershon, Nahum; Eick, Stephen G. (Eds.): *Proceedings of IEEE Information Visualization 1995. Conference: Atlanta, GA, October 30-31 1995.* Los Alamitos, CA (IEEE Computer Soc. Press) 1995. p. 122-128.

[Gershon, Winstead, LeVasseur et al. 1995] Gershon, Nahum; Winstead, Joel; LeVasseur, Joshua et al.: *Visualizing World Wide Web Information Resources.* In: *Conference on Information and Knowledge Management. Workshop on New Paradigms in Information Visualization and Manipulation. Baltimore, MD, December 2 1995.* http://ifib41.ifib.uni-karlsruhe.de/hyperd/zb/mitre/mitre.html [2001-03-03].

[Ginsburg, Marks, Shieber 1996] Ginsburg, Adam; Marks, Joe; Shieber, Stuart: *A Viewer for PostScript Documents.* In: *UIST 96: 9th ACM Symposium on User Interface Software and Technology* New York (ACM Press) 1996. p. 31-32.

[Goldstein, Roth 1994] Goldstein, Jade; Roth, Steven F.: *Using aggregation and dynamic queries for exploring large data sets.* In: Adelson, B.; Dumais, S.; Olson, J. S. (Eds.): *CHI 1994: Conference Proceedings Human Factors in Computing Systems. Conference: Boston, MA, April 24-28 1994.* New York (ACM Press) 1994. p. 23-29.

[Golovchinsky 1997] Golovchinsky, Gene: *Queries? Links? Is there a Difference?* In: *CHI 1997: Conference Proceedings Human Factors in Computing Systems. Conference: Atlanta, GA, March 22-27 1997.* New York (ACM Press) 1997. p. 407-413.

[Golovchinsky 1997a] Golovchinsky 1997a: *From Information Retrieval to Hypertext and Back Again: The Role of Interaction in the Information Exploration Interface.* University of Toronto, Graduate Department of Mechanical and Industrial Engineering, PhD Thesis 1997

[Golovchinsky 1997b] Golovchinsky, Gene: *What the Query Told the Link: The Integration of Hypertext and Information Retrieval.* In: Bernstein, Mark; Carr, Les; Østerbye, Kasper (Eds.): *Hypertext 1997: Proceedings of the Eigth ACM Conference on Hypertext. Conference: University of Southampton, United Kingdom, April 6-11 1997.* New York (ACM Press) 1997. S. 67-74.

[Golovchinsky, Chignell 1997] Golovchinsky, Gene; Chignell, Mark H.: *The Newspaper as an Information Exploration Metaphor* In: *Journal of Information Processing & Management*, 33 (1997) 5, p. 663-683.

[Grewal, Burden, Jackson et al. 1999] Grewal, Ratvinder S.; Burden, Peter; Jackson, Mike et al.: *A novel interface for representing search-engine results.* In: *IEE Informatics Colloquium: Lost in the Web - Navigation on the Internet. Conference: London, United Kingdom, November 2 1999.* p. 7/1-7/10. http://seed.scit.wlv.ac.uk/papers/lostintheweb99.html [2001-02-05]

[Grewal, Jackson, Burden et al. 2000] Grewal, Ratvinder S.; Jackson, Mike; Burden, Peter et al.: *A Visual Technique to indicate Overall and Component Relevancy of Search-Engine Results.* In: Tjoa, A Min; Wagner, Roland R.; Al-Zobaidie, Ala (Eds.): *Proceedings 11th International Workshop on Database and Expert Systems Applications. Conference: Greenwich, London, United Kingdom, September 4-8 2000.* Los Alamitos, CA (IEEE Computer Society) 2000. p. 608-612.

[Grewal, Jackson, Wallis et al. 1999] Grewal, Ratvinder S.; Jackson, Mike; Wallis, Jon et al.: *Using Visualisation to Interpret Search Engine Results.* In: *Proceeding of The Active Web. Conference: Staffordshire University, United Kingdom, January 1999.* p. 15-25. http://seed.scit.wlv.ac.uk/papers/activeweb99.html [2001-02-05]

[Guastello, Traut, Korienek 1989] Guastello, Stephen J.; Traut, Mary; Korienek, Gene: *Verbal versus Pictorial Representations of Objects in a Human-Computer Interface.* In: *International Journal of Man-Machine Studies*, 31 (1989) 1, p. 99-120.

[Guinan, Smeaton 1992] Guinan, Catherine; Smeaton, Alan F.: *Information Retrieval from Hypertext using Dynamically Planned Guided Tours.* In: *Hypertext 1992 / ECHT 1992: European Conference on Hypertext Technology. Conference: Milan, Italy, November 30 - December 4 1992.* New York (ACM Press) 1992. p. 122-130.

[GVU 1994 – 1998] http://www.cc.gatech.edu/gvu/user_surveys/ [2000-12-30]

[Hammond, Allinson 1987] Hammond, Nick V.; Allinson, Lesley J.: *The travel Metaphor as design principle and training aid for navigating around complex systems.* In: Diaper, D.; Winder, R. (Eds.): *HCI ,87, People and Computers III* (=British Computer Society Conference Series.) p. 75-90.

[Hascoët 1998] Hascoët, Mountaz: *Analytical versus empirical evaluation of spatial displays.* In: Karat, Clare-Marie; Karat, John; Horrocks, Ian (Eds.): *CHI 1998: Conference Proceedings Human Factors in Computing Systems. Conference: Los Angeles, CA, April 21-23 1998.* New York (Addison-Wesley) 1998. p. 257-259.

[Havre, Hetzler, Nowell 2000] Havre, Susan; Hetzler, Beth; Nowell, Lucy T.: *ThemeRiver: Visualizing Theme Changes over Time.* In: Roth, Steven F.; Keim, Daniel A. (Eds.): *Proceedings of IEEE Information Visualization 2000. Conference: Salt Lake City, UT, October 9-10 2000.* Los Alamitos, CA (IEEE Computer Soc. Press) 2000. p. 115-123.

[Hawking, Craswell, Thistlewaite et al. 1999] Hawking, David; Craswell, Nick; Thistlewaite, Paul et al.: *Results and Challenges in Web Search Evaluation.* In: *WWW 8: Eighth International World Wide Web Conference. Conference: Toronto, Canada, May 11-14 1999.* http://www8.org/w8-papers/2c-search-discover/results/results.html [1998-08-17]

[He, Göker 2000] He, Daqing; Göker, Ayse: *Detecting session boundaries from Web user logs.* In: *IRSG 2000: Proceedings of the 22nd Annual Colloquium on Information Retrieval Research. Conference: Cambridge, United Kingdom, April 5-7 2000.* http://irsg.eu.org/irsg2000online/papers/source/he.pdf [2000-12-10].

[Hearst 1995] Hearst, Marti A.: *TileBars: Visualization of Term Distribution Information in Full Text Information Access.* In: Katz, Irvin R.; Mack, Robert L.; Marks, Linn et al. (Eds.): *CHI 1995: Conference Proceedings Human Factors in Computing Systems. Conference: Denver, CO, May 7-11 1995.* New York (ACM Press) 1995. p. 59-66.

[Hearst 1999] Hearst, Marti A.: *User Interfaces and Visualization.* In: Baeza-Yates, Ricardo A.; Ribeiro-Neto, Berthier (Eds.): *Modern Information Retrieval.* Reading, MA (Addison Wesley Longman) 1999. p. 257-324.

[Hearst, Karadi 1997] Hearst, Marti A.; Karadi, Chandu: *Cat-a-Cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy.* In: Belkin, Nicholas J.; Narasimhalu, A. Desai; Willett, Peter (Eds.): *SIGIR 1997: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Conference: Philadelphia, PA, July 27-31 1997.* New York (ACM Press) 1997. p. 246-255.

[Hearst, Pedersen 1996a] Hearst, Marti A.; Pedersen, Jan O.: *Reexamining the cluster hypothesis: Scatter/gather on retrieval results.* In: Frei, Hans-Peter; Harman, Donna K.; Schäuble, Peter et al. (Eds.): *SIGIR 1996: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Conference: Zürich, Switzerland, August 18 -22 1996.* New York (ACM Press) 1996. p. 76-84.

[Hearst, Pedersen, Pirolli et al. 1995] Hearst, Marti A.; Pedersen, Jan O.; Pirolli, Peter L.: *Four TREC-4 Tracks: The Xerox Site Report.* In: Harman, Donna K. (Ed.): *Fourth Text REtrieval Conference (TREC-4). Conference: Gaithersburg, MD, November 1-3 1995.* Gaithersburg, MD (National Institute of Standards and Technology) 1995. ftp://parcftp.xerox.com/pub/hearst/trec4.ps [1999-03-23]

[Heidorn, Cui 2000] Heidorn, P. Bryan; Cui, Hong: *The Interaction of Result Set Display Dimensionality and Cognitive Factors in Information Retrieval Systems.* In: Kraft, Donald H. (Ed.): *ASIS 2000: Proceedings of the 63 rd Annual Meeting (Vol. 37). Conference: Chicago, IL, November 13 - 16 2000.* Medford, NY (Information Today) 2000. http://bibiana.lis.uiuc.edu/~heidorn/heidorn3.pdf [2000-07-28]

[Hemmje 1993a] Hemmje, Matthias: *Eine inhaltsorientierte, intuitive 3D-Benutzerschnittstelle für Information-Retrieval-Systeme.* In: Knorz, Gerhard; Krause, Jürgen; Womser-Hacker, Christa (Eds.): *Information Retrieval '93: Von der Modellierung zur Anwendung, Proceedings der 1. Tagung Information Retrieval '93. Conference: Regensburg, Germany, September 13-15 1993.* Konstanz (Universitätsverlag Konstanz) 1993. p. 82-99.

[Hemmje, Kunkel, Willett 1994] Hemmje, Matthias; Kunkel, Clemens; Willett, Alexander: *LyberWorld - A Visualization User Interface Supporting Fulltext Retrieval.* In: Croft, W. Bruce; van Rijsbergen, Cornelis J. (Eds.): *SIGIR 1994: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval Conference: Dublin, Ireland, July 3-6 1994.* New York (ACM Press) 1994. p. 249-259.

[Henderson, Card 1986] Henderson, D. Austin; Card, Stuart K.: *Rooms: The Use of Multiple Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface.* In: *ACM Transactions on Graphics*, 5 (1986) 3, p. 211-243.

[Hendley, Drew, Wood et al. 1995] Hendley, Robert J.; Drew, Nick S.; Wood, Andrew M.: *Narcissus: Visualizing Information.* In: Gershon, Nahum; Eick, Stephen G. (Eds.): *Proceedings of IEEE Information Visualization 1995. Conference: Atlanta, GA, October 30-31 1995.* Los Alamitos, CA (IEEE Computer Soc. Press) 1995. p. 90-96.

[Henninger, Belkin 1996] Henninger, Scott; Belkin, Nicholas J.: *Interface Issues and Interaction Strategies for Information Retrieval Systems.* In: Tauber, M. J.; Bellotti, V.; Jeffries, R. et al. (Eds.): *CHI 1996: Conference Proceedings Human Factors in Computing Systems. Conference: Vancouver, BC, Canada, April 13-18 1998.* New York (ACM Press) 1996. p. 352-353.

[Heo, Morse, Willms et al. 1996] Heo, Misook; Morse, Emile L.; Willms, Swantje et al.: *Multi-level Navigation of a Document Space.* In: *Proceedings of WebNet 1996 - World Conference on the WWW, Internet & Intranet Conference. San Francisco, CA, October 15-19 1996.* CD-ROM

[Hertzum, Frøkjær 1996] Hertzum, Morten; Frokjaer, Erik: *Browsing and querying in online documentation: A study of user interfaces and the interaction process.* In: *ACM Transactions on Computer Human Interaction*, 3 (1996) 2, p. 136-161.

[Higgins, Lucas, Senn 1999] Higgins, Michael; Lucas, Peter; Senn, Jeffrey: *VisageWeb: Visualizing WWW Data in Visage.* In: Will, Graham; Keim, Daniel A. (Eds.): *Proceedings of IEEE Information Visualization 1999. Conference: San Francisco, CA, October 24-29 1999.* Los Alamitos, CA (IEEE Computer Soc. Press) 1999. p. 100-107.

[Hightower, Ring, Helfman et al. 1998] Hightower, Ron R.; Ring, Laura T.; Helfman, Jonathan I.: *Graphical multiscale Web histories: A study of padprints.* In: *Hypertext 1998. Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia. Conference: Pittsburgh, PA, June 20-24 1998.* New York (ACM Press) 1998. p. 58-65.

[Hölscher 1998] Hölscher, Christoph: *How Internet Experts Search For Information On The Web.* In: Maurer, Hermann; Olson, Richard O. (Eds.): *Proceedings of WebNet 1998 - World Conference on the WWW, Internet & Intranet Conference: Orlando, FL, November 7-12 1998.* CD-ROM

[Hölscher 1998a] Hölscher, Christoph: *Personal Communication.* [Jansen, Pooch 2000] *cite a number of facts from* [Hölscher 1998] *which are not included in the two page WebNet'98-paper and the 13 slides WebNet'98-presentation from Hölscher. Hölscher reported these facts in his talk and included them in a document send to Jansen. This document had also been made available to the author and is here referenced as* [Hölscher 1998a].

[Hölscher 2000] Hölscher, Christoph: *Informationssuche im Internet: Web-Expertise und Wissenseinflüsse.* Dissertation Albert-Ludwigs-Universität Freiburg, Institut für Informatik und Gesellschaft. IIG-Berichte ; 2000,2

[Hölscher, Strube 1999] Hölscher, Christoph; Strube, Gerhard: *Searching on the Web: Two Types of Expertise* In: Hearst, Marti A. (Ed.): *SIGIR 1999: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval Conference: Berkeley, CA, August 15-19 1999.* New York (ACM Press) 1999. p. 305-306.

[Hölscher, Strube 2000] Hölscher, Christoph; Strube, Gerhard: *Web Search Behavior of Internet Experts and Newbies.* In: *WWW 9: 9th International World Wide Web Conference. Conference: Amsterdam, Netherlands, May 15-19 2000.* http://www9.org/w9cdrom/81/81.html [2000-09-09].

[IBM 2000] http://www.software.ibm.com/data/pubs/papers/bisolution/index.html [2000-01-18]

[Industry USability Reporting project 1999] IUSR Industry USability Reporting project: *Common Industry Format for Usability Test Reports. Version 1.1* WWW 1999.

[Inselberg 1985] Inselberg, Alfred: *The plane with parallel coordinates.* In: *The Visual Computer*, 1 (1985), p. 69-91.

[ISO 9241-11] ISO/DIS 9241-11.2 1997 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability.

[ISO 13407] ISO/DIS 13407 Human Centred Design for Interactive Systems.

[Jacob 1989] Jacob, Robert J.: *Direct Manipulation In The Intelligent Interface.* In: Hancock, Peter A.; Chignell, Mark H. (Eds.): *Intelligent Interfaces: Theory, Research and Design.* Amsterdam (North Holland) 1989. p. 165-212.

[Jansen, Pooch 2000] Jansen, Bernard J.; Pooch, Udo: *A Review of Web Searching Studies and a Framework for Future Research. Submitted to the Journal of the American Society of Information Management.* http://jimjansen.tripod.com/academic/pubs/wus.html [2000-10-01]

[Jansen, Spink, Bateman et al. 1998] Jansen, Bernard. J.; Spink, Amanda; Bateman, Judy et al.: *Real Life Information Retrieval: a Study of User Queries on the Web.* In: *ACM SIGIR Forum*, 32 (1998) 1, p. 5-17.

[Jansen, Spink, Bateman et al. 1998a] Jansen, Bernard. J.; Spink, Amanda; Bateman, Judy et al.: *Searchers, the Subjects They Search, and Sufficiency: A Study of a Large Sample of EXCITE Searches.* Aus: Maurer, Hermann; Olson, Richard O. (Eds.): *Proceedings of WebNet 1998 - World Conference on the WWW, Internet & Intranet Conference: Orlando, FL, November 7-12 1998.* CD-ROM.

[Jansen, Spink, Saracevic 1998] Jansen, Bernard. J.; Spink, Amanda; Saracevic, Tefko: *Failure Analysis in Query Construction: Data and Analysis from A Large Sample of Web Queries.* In: Witten, Ian H.; Akscyn, Robert M.; Shipman, Frank M. (Eds.): *ACM Digital Libraries 1998: Proceedings of the 3rd ACM International Conference on Digital Libraries. Conference: Pittsburgh, PA, June 23-26 1998.* New York (ACM Press) 1998. p. 289-290.

[Jansen, Spink, Saracevic 2000] Jansen, Bernard. J.; Spink, Amanda; Saracevic, Tefko: *Real life, real users, and real needs: a study and analysis of user queries on the Web.* In: *Information Processing & Management*, 36 (2000) 2, p. 207-227.

[Jerding, Stasko 1995] Jerding, Dean F.; Stasko, John T.: *The Information Mural: A technique for displaying and navigating large information spaces.* In: Gershon, Nahum; Eick, Stephen G. (Eds.): *Proceedings of IEEE Information Visualization 1995. Conference: Atlanta, GA, October 30-31 1995.* Los Alamitos, CA (IEEE Computer Soc. Press) 1995. p. 43-50.

[Jerding, Stasko 1997] Jerding, Dean F.; Stasko, John T.: *The Information Mural: A technique for displaying and navigating large information spaces.* Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, GA, Technical Report GIT-GVU-97-24, December 1997

[Jog, Shneiderman 1995] Jog, Ninad K.; Shneiderman, Ben: *Starfield Information Visualization with Interactive Smooth Zooming.* Aus: Spaccapietra, Stefano; Jain, Ramesh (Eds.): *Visual Database Systems 3, Visual Information Management, Proceedings of the third IFIP 2.6 working conference on visual database systems. Conference: Lausanne, Switzerland, March 27-29 1995.* London (Chapman & Hall) 1995. p. 3-14.

[Jones 1998] Jones, Steve: *Graphical Query Specification and Dynamic Results Preview for a Digital Library.* In: *UIST 98: 11th ACM Symposium on User Interface Software and Technology* New York (ACM Press) 1998. p. 143-151.

[Jones 1998a] Jones, Steve: *Dynamic Query Result Previews for a Digital Library.* In: Witten, Ian H.; Akscyn, Robert M.; Shipman, Frank M. (Eds.): *ACM Digital Libraries 1998: Proceedings of the 3rd ACM International Conference on Digital Libraries. Conference: Pittsburgh, PA, June 23-26 1998.* New York (ACM Press) 1998. p. 291-292.

[Jones, McInnes, Staveley 1999] Jones, Steve; McInnes, Shona; Staveley, Mark S.: *A Graphical User Interface for Boolean Query Specification.* In: *International Journal on Digital Libraries*, 2 (1999) 2/3, p. 207-223.

[Kaasten, Greenberg 2000] Kaasten, Shaun; Greenberg, Saul: *Designing an Integrated Bookmark / History System for Web Browsing.* In: *Proceedings of the Western Computer Graphics Symposium 2000. Conference: Panorama Mountain Village, BC, Canada, March 26-29 2000.* http://www.cpsc.ucalgary.ca/grouplab/papers/2000/00-GrouplabPapers.Skigraph/kaasten/kaasten.pdf [2001-02-07].

[Kaasten, Greenberg 2001] Kaasten, Shaun; Greenberg, Saul: *Integrating Back, History and Bookmarks in Web Browsers.* http://www.cpsc.ucalgary.ca/grouplab/papers/2000/00-IntegratedHistory.Report/00-IntegratedHistory-RR.pdf [2001-02-07]

[Kamba, Bharat, Albers 1995] Kamba, Tomonari; Bharat, Krishna; Albers, Michael C.: *The Krakatoa Chronicle - An Interactive, Personalized NewsPaper on the Web.* Georgia Tech Technical Report GIT-GVU-95-25.

[Kaugars 1998] Kaugars, Karlis J.: *Integrated multi scale text retrieval visualization.* In: Karat, Clare-Marie; Karat, John; Horrocks, Ian (Eds.): *CHI 1998: Conference Proceedings Human Factors in Computing Systems. Conference: Los Angeles, CA, April 21-23 1998.* New York (Addison-Wesley) 1998. p. 307-308.

[Keim, Kriegel 1994] Keim, Daniel A.; Kriegel, Hans-Peter: *VisDB: Database exploration using multidimensional visualization* In: *IEEE Computer Graphics and Applications*, 14 (1994) 5, p. 40-49.

[Kirsch 1998] Kirsch, Steven T.: *Infoseek's experiences searching the Internet.* In: *SIGIR Forum*, 32 (1998) 2, p. 3-7.

[Kleiboemer, Lazear, Pedersen 1996] Kleiboemer, Adrienee J.; Lazear, Manette B.; Pedersen, Jan O.: *Tailoring a Retrieval System for Naive Users.* In: Baird, Henry S. (Ed.): *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval (SDAIR) Conference: Las Vegas, NV, April 15-17 1996.* p. 209-216.

[Koenemann, Belkin 1996] Koenemann, Jürgen; Belkin, Nicholas J.: *A Case For Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness.* In: Tauber, M. J.; Bellotti, V.; Jeffries, R. et al. (Eds.): *CHI 1996: Conference Proceedings Human Factors in Computing Systems. Conference: Vancouver, BC, Canada, April 13-18 1998.* New York (ACM Press) 1996. p. 205-212.

[Kohonen 1998] Kohonen, Teuvo: *Self-organization of very large document collections: State of the art.* In: Niklasson, Lars; Bodén, Mikael; Ziemke, Tom (Eds.): *Proceedings of ICANN98, the 8th International Conference on Artificial Neural Networks. Conference: Skövde, Sweden, 1998 September 2-4* Springer (London) 1998. p. 65-74, vol. 1.

[Körber 2000] Körber, Sven: *Suchmuster erfahrener und unerfahrener Suchmaschinennutzer im deutschsprachigen World Wide Web. Ein Experiment.* Unveröffentlichte Magisterarbeit, Westfälische Wilhelms-Universität Münster. http://kommunix.uni-muenster.de/IfK/examen/koerber/suchmuster.pdf, [2001-01-12]

[Korfhage 1991] Korfhage, Robert R.: *To See or Not to See - Is That the Query?* In: Bookstein, Abraham; Chiaramella, Yves; Salton, Gerard et al. (Eds.): *SIGIR 1991: Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Vol 1. Conference: Chicago, IL, October 13-16 1991.* New York (ACM Press) 1991. p. 134-141.

[Krohn 1995] Krohn, Uwe: *Visualization of navigation retrieval in virtual information spaces.* In: CIKM 1995: *Conference on Information and Knowledge Management. Workshop on New Paradigms in Information Visualization and Manipulation. Conference: Baltimore, MD, December 2 1995.* http://www.cs.umbc.edu/~cikm/1995/npiv/uwe/paper.ps [1999-01-13]

[Lamping, Rao 1996] Lamping, John; Rao, Ramana: *Visualizing Large Trees Using the Hyperbolic Browser.* In: Tauber, M. J.; Bellotti, V.; Jeffries, R. et al. (Eds.): *CHI 1996: Conference Proceedings Human Factors in Computing Systems. Conference: Vancouver, BC, Canada, April 13-18 1998.* New York (ACM Press) 1996. p. 388-389

[Lamping, Rao, Pirolli 1995] Lamping, John; Rao, Ramana; Pirolli, Peter L.: *A focus + context technique based on hyberbolic geometry for visualizing large hierarchies.* In: Katz, Irvin R.; Mack, Robert L.; Marks, Linn et al. (Eds.): *CHI 1995: Conference Proceedings Human Factors in Computing Systems. Conference: Denver, CO, May 7-11 1995.* New York (ACM Press) 1995. p. 401-408.

[Larkin, Simon 1987] Larkin, Jill H.; Simon, Herbert A.: *Why a diagram is (sometimes) worth 10,000 words.* In: *Cognitive Science*, 11 (1987) 1, p. 65-99.

[Lau, Horvitz 1999] Lau, Tessa; Horvitz, Eric: *Patterns of Search: Analyzing and Modeling Web Query Dynamics.* In: *Proceedings of the Seventh International Conference User Modeling Conference: Banff, Canada, June 1999.* New York (Springer) 1999. p. 119-128.

[Lazonder, Biemans, Wopereis 2000] Lazonder, Ard W.; Biemans, Harm J.; Wopereis, Iwan G.: *Differences between Novice and Experienced Users in Searching Information on the World Wide Web.* In: *Journal of the American Society for Information Science (JASIS)*, 51 (2000) 6, p. 576-581.

[Leouski, Allan 1998] Leouski, Anton V.; Allan, James: *Visual interactions with a multidimensional ranked list.* In: Croft, W. Bruce et al. (Eds.): *SIGIR 1998: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval Conference: Melbourne, Australia, August 24-28 1998.* New York (ACM Press) 1998. p. 353-354.

[Leouski, Allan 1998a] Leouski, Anton V.; Allan, James: *Evaluating a Visual Navigation System for a Digital Library.* In: Nikolaou, Christos; Stephanidis, Constantine (Eds.): *Research and Advanced Technology for Digital Libraries, Second European Conference, ECDL 98. Conference: Heraklion, Crete, Greece, September 21-23 1998.* Berlin (Springer) 1998. p. 535-554.

[Leung, Apperley 1994] Leung, Ying K.; Apperley, Mark D.: *A review and taxonomy of distortion-oriented presentation techniques.* In: *ACM Transactions on Computer Human Interaction*, 1 (1994) 2, p. 126-160.

[Lin 1995] Lin, Xia: *Searching and browsing on map displays.* In: Kinney, Thomas (Ed.): *ASIS 1995: Proceedings of the 58th Annual Meeting (Vol. 32). Conference: Chicago, IL, October 9 - 12 1995.* Medford, NY (Information Today) 1995. p. 13-18.

[Lin, Soergel, Marchionini 1991] Lin, Xia; Soergel, Dagobert; Marchionini, Gary: *A Self-Organizing Semantic Map for Information Retrieval.* In: Bookstein, Abraham; Chiaramella, Yves; Salton, Gerard et al. (Eds.): *SIGIR 1991: Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Conference: Chicago, IL, October 13-16 1991.* New York (ACM Press) 1991. p. 262-269.

[Mackinlay 1986] Mackinlay, Jock D.: *Automating the design of graphical presentations of relational information.* In: *ACM Transactions on Graphics*, 5 (1986) 2, p. 110-141.

[Mackinlay, Rao, Card 1995] Mackinlay, Jock D.; Rao, Ramana; Card, Stuart K.: *An organic user interface for searching citation links.* In: Katz, Irvin R.; Mack, Robert L.; Marks, Linn et al. (Eds.): *CHI 1995: Conference Proceedings Human Factors in Computing Systems. Conference: Denver, CO, May 7-11 1995.* New York (ACM Press) 1995. p. 67-73.

[Mackinlay, Robertson, Card 1991] Mackinlay, Jock D.; Robertson, George G.; Card, Stuart K.: *The Perspective Wall: Detail and Context Smoothly Integrated.* In: Robertson, S. P.; Olson, G. M.; Olson, J. S. (Eds.): *CHI 1991: Conference Proceedings Human Factors in Computing Systems. Conference: New Orleans, LA, April 27 - May 2 1991.* New York (ACM Press) 1991. p. 173-179.

[Mander, Salomon, Wong 1992] Mander, Richard; Salomon, Gitta; Wong, Yin Yin: *'Pile' metaphor for supporting casual organization of information.* In: Bauersfeld, P.; Bennett, J.; Lynch, G. (Eds.): *CHI 1992: Conference Proceedings Human Factors in Computing Systems. Conference: Monterey, CA, May 3-7 1992.* New York (ACM Press) 1992. p. 627-634.

[Mann 1999] Mann, Thomas M.: *Visualization of WWW-Search Results.* In: Cammelli, Antonio; Tjoa, A Min; Wagner, Roland R. (Eds.): *Proceedings Tenth International Workshop on Database and Expert Systems Applications. Conference: Florence, Italy, September 1-3 1999.* Los Alamitos, CA (IEEE Computer Society) 1999. p. 264-268.

[Mann, Reiterer 1999] Mann, Thomas M.; Reiterer, Harald: *Case Study: A Combined Visualization Approach for WWW-Search Results.* In: *IEEE Information Visualization Symposium 1999 Late Breaking Hot Topics Proceedings. Supplement to: Will, Graham; Keim, Daniel (Eds.): Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis'99). Conference: San Francisco, CA, October 24-29 1999. Los Alamitos, CA (IEEE Computer Soc. Press) 1999. Handed out to the conferees.* San Francisco 1999. p. 59-62.

[Mann, Reiterer 2000] Mann, Thomas M.; Reiterer, Harald: *Evaluation of Different Visualization of WWW Search Results.* In: Tjoa, A Min; Wagner, Roland R.; Al-Zobaidie, Ala (Eds.): *Proceedings 11th International Workshop on Database and Expert Systems Applications. Conference: Greenwich, London, United Kingdom, September 4-8 2000.* Los Alamitos, CA (IEEE Computer Society) 2000. p. 586-590.

[Marchionini 1992] Marchionini, Gary: *Interfaces for end-user information seeking.* In: *Journal of the American Society for Information Science (JASIS)*, 43 (1992) 2, p. 156-163.

[Marchionini 1997] Marchionini, Gary: *Information Seeking in Electronic Environments.* Cambridge (Cambridge University Press) 1997.

[McCrickard, Kehoe 1997] McCrickard, D. Scott; Kehoe, Colleen M.: *Visualizing Search Results using SQWID.* In: *WWW 6: Sixth International World Wide Web Conference. Conference: Santa Clara, CA, April 7 - 11 1997.* http://www.cc.gatech.edu/grads/m/Scott.McCrickard/sqwid/Doc/www6.html [1999-03-22]

[Meyer, Sit, Spaulding et al. 1997] Meyer, Beth; Sit, Richard A.; Spaulding, Victoria A.: *Age Group Differences in World Wide Web Navigation.* In: *CHI 1997: Conference Proceedings Human Factors in Computing Systems. Conference: Atlanta, GA, March 22-27 1997.* New York (ACM Press) 1997. p. 157-164.

[Mitchell, Day, Hirschman 1995] Mitchell, Richard; Day, David; Hirschman, Lynette: *Fishing for Information on the Internet.* In: Gershon, Nahum; Eick, Stephen G. (Eds.): *Proceedings of IEEE Information Visualization 1995. Conference: Atlanta, GA, October 30-31 1995.* Los Alamitos, CA (IEEE Computer Soc. Press) 1995. p. 105-111.

[Moore, Murray 2000] Moore, Alvin; Murray, Brian H.: *Sizing the Internet. A Cyveillance study.* Cyveillance, Inc. July 10, 2000. Online http://www.cyveillance.com/resources/7921S_Sizing_the_Internet.pdf [2001-01-13]

[Morse 1999] Morse, Emile L.: *Evaluation of Visual Information Browsing Displays.* Dissertation. Graduate Faculty of the Department of Information Science and Telecommunications of the School of Information Sciences. University of Pittsburgh. 1999

[Morse, Lewis 1997] Morse, Emile L.; Lewis, Michael: *Why Information Retrieval Visualizations Sometimes Fail.* In: *Proceedings of IEEE Conference on Systems Man and Cybernetics 1997* http://www.itl.nist.gov/iaui/vvrg/emorse/papers/smc97.html [2001-01-27]

[Morse, Lewis, Korfhage et al. 1998] Morse, Emile L.; Lewis, Michael; Korfhage, Robert R.: *Evaluation of Text, Numeric and Graphical Presentations for Information Retrieval Interfaces: User Preference and Task Performance Measures* In: *Proceedings of IEEE Conference on Systems Man and Cybernetics 1997* http://www.itl.nist.gov/iaui/vvrg/emorse/papers/smc98.html [2000-12-03]

[Morse, Lewis, Olsen 2000] Morse, Emile L.; Lewis, Michael; Olsen, Kai A.: *Evaluating visualizations: using a taxonomic guide.* In: *International Journal of Human-Computer Studies*, 53 (2000) 5, p. 637-662.

[Mukherjea, Foley 1995] Mukherjea, Sougata; Foley, James D.: *Visualizing the World-Wide Web with the Navigational View Builder. Technical report 95-09.* Graphics, Visualisation and Usability Center, Georgia Tech, USA 1995

[Mukherjea, Foley, Hudson 1995] Mukherjea, Sougata; Foley, James D.; Hudson, Scott E.: *Visualizing complex hypermedia networks through multiple hierarchical views.* In: Katz, Irvin R.; Mack, Robert L.; Marks, Linn et al. (Eds.): *CHI 1995: Conference Proceedings Human Factors in Computing Systems. Conference: Denver, CO, May 7-11 1995.* New York (ACM Press) 1995. p. 331-337.

[Mukherjea, Hirata, Hara 1997] Mukherjea, Sougata; Hirata, Kyoji; Hara, Yoshinori: *Towards a Multimedia World Wide Web Information Retrieval Engine.* In: *WWW 6: Sixth International World Wide Web Conference. Conference: Santa Clara, CA, April 7 - 11 1997.* http://www.scope.gmd.de/info/www6/technical/paper003/paper3.html [2000-11-15]

[Munzner 1997] Munzner, Tamara: *H3: laying out large directed graphs in 3D hyperbolic space.* In: *Proceedings of IEEE Information Visualization 1997. Conference: London, United Kingdom, August 27-29 1997.* Phoenix, AZ. (IEEE Computer Soc. Press) 1997. p. 2-10.

[Munzner, Burchard 1995] Munzner, Tamara; Burchard, Paul: *Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space.* In: *Proceedings of the First Annual Symposium on the VRML Modeling language (VRML 95). Special issue of Computer Graphics Conference: San Diego, CA, December 14-15 1995.* New York (ACM SIGGRAPH) 1995. p. 33-38.

[Mußler 2002] Mußler, Gabriela: *Retrieving Business Information from the WWW.* Unversität Konstanz. Computer and Information Science. Dissertation (forthcoming 2002)

[Mußler, Reiterer, Mann 2000] Mußler, Gabriela; Reiterer, Harald; Mann, Thomas M.: *INSYDER - Information Retrieval Aspects of a Business Intelligence System.* In: Knorz, Gerhard; Kuhlen, Rainer (Eds.): *Proceedings des 7. Internationalen Symposiums für Informationswissenschaft. Conference: Darmstadt, Germany, November 8-10 2000.* Konstanz. Germany (UKV Universitätsverlag Konstanz) 2000. p. 127-143.

[Navarro-Prieto, Scaife, Rogers 1999] Navarro-Prieto, Raquel; Scaife, Mike; Rogers, Yvonne: *Cognitive Strategies in Web Searching.* In: *HFWEB 1999: Proceedings of the 5th Conference on Human Factors & the Web* http://zing.ncsl.nist.gov/hfweb/proceedings/navarro-prieto/index.html [2000-10-02]

[Nielsen 1998] Nielsen, Jakob: *2D is Better Than 3D. ... because people are not frogs.* http://www.useit.com/alertbox/981115.html [1998-10-10]

[Nielsen 1998a] Nielsen, Jakob: *Usability engineering.* Boston, MA (AP Professional) 1998.

[Nonogaki, Ueda 1991] Nonogaki, Hajime; Ueda, Hirotada: *FRIEND21 project: A construction of 21st century human interface.* In: Robertson, S. P.; Olson, G. M.; Olson, J. S. (Eds.): *CHI 1991: Conference Proceedings Human Factors in Computing Systems. Conference: New Orleans, LA, April 27 - May 2 1991.* New York (ACM Press) 1991. p. 407-414

[North 1997] http://www.cs.umd.edu/~north/infoviz.html [2001-01-02]

[North 2000] North, Christopher L.: *A User Interface for Coordinating Visualizations based on Relational Schemata: Snap-Together Visualization.* University of Maryland Computer Science Dept. Doctoral Dissertation, May 2000.

[North, Shneiderman 1997] North, Christopher L.; Shneiderman, Ben: *A Taxonomy Of Multiple Window Coordinations.* University of Maryland, College Park, Dept. of Computer Science, Technical Report #CS-TR-3854

[North, Shneiderman 2000] North, Christopher L.; Shneiderman, Ben: *Snap-Together Visualization: A User Interface for Coordinating Visualizations of a Relational Database.* In: Di Gesù, Vito; Levialdi, Stefano; Tarantino, Laura (Eds.): *AVI 2000: International Workshop on Advanced Visual Interfaces. Conference: Palermo, Italy, May 23-26 2000.* New York (ACM Press) 2000. p. 128-135

[North, Shneiderman 2000a] North, Christopher L.; Shneiderman, Ben: *Snap-Together Visualizations: Can Users Construct and Operate Coordinated Views.* In: *International Journal of Human-Computer Studies*, 53 (2000) 5, p. 715-739.

[Nowell, France, Hix et al. 1996] Nowell, Lucy T.; France, Robert K.; Hix, Deborah et al.: *Visualizing Search Results: Some Alternatives to Query-Document Similarity.* In: Frei, Hans-Peter; Harman, Donna K.; Schäuble, Peter et al. (Eds.): *SIGIR 1996: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Conference: Zürich, Switzerland, August 18 -22 1996.* New York (ACM Press) 1996. p. 67-75.

[O'Day, Jeffries 1993] O'Day, Vicki L.; Jeffries, Robin: *Orienteering in an information landscape: how information seekers get from here to there.* In: *CHI 1993: Conference Proceedings Human Factors in Computing Systems.* New York (ACM Press) 1993. p. 438-445.

[Ogden, Davis, Rice 1998] Ogden, William C.; Davis, Mark W.; Rice, Sean: *Document Thumbnail Visualization for Rapid Relevance Judgments: When do They Pay Off?* In: Voorhees, Ellen M.; Harman, Donna K. (Eds.): *NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC-7). Conference: Gaithersburg, MD, November 09-11 1998.* Gaithersburg, MD, USA (National Institute of Standards and Technology) 1998. p. 528-534.

[OLIVE 1997] http://otal.umd.edu/Olive/ [2001-01-02]

[Pearson, Steinmetz 1993] Pearson, Pery; Steinmetz, Joe: *JOVE Project: 3D Tree-Mapped Universes and Data Sphere Navigation, A Method to Increase the Density of Hierarchical Information Visualization in a Finite Display Space.* http://www.hitl.washington.edu/projects/knowledge_base/virtual-worlds/JOVE/Articles/InfoViz/infoviz.txt [1999-01-11]

[Pejtersen 1989] Pejtersen, Annelise M.: *A Library System for Information Retrieval Based on a Cognitive Task Analysis and Supported by an Icon-Based Interface.* In: Belkin, Nicholas J.; van Rijsbergen, Cornelis J. (Eds.): *SIGIR 1989: Proceedings of the 12th International Conference on Research and Development in Information Retrieval. Conference: Cambridge, MA, June 25-28 1989.* New York (ACM Press) 1989. p. 40-47.

[Peters 1993] Peters, Thomas A.: *The History and Development of Transaction Log Analysis.* In: *Library Hi Tech*, 11 (1993) 2, p. 41-66. cited by [Jansen, Pooch 2000]

[Pitkow, Kehoe 1996] Pitkow, James E.; Cehoe, Colleen M.: *Emerging Trends in the WWW user population.* In: *Communications of the ACM*, 39 (1996) 6, p. 106-108.

[Plaisant, Milash, Rose et al. 1996] Plaisant, Catherine; Milash, Brett; Rose, Anne et al.: *LifeLines: Visualizing personal histories* In: Tauber, M. J.; Bellotti, V.; Jeffries, R. et al. (Eds.): *CHI 1996: Conference Proceedings Human Factors in Computing Systems. Conference: Vancouver, BC, Canada, April 13-18 1998.* New York (ACM Press) 1996. p. 221-227, 518.

[Pollock, Hockley 1997] Pollock, Annabel; Hockley, Andrew: *What's Wrong with Internet Searching.* In: *D-Lib Magazine*, (1997) March, http://www.dlib.org/dlib/march97/bt/03pollock.html [1999-03-15]

[Rao, Card 1994] Rao, Ramana; Card, Stuart K.: *The Table Lens. Merging graphical and symbolic representations in an interactive focus + context visualization for tabular information.* In: Adelson, B.; Dumais, S.; Olson, J. S. (Eds.): *CHI 1994: Conference Proceedings Human Factors in Computing Systems. Conference: Boston, MA, April 24-28 1994.* New York (ACM Press) 1994. p. 318-322.

[Rauber, Bina 1999] Rauber, Andreas; Bina, Harald: *A Metaphor Graphics Based Representation of Digital Libraries on the World Wide Web: Using the libViewer to Make Metadata Visible* In: Cammelli, Antonio; Tjoa, A Min; Wagner, Roland R. (Eds.): *Proceedings Tenth International Workshop on Database and Expert Systems Applications. Conference: Florence, Italy, September 1-3 1999.* Los Alamitos, CA (IEEE Computer Society) 1999. p. 286-290.

[Rauber, Bina 2000] Rauber, Andreas; Bina, Harald: *„ 'Andreas Rauber'? Conference Pages Are over There, German Documents on the Lower Left, ...": An „Old-Fashioned" Approach to Web Search Results Visualization.* In: Tjoa, A Min; Wagner, Roland R.; Al-Zobaidie, Ala (Eds.): *Proceedings 11th International Workshop on Database and Expert Systems Applications. Conference: Greenwich, London, United Kingdom, September 4-8 2000.* Los Alamitos, CA (IEEE Computer Society) 2000. p. 615-619.

[Redmond-Pyle, Moore 1995] Redmond-Pyle, David; Moore, Alan: *Graphical User Interface Design and Evaluation (GUIDE). A practical process.* London (Prentice Hall) 1995.

[Reiterer, Mann 2000] Reiterer, Harald; Mann, Thomas M.: *Visual Information Retrieval for the Web.* In: *17th International CODATA Conference. Book of Abstracts. Conference: Baveno, Italy, October 15-19 2000.,* Paris, France (CODATA) 2000. p. 33-34.

[Reiterer, Mußler, Mann 2001] Reiterer, Harald; Mußler, Gabriela; Mann, Thomas M.: *A visual information seeking system for Web search.* In: Oberquelle, Horst; Oppermann, Reinhard; Krause, Jürgen (Eds.): *Mensch & Computer 2001. Conference: Bad Honnef, Germany, March 5-8 2001.* Stuttgart (Teubner) 2001. S. 297-306.

[Reiterer, Mußler, Mann et al. 2000] Reiterer, Harald; Mußler, Gabriela; Mann, Thomas M.: *INSYDER - An Information Assistant for Business Intelligence.* In: Belkin, Nicholas J.; Ingwersen, Peter; Leong, Mun-Kew (Eds.): *SIGIR 2000: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Conference: Athens, Greece, July 24-28 2000.* New York (ACM Press) 2000. p. 112-119.

[Rennison 1994] Rennison, Earl: *Galaxy of News: An Approach to Visualizing and Understanding Expansive News Landscapes.* In: *UIST 94: 7th ACM Symposium on User Interface Software and Technology* New York (ACM press) 1994. p. 3-12.

[Resnick, Iacovou, Sucak et al. 1994] Resnick, Paul; Iacovou, Neophytos; Suchak, Mitesh et al.: *GroupLens: An open architecture for collaborative filtering of netnews.* In: *CSCW 94. Proceedings of the conference on Computer supported cooperative work* New York (ACM press) 1994. p. 175-186.

[Risden, Czerwinski, Munzner et al. 2000] Risden, Kirsten; Czerwinski, Mary P.; Munzner, Tamara: *An initial examination of ease of use for 2D and 3D information visualizations of Web content.* In: *International Journal of Human-Computer Studies*, 53 (2000) 5, p. 695-714.

[Robertson, Card, Mackinlay 1993] Robertson, George G.; Card, Stuart K.; Mackinlay, Jock D.: *Information Visualization Using 3-D Interactive Animation.* In: *Communications of the ACM*, 36 (1993) 4, p. 56-71.

[Robertson, Czerwinski, Larson et al. 1998] Robertson, George G.; Czerwinski, Mary P.; Larson, Kevin et al.: *Data mountain: using spatial memory for document management.* In: *UIST 98: 11th ACM Symposium on User Interface Software and Technology* New York (ACM Press) 1998. p. 153-162.

[Robertson, Mackinlay 1993] Robertson, George G.; Mackinlay, Jock D.: *The Document Lens.* In: *UIST 93: 6th ACM Symposium on User Interface Software and Technology.* New York (ACM press) 1993. p. 101-108.

[Robertson, Mackinlay, Card 1991] Robertson, George G.; Mackinlay, Jock D.; Card, Stuart K.: *Cone Trees: Animated 3D Visualizations of Hierarchical Information.* In: Robertson, S. P.; Olson, G. M.; Olson, J. S. (Eds.): *CHI 1991: Conference Proceedings Human Factors in Computing Systems. Conference: New Orleans, LA, April 27 - May 2 1991.* New York (ACM Press) 1991. p. 189-194.

[Rohrer, Swing 1997] Rohrer, Randall M.; Swing, Edward: *Web-Based Information Visualization* In: *IEEE Computer Graphics and Applications*, 17 (1997) 4, p. 52-59.

[Rose, Mander, Oren et al. 1993] Rose, Daniel E.; Mander, Richard; Oren, Tim et al.: *Content Awareness in a File System Interface: Implementing the ‚Pile' Metaphor for Organizing Information.* In: Korfhage, Robert R.; Rasmussen, Edie M.; Willett, Peter (Eds.): *SIGIR 1993: Proceedings of the 16th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Conference: Pittsburgh, PA, June 27 - July 1 1993.* New York (ACM Press) 1993. p. 260-269.

[Roth, Mattis 1990] Roth, Steven F.; Mattis, Joe: *Data characterization for intelligent graphics presentation.* In: Carrasco Chew, J.; Whiteside, J. (Eds.): *CHI 1990: Conference Proceedings Human Factors in Computing Systems. Conference: Seattle, WA, April 1-5 1990.* New York (ACM Press) 1990. p. 193-200.

[Röttgers 1999] Röttgers, Heike: *Effizienz der Benutzerschnittstellen von Internet-Suchmaschinen.* Hamburg: Diplomarbeit, Fachhochschule Hamburg, Fachbereich Bibliothek und Information 1999

[Sammon 1969] Sammon, John W.: *A nonlinear mapping for data structure analysis.* In: *IEEE Transactions on Computers*, 18 (1969) 5, p. 401-409

[Sarkar, Brown 1992] Sarkar, Manojit; Brown, Marc H.: *Graphical Fisheye Views of Graphs.* Digital SRC Research Report 84a (revised version of Proc. of ACM CHI'92) 1992.

[Schmid, Hinterberger 1994] Schmidt, Claudia; Hinterberger, Hans: *Comparative Multivariate Visualization Across Conceptually Different Graphic Displays.* In: French, James C.; Hinterberger, Hans (Eds.): *Seventh International Working Conference on Scientific and Statistical Database Management.* Charlottesville, VA (IEEE Computer Society Press) 1994. p. 42-51.

[Sebrechts, Vasilakis, Miller et al. 1999] Sebrechts, Marc M.; Vasilakis, Joanna; Miller, Michael p.: *Visualization of Search Results: A Comparative Evaluation of Text, 2D, and 3D Interfaces.* In: Hearst, Marti A. (Eds.): *SIGIR 1999: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval Conference: Berkeley, CA, August 15-19 1999.* New York (ACM Press) 1999. p. 3-10.

[Sen, Boe 1991] Sen, Tarun; Boe, Warren J.: *Confidence and accuracy in judgements using computer displayed information.* In: *Behavior and Information Technology*, 10 (1991) 1, p. 53-64.

[Shneiderman 1982] Shneiderman, Ben: *The Future of Interactive Systems and the Emergence of Direct Manipulation.* In: *Behavior and Information Technology*, 1 (1982) 3, p. 237-256.

[Shneiderman 1987] Shneiderman, Ben: *Designing the User Interface. Strategies for Effective Human-Computer Interaction.* Reading, MA (Addison-Wesley) 1987.

[Shneiderman 1991] Shneiderman, Ben: *Visual User Interfaces for Information Exploration.* In: *Proceeding of the 4th Annual Meeting of The American Society for Information Sciences. Conference: Washington, DC, October 27-31 1991*. p. 379-384.

[Shneiderman 1992] Shneiderman, Ben: *Tree Visualization with Tree-maps: A 2-d space-filling approach.* In: *ACM Transactions on Graphics*, 11 (1992) 1, p. 92-99.

[Shneiderman 1996] Shneiderman, Ben: *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations.* In: *Proceedings of IEEE Symposium on Visual Languages 96 Conference: Boulder, CO, September 3-6 1996.* Los Alamitos, CA (IEEE Computer Soc. Press)1996. p. 336-343.

[Shneiderman 1998] Shneiderman, Ben: *Designing the User Interface. Strategies for Effective Human-Computer Interaction. 3rd edition* Reading, MA (Addison-Wesley) 1998.

[Shneiderman, Byrd, Croft 1997] Shneiderman, Ben; Byrd, Donald; Croft, W. Bruce: *Clarifying Search: A User-Interface Framework for Text Searches.* In: *D-Lib Magazine*, (1997), http://www.dlib.org/dlib/january97/retrieval/01shneiderman.html [1999-08-17]

[Silverstein, Henzinger, Marais et al. 1998] Silverstein, Craig; Henzinger, Monika; Marais, Hannes et al.: *Analysis of a Very Large AltaVista Query Log.* Digital SRC Technical Note 1998-014. 1998 October 26

[Silverstein, Henzinger, Marais et al. 1999] Silverstein, Craig; Henzinger, Monika; Marais, Hannes et al.: *Analysis of a Very Large Web Search Engine Query Log.* In: *SIGIR Forum*, 33 (1999) 1, p. 6-12.

[Spence, Apperley 1982] Spence, Robert; Apperley, Mark D.: *Data base navigation: An office environment for the professional.* In: *Behavior and Information Technology*, 1 (1982) 1, p. 43-54.

[Spenke, Beilken, Berlage 1996] Spenke, Michael; Beilken, Christian; Berlage, Thomas: *FOCUS: The Interactive Table for Product Comparison and Selection.* In: *UIST 96: 9th ACM Symposium on User Interface Software and Technology* New York (ACM Press) 1996. p. 41-50.

[Spink, Bateman, Jansen 1998] Spink, Amanda; Bateman, Judy; Jansen, Bernard. J.: *Searching heterogeneous collections on the Web: behaviour of Excite users.* In: *Information Research*, 4 (1998) 2, http://www.shef.ac.uk/~is/publications/infres/paper53.html [1999-08-18]

[Spink, Xu 2000] Spink, Amanda; Xu, Jack L.: *Selected results from a large study of Web searching: the Excite study.* In: *Information Research*, 6 (2000) 1, http://www.shef.ac.uk/~is/publications/infres/paper90.html [2000-12-28]

[Spoerri 1993] Spoerri, Anselm: *InfoCrystal: A Visual Tool for Information Retrieval* In: Nielson, Gregory M.; Bergeron, Daniel (Eds.): *Proceedings of IEEE Visualization 1993. Conference: San Jose, CA, October 25-29 1993.* Los Alamitos, CA (IEEE Computer Soc. Press) 1993. p. 150-157.

[Spoerri 1993a] Spoerri, Anselm: *InfoCrystal: a visual tool for information retrieval management.* In: *Proceedings of the second international conference on Information and knowledge management. Conference: Washington DC, USA, 1993 November 1 - 5* 1993. p. 11-20.

[Spotfire 2001] http://www.spotfire.com [2001-02-22]

[Spring, Morse, Heo 1996] Spring, Michael B.; Morse, Emile L.; Heo, Misook: *Multi-level navigation of a document space.* http://www.sis.pitt.edu/~spring/mlnds/mlnds/mlnds.html [1998-09-09]

[Stasko, Catrambone, Guzdial et al. 2000] Stasko, John T.; Catrambone, Richard; Guzdial, Mark et al.: *An Evaluation of Space-Filling Information Visualizations for Depicting Hierarchical Structures* In: *International Journal of Human-Computer Studies*, 53 (2000) 5, p. 663-694.

[Stone, Fishkin, Bier 1994] Stone, Maureen C.; Fishkin, Ken; Bier, Eric A.: *The Movable Filter as a User Interface Tool.* In: Adelson, B.; Dumais, S.; Olson, J. S. (Eds.): *CHI 1994: Conference Proceedings Human Factors in Computing Systems. Conference: Boston, MA, April 24-28 1994.* New York (ACM Press) 1994. p. 306-312.

[Stubblefield 1998] Stubblefield, William A.: *Patterns of change in design metaphor - a case study.* In: Karat, Clare-Marie; Karat, John; Horrocks, Ian (Eds.): *CHI 1998: Conference Proceedings Human Factors in Computing Systems. Conference: Los Angeles, CA, April 21-23 1998.* New York (Addison-Wesley) 1998. p. 73-80.

[Sullivan 2000] Sullivan, Dan: *Numbers, Numbers—But What Do They Mean?* http://searchenginewatch.internet.com/sereport/00/03-numbers.html [2000-09-08]

[Sullivan 2000a] Sullivan, Danny: *Search Spotlight.* http://searchenginewatch.internet.com/facts/spotlight.html [2000-12-29].

[Swan, Allan 1996] Swan, Russell C.; Allan, James: *Improving Interactive Information Retrieval Effectiveness with 3-D Graphics.* Technical Report IR-100. Department of Computer Science, University of Massachusetts, Amherst, 1996.

[Swan, Allan 1998] Swan, Russell C.; Allan, James: *Aspect Windows, 3-D Visualizations, and Indirect Comparisons of Information Retrieval Systems.* In: Croft, W. Bruce et al. (Eds.): *SIGIR 1998: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval Conference: Melbourne, Australia, August 24-28 1998.* New York (ACM Press) 1998. p. 173-183.

[Tauscher, Greenberg 1997] Tauscher, Linda M.; Greenberg, Saul: *How people revisit Web pages: Empirical findings and implications for the design of history systems.* In: *International Journal of Man-Machine Studies*, 47 (1997) 1, p. 97-137.

[Tauscher, Greenberg 1997a] Tauscher, Linda M.; Greenberg, Saul: *Revisitation Patterns in World Wide Web Navigation.* In: *CHI 1997: Conference Proceedings Human Factors in Computing Systems. Conference: Atlanta, GA, March 22-27 1997.* New York (ACM Press) 1997. p. 399-406.

[Tesler, Strasnick 1992] Tesler, Joel D.; Strasnick, Steven L.: *FSN: The 3D File System Navigator.* FSN (pronounced fusion) 3D File System Navigator for IRIX developped 1992 by Joel D. Tesler and Steven L. Strasnick from Silicon Graphics, Inc. Available via anonymous ftp at ftp.sgi.com/~ftp/sgi/fsn [2001-01-05]. See also http://www.sgi.com/fun/freeware/3d_navigator.html [2001-01-05]

[Triebe, Wittstock, Schiele 1987] Triebe, Johannes K.; Wittstock, Marion; Schiele, F.: *Arbeitswissenschaftliche Grundlagen der Software-Ergonomie.* Berlin (Bundesanstalt für Arbeitssschutz) 1987. (= Schriftenreihe der Bundesanstalt für Arbeitsschutz, Sonderschrift S. 24)

[Tufte 1983] Tufte, Edward R.: *The Visual Display of Quantitative Information.* Cheshire, CT (Graphics Press) 1983.

[Tweedie, Spence, Dawkes et al. 1996] Tweedie, Lisa A.; Spence, Robert; Dawkes, Huw et al.: *Externalising abstract mathematical models.* In: Tauber, M. J.; Bellotti, V.; Jeffries, R. et al. (Eds.): *CHI 1996: Conference Proceedings Human Factors in Computing Systems. Conference: Vancouver, BC, Canada, April 13-18 1998.* New York (ACM Press) 1996. p. 406-412.

[Tweedie, Spence, Williams et al. 1994] Tweedie, Lisa A.; Spence, Robert; Williams, David et al.: *The Attribute Explorer. Video Track.* In: Adelson, B.; Dumais, S.; Olson, J. S. (Eds.): *CHI 1994: Conference Proceedings Human Factors in Computing Systems. Conference: Boston, MA, April 24-28 1994.* New York (ACM Press) 1994. p. 435-436.

[Van Rijsbergen 1979] Van Rijsbergen, Cornelis J.: *Information Retrieval Second Edition.* London (Butterworths) 1979.

[Veerasamy 1996] Veerasamy, Aravindan: *Interactive TREC-4 at Georgia Tech.* In: Harman, Donna K. (Ed.): *NIST Special Publication 500-236: The Fourth Text REtrieval Conference (TREC-4). Conference: Gaithersburg, MD, November 1-3 1995.* Gaithersburg, MD (National Institute of Standards and Technology) 1995. p. 421-431.

[Veerasamy 1997] Veerasamy, Aravindan: *Visualization and User Interface Techniques for Interactive Information Retrieval Systems.* PhD Thesis, Georgia Institute of Technology. March 1997

[Veerasamy, Belkin 1996] Veerasamy, Aravindan; Belkin, Nicholas J.: *Evaluation of a Tool for Visualization of Information Retrieval Results.* In: Frei, Hans-Peter; Harman, Donna K.; Schäuble, Peter et al. (Eds.): *SIGIR 1996: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Conference: Zürich, Switzerland, August 18 -22 1996.* New York (ACM Press) 1996. p. 85-92.

[Veerasamy, Heikes 1997] Veerasamy, Aravindan; Heikes, Russell: *Effectiveness of a graphical display of retrieval results.* In: Belkin, Nicholas J.; Narasimhalu, A. Desai; Willett, Peter (Eds.): *SIGIR 1997: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Conference: Philadelphia, PA, July 27-31 1997.* New York (ACM Press) 1997. p. 236-245.

[Veerasamy, Hudson, Navathe 1995] Veerasamy, Aravindan; Hudson, Scott E.; Navathe, Shamkant B.: *Querying, Navigating and Visualizing an Online Library Catalog* . Graphics, Visualization & Usability (GVU) Center at Georgia Tech: Technical Report 95-24. 1995.

[Veerasamy, Navathe 1995] Veerasamy, Aravindan; Navathe, Shamkant B.: *Querying, Navigating and Visualizing a Digital Library Catalog.* In: *Digital Libraries 1995: The Second Annual Conference on the Theory and Practice of Digital Libraries. Conference: Austin, TX, June 11-13 1995.* http://www.csdl.tamu.edu/DL95/papers/veerasamy/veerasamy.html [1999-03-24]

[Vroomen 1998] Vroomen, Louis C.: *Information Visualization: An Overview.* 1998. http://www.crim.ca/~vroomen/writing/technical/reports/infovis.html [1999-01-12]

[Wang, Hawk, Tenopir 2000] Wang, Pieling; Hawk, William B.; Tenopir, Carol: *Users' interaction with World Wide Web resources: an exploratory study using a holistic approach.* In: *Information Processing & Management*, 36 (2000) 2, p. 229-251.

[Washburne 1927] Washburne, John N.: *An Experimental Study of Various Graphic, Tabular and Textual Methods of Presenting Quantitative Material. Part I.* In: *The Journal of Educational Psychology*, 18 (1927) 6, p. 361-376.

[Washburne 1927a] Washburne, John N.: *An Experimental Study of Various Graphic, Tabular and Textual Methods of Presenting Quantitative Material. Part II.* In: *The Journal of Educational Psychology*, 18 (1927) 7, p. 465-476.

[Weber, Groner 1999] Weber, Christian; Groner, Rudolf: *Suchstrategien im WWW bei Laien und Experten.* In: Wirth, Wolfgang; Schweiger, Werner (Eds.): *Selektion im Internet.* Opladen (Westdeutscher Verlag) 1999. p. 181-198.

[Wehrend, Lewis 1990] Wehrend, Stephen C.; Lewis, Clayton: *A problem-oriented classification of visualization techniques.* In: Kaufman, Arie E. et al. (Eds.): *Proceedings of the First IEEE Conference on Visualization. Conference: San Francisco, CA, October 23 - 26 1990.* Los Alamitos, CA. (IEEE Computer Soc. Press) 1990. p. 139-143.

[Williamson, Shneiderman 1992] Williamson, Christopher; Shneiderman, Ben: *The Dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration systems.* In: Belkin, Nicholas J.; Ingwersen, Peter; Pejtersen, Annelise M. (Eds.): *SIGIR 1992: Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Conference: Copenhagen, Denmark, June 21-24 1992.* New York (ACM Press) 1992. p. 338-346.

[Wills 1995] Wills, Graham J.: *Visual Exploration of Large Structured Data Sets.* In: Eurostat (Eds.): *New Techniques and Trends in Statistics.* Amsterdam (IOS Press) 1995. p. 237-246.

[Wise 1999] Wise, James A.: *The Ecological Approach to Text Visualization.* In: *Journal of the American Society for Information Science (JASIS)*, 50 (1999) 13, p. 1224-1233.

[Wise, Thomas, Pennock et al. 1995] Wise, James A.; Thomas, James J.; Pennock, Kelly et al.: *Visualizing the Nonvisual: Spatial Analysis and Interaction with Information from Text Documents.* In: Gershon, Nahum; Eick, Stephen G. (Eds.): *Proceedings of IEEE Information Visualization 1995. Conference: Atlanta, GA, October 30-31 1995.* Los Alamitos, CA (IEEE Computer Soc. Press) 1995. p. 51-58.

[Wiss, Carr 1999] Wiss, Ulrika; Carr, David A.: *An Empirical Study of Task Support in 3D Information Visualizations.* In: Will, Graham; Keim, Daniel A. (Eds.): *Proceedings of IEEE Information Visualization 1999. Conference: San Francisco, CA, October 24-29 1999.* Los Alamitos, CA (IEEE Computer Soc. Press) 1999. S. 392-399.

[Wistrand 1995] Wistrand, Erik: *Visualization Methods for Dynamic Queries Databases.* 1995. http://www.cs.chalmers.se/~wistrand/vizmethods.ps [1999-02-23]

[Wolte 1998] Wolte, Josef: *Information Pyramids: Compactly Visualising Large Hierarchies.* Graz University of Technology, Institute for Information Processing and Computer Supported New Media (IICM), Master's Thesis 1998

[Wood, Drew, Beale et al. 1995] Wood, Andrew M.; Beale, Russell; Drew, Nick S. et al.: *HyperSpace: Web Browsing with Visualisation.* In: *WWW 3: Third International World Wide Web Conference. Conference: Darmstadt, Germany, 1995* http://www.cs.bham.ac.uk/~amw/hyperspace/www95 [1998-11-16]

[Xu 1999] Xu, Jack L.: *Internet Search Engines: Real World IR Issues and Challenges.* Presentation to CIKM 99, October 31-November 4, 1999. Kansa City, MI. cited by [Spink, Xu 2000]

[Yamaguchi, Hosomi, Miyashita 1997] Yamaguchi, Tomoharu; Hosomi, Itaru; Miyashita, Toshiaki: *WebStage: An Active Media Enhanced World Wide Web Browser* In: *CHI 1997: Conference Proceedings Human Factors in Computing Systems. Conference: Atlanta, GA, March 22-27 1997.* New York (ACM Press) 1997. p. 391-398.

[Young, Shneiderman 1993] Young, Degi; Shneiderman, Ben: *A graphical filter / flow model for Boolean queries: An implementation and experiment.* In: *Journal of the American Society for Information Science (JASIS)*, 44 (1993) 6, p. 327-339.

[Zamir 1998] Zamir, Oren E.: *Visualization of Search Results in Document Retrieval Systems. General Examination.* Department of Computer Science and Engineering, University of Washington, September 1998.

[Ziegler, Fähnrich 1988] Ziegler, Jürgen E.; Fähnrich, Klaus-Peter: *Direct Manipulation.* In: Helander, Martin (Ed.): *Handbook of Human-Computer Interaction* Amsterdam (North-Holland) 1988. p. 123-133.

[Zimmer 1997] Zimmer, Dieter E.: *Bauarbeiten an einem Luftschloß. Die digitale Bibliothek (I) - Eine Artikelserie für Nutzer und Verächter der Computernetze.* In: Die Zeit, Nr. 39 2000-09-19.

7. Index of figures and tables

7.1. Figures

Figure 1: Structure of the thesis (main parts) 17
Figure 2: High-level tasks by [Hearst 1999] and [Shneiderman 1998] 21
Figure 3: Information seeking functions according to [Marchionini 1992] p. 157 FIG. 1. 24
Figure 4: Information-seeking process according to [Marchionini 1997] p. 50 Figure 3.3 25
Figure 5: Simplified diagram of the standard model of the information access processes according to [Hearst 1999] p. 263, Figure 10.2 25
Figure 6: Number of queries per session according to [Jansen, Spink, Bateman et al. 1998], [Silverstein, Henzinger, Marais et al. 1999] 33
Figure 7: Terms in query according to [Hölscher 2000], [Röttgers 1999], [Jansen, Spink, Bateman et al. 1998], and [Silverstein, Henzinger, Marais et al. 1999] 34
Figure 8: Average number of terms per query over time and area 36
Figure 9: Result pages viewed per query according to [Hölscher 1998a], and [Silverstein, Henzinger, Marais et al. 1999] 39
Figure 10: Distribution of modifier usage for queries with modifiers 12 Experts / Fireball / Excite 41
Figure 11: Distribution of modifier usage for all queries 12 Experts / Fireball / Excite 42
Figure 12: Percentage of additional result screens from all result screens 43
Figure 13: Slightly modified reference model for visualization adapted from [Card, Mackinlay, Shneiderman 1999] Figure 1.23 47
Figure 14: Classification of post-retrieval document visualization techniques according to [Zamir 1998] Fig. 1 49
Figure 15: Dimensions of the classification scheme for visualizations 50
Figure 16: Classification of visualization components 50
Figure 17: Starting point of a metaphor in the reference model of visualization 51
Figure 18: Expanded reference model for visualization: Brushing 60
Figure 19: Reference model for visualization: Panning and zooming 61
Figure 20: Panning and zooming, including different types of zooming 62
Figure 21: Reference model for visualization: Focus-plus-context 63
Figure 22: Reference model for visualization: Magic Lenses 64
Figure 23: Reference model for visualization: Animation 65
Figure 24: Reference model for visualization: Overview plus detail 66
Figure 25: Principle of the Query Reformulation Workspace used in the AI-STARS system by [Anick, Brennan, Flynn et al. 1990] 67
Figure 26: Principle of Filter/Flow component by [Shneiderman 1991], [Young, Shneiderman 1993] 68
Figure 27: Principle of Movable Filters / Magic Lens Filters by [Fishkin, Stone 1995] 68
Figure 28: Venn diagram for the concepts (visualization OR visualisation), search, results. 69
Figure 29: Principle of the Query workspace with Venn Diagrams in the VQuery system by [Jones 1998], [Jones 1998a] 69
Figure 30: Principle of the InfoCrystal by [Spoerri 1993], [Spoerri 1993a] 70
Figure 31: Principle of the "Bracket"-visualization by [Bürdek, Eibl, Krause 1999], [Eibl 1999] 71
Figure 32: Principle of the Bargraph in the VIEWER system by [Berenci, Carpineto, Giannini 1998] 71
Figure 33: Principle of the Keyword-Concept Matrix or Concept Control used in the NIRVE system by [Cugini, Laskowski, Piatko 1998], [Cugini, Laskowski, Sebrechts 2000]. 72
Figure 34: Principle of the 2D Global View used in the NIRVE system by [Cugini, Laskowski, Sebrechts 2000] 72
Figure 35: Principle of the Request Map by [Fowler, Fowler, Wilson 1991], [Fowler, Wilson, Fowler 1992] 73
Figure 36: Principle of Positive / Negative Feedback by [Veerasamy, Navathe 1995], [Veerasamy, Hudson, Navathe 1995] 73
Figure 37: Principle of LiveTopics / Cow9 by [Bourdoncle 1997] 74
Figure 38: Thumbnail Views 76
Figure 39: Principle of "closed" and "thumbnail" views with relevance indicator and colored keyword hightlighting by [Kaugars 1998] 76
Figure 40: Thumbnail views in the J24 interface by [Ogden, Davis, Rice 1998] 77
Figure 41: Principle of the Document Lens by [Robertson, Mackinlay 1993]. 77
Figure 42: Principle of semi-open documents by [Kaugars 1998], [Ogden, Davis, Rice 1998] 78
Figure 43: Relevance Curve and Relevant Extracts from DigOut4U of Arisem S.A. Paris 78
Figure 44: Principle of the SeeSoft bar view according to [Eick 1994], [Wills 1995] 79
Figure 45: Possible mappings in the Matrix of Icons of the Graph View of the Envision system according to [Nowell, France, Hix et al. 1996] 80
Figure 46: Principle of Dotplots according to [Church, Helfman 1993] 80
Figure 47: Examples of Dotplots. From left to right: plots of the first 308 characters of the first paragraph of the Declaration of Independence of the United States of America, of the first 308 characters of the abstract from [Mann 1999], and of the 308 characters dada poem "What a be what a be what a beauty" by Kurt Schwitters. 81
Figure 48: Principle of bar-graphs by [Veerasamy 1996] / [Veerasamy, Belkin 1996] 82
Figure 49: Principle of stacked histograms / VQRa of the WInquery system by [Shneiderman, Byrd, Croft 1997], [Byrd 1999] 82
Figure 50: Principle of the Iconic Representation in the 3D Document Space of the NIRVE system by [Cugini, Laskowski, Piatko 1998] 83
Figure 51: Principle of R-Wheels (Result Wheels) by [Grewal, Burden, Jackson et al. 1999], [Grewal, Jackson, Burden et al. 2000] 83
Figure 52: Additional ideas of [Veerasamy 1997] and [Grewal, Jackson, Burden et al. 2000] 83
Figure 53: Principle of the Retrieval History Histogram of the VOIR system by [Golovchinsky 1997] 84
Figure 54: Principle of the TileBars by [Hearst 1995] 85
Figure 55: Principle of Mural and TileBars in the CASCADE system by [Spring, Morse, Heo 1996], [Heo, Morse, Willms et al. 1996] 85
Figure 56: Principle of VQRb in the FancyV prototype by [Byrd 1999] 86
Figure 57: Example of the libViewer from the SOMLib system [Rauber, Bina 1999], [Rauber, Bina 2000] 87
Figure 58: Principle of Entity Relation Visualization by [Chase, D'Amore, Gershon et al. 1998] Figure 3e. 87
Figure 59: Principle of the Bargraph in the XINQUERY system according to [Shneiderman, Byrd, Croft 1997] 89

Figure 60: Principle of the FISH component from the Starfish system of [Mitchell, Day, Hirschman 1995] 90
Figure 61: Example of the Treemap View from the Information Navigator [Au, Carey, Sewraz et al. 2000] 90
Figure 62: Principle of the Value Bars by [Chimera 1992] 91
Figure 63: Dynamic Homefinder showing selected houses in the area of Washington D.C. 91
Figure 64: Principle of the FilmFinder demonstrated by using Spotfire Pro 3.3.4 Demo 92
Figure 65: Scatterplot with two axes 93
Figure 66: Three-Keyword Axes Display (left figure), 3-D Axes (right figure) from the NIRVE system. Courtesy of NIST John V. Cugini 93
Figure 67: Principle of the Matrix of Icons of the Envision system according to [Nowell, France, Hix et al. 1996] 94
Figure 68: Principle of the map display of the VisageWeb system according to [Higgins, Lucas, Senn 1999] 95
Figure 69: Galaxy view from the SPIRE system. Courtesy of Pacific Northwest National Laboratory 95
Figure 70: Principle of the reference points - documents display of the VIBE system and explanations according to [Korfhage 1991] page 138 96
Figure 71: Example of the reference points - hosts/documents display of the SQWID system [McCrickard, Kehoe 1997] 97
Figure 72: Example of the Radial visualization from the Information Navigator [Au, Carey, Sewraz et al. 2000], [Carey, Kriwaczek, Rüger 2000] 98
Figure 73: Document space with two reference points 98
Figure 74: ThemeView from the SPIRE system. Courtesy of Pacific Northwest National Laboratory 99
Figure 75: Principle of the self-organizing semantic map according to [Lin, Soergel, Marchionini 1991] 100
Figure 76: SOMs of the 20 document result set of the WebViz-example 100
Figure 77: SOM of the AltaVista query “visualization search results internet” 101
Figure 78: SOM example from the Visual SiteMap system 101
Figure 79: Self Organizing Maps of the WEBSOM system [Kohonen 1998] 102
Figure 80: Example of the libViewer from the SOMLib system [Rauber, Bina 1999] 102
Figure 81: Example of the Sammon View from the Information Navigator [Au, Carey, Sewraz et al. 2000], [Carey, Kriwaczek, Rüger 2000] 103
Figure 82: Principles of two of the pixel-oriented visualization techniques used in the VisDB system by [Keim, Kriegel 1994] 104
Figure 83: Pixel oriented visualization techniques. Examples taken from [Keim, Kriegel 1994] Figure 6. Eight dimensional data plus overall relevance, 1000 items. Courtesy of Daniel A. Keim 104
Figure 84: Principle of Circle Segments 104
Figure 85: Principle of the Document Spiral in the NIRVE system by [Cugini, Piatko, Laskowski 1997] 105
Figure 86: Document Spiral of the NIRVE system. Courtesy of NIST John V. Cugini 105
Figure 87: Principle of the Nearest Neighbor Circle (NNC) and example from the NIRVE system. Courtesy of NIST John V. Cugini 106
Figure 88: Principle of the Spoke and Wheel and example from the NIRVE system. Courtesy of NIST John V. Cugini 107
Figure 89: Principle of the spiral display by [Hascoët 1998] 107
Figure 90: Principle of Interactive Histograms in the Attribute Explorer by [Tweedie, Spence, Williams et al. 1994] 108
Figure 91: Principle of Interactive Histograms in the Influence Explorer by [Tweedie, Spence, Dawkes et al. 1996] 108
Figure 92: Principle of Parallel Coordinates 108
Figure 93: Principle of the Perspective Wall by [Mackinlay, Robertson, Card 1991] 109
Figure 94: Principle of the Bifocal Display by [Spence, Apperley 1982] 109
Figure 95: Example of Table Lens / Eureka, ordered by Year (left) and Relevance (right) 110
Figure 96: “Bar-graph view” of Table Lens / Eureka 110
Figure 97: Example of FOCUS / InfoZoom, ordered by Year (left) and Relevance (right) 110
Figure 98: Example of the FOCUS / InfoZoom overview function 111
Figure 99: Types of visualizations tested by [Washburne 1927] 121
Figure 100: Bar-chart tested by [Grewal, Jackson, Burden et al. 2000] plus untested alternative view 124
Figure 101: The INSYDER system, example of SOI building and construction 130
Figure 102: The INSYDER system, example of SOI CAD 130
Figure 103: INSYDER Architecture 134
Figure 104: INSYDER HTML/JavaScript-prototype 135
Figure 105: INSYDER VisualBasic-prototype 135
Figure 106: INSYDER HTML / JavaScript / Applet – file- prototype 136
Figure 107: INSYDER HTML / JavaScript / Applet – HTTP – prototype 136
Figure 108: Main branches of the mindmap of planned functions for the INSYDER system 137
Figure 109: Complete mindmap of planned functions for the INSYDER system 137
Figure 110: Example branch “Action” from the mindmap of planned functions for the INSYDER system 138
Figure 111: Navigation concept 140
Figure 112: Tabbed pane 140
Figure 113: Availability of term ranking details in the INSYDER system 146
Figure 114: Example for Document Group Tooltip 147
Figure 115: HTML-List, INSYDER integrated browser 148
Figure 116: HTML-List with navigation elements, external browser 148
Figure 117: ResultTable from the INSYDER system 149
Figure 118: Doubles and Relevance Curve in the ResultTable of the INSYDER system, San Francisco example 149
Figure 119: ScatterPlot Date / Relevance from the INSYDER system 150
Figure 120: ScatterPlot Date / Relevance zoomed 1995 – 2001, Tooltip and options 150
Figure 121: Scatterplot server type (category) / number, and vector mode (keyword visualization / number) 151
Figure 122: Selected documents (red dots), Document groups (rectangles), San Francisco example 151
Figure 123: BarGraph from the INSYDER system 152
Figure 124: TileBar view with Tooltip 153
Figure 125: SegmentView - pop-up window 153
Figure 126: TileBar 3 Steps 154

Figure 127: TileBar 3 Sizes 154
Figure 128: TileBar continuous size 154
Figure 129: StackedColumn Wide 155
Figure 130: StackedColumn Small 155
Figure 131: Document Vector 156
Figure 132: History of the INSYDER visualizations ideas for the result phase 157
Figure 133: Example user interface conditions for the first three tasks of group 1. 161
Figure 134: User characteristics: Age, Gender, and Profession 162
Figure 135: User characteristics: Computer Experience, WWW Dependency, Search engine/IR-system Usage 163
Figure 136: Bargraph with 30 hits: one, three, or eight keywords 164
Figure 137: Bargraph with 500 hits: one, three, or eight keywords 164
Figure 138: SegmentView (TileBars 3 Steps): one, three, or eight keywords 164
Figure 139: ScatterPlot: 30 or 500 hits 164
Figure 140: Selected tasks and their position in the information seeking process 165
Figure 141: Overview of the final test procedure for the members of the five different groups 170
Figure 142: Tasks ResultTable vs. Visualization 176
Figure 143: Usage time ResultTable vs. Visualization 177
Figure 144: Usage patterns of the visualizations 177
Figure 145: Suitability: Adaptation to the demands 179
Figure 146: Suitability: Helpfulness to support work 179
Figure 147: Suitability: Want be supported by these visualizations in the future 179
Figure 148: Suitability: Visualizations complement each other 180
Figure 149: Reorientation after changing the visualization 180
Figure 150: Components best to work with 181
Figure 151: Dispensable components 182
Figure 152: Able to work with the components after a longer break 182
Figure 153: Self descriptiveness of the components 182
Figure 154: Benefit unclear 183
Figure 155: The application is intuitively learnable 184
Figure 156: Felt learning effects, now able to manage the work with the components better 184
Figure 157: Ability to operate the application at the end of the test 184
Figure 158: Confused when working with the visualizations 185
Figure 159: Confidence about having always proceeded correctly 185
Figure 160: Greater efficiency of visualizations compared to search engines known to the users 185
Figure 161: Components that helped most to solve the given problem 186
Figure 162: Design of the components 186
Figure 163: Too complex or overloaded components 187
Figure 164: Candidates for design improvements 187
Figure 165: Using the visualizations was a waste of time 187
Figure 166: Fun factor of the components 188
Figure 167: Mood before the test 188
Figure 168: Mood after the test (left) and changes (right) 189
Figure 169: Summarized user ratings of the components 190
Figure 170: Mean user satisfaction index and standard deviation 191
Figure 171: Summarized user ratings of the components per user group 191
Figure 172: Mean user satisfaction index and standard deviation per user group 192
Figure 173: Box plot legend 193
Figure 174: Mean task time per user 194
Figure 175: Box plot task time per user (min / 25%-quartile / median / 75%-quartile / max) 194
Figure 176: Mean effectiveness per user 195
Figure 177: Box plot effectiveness per user (min / 25%-quartile / median / 75%-quartile / max) 195
Figure 178: Average effectiveness and task time per user 195
Figure 179: Mean efficiency per user 196
Figure 180: Box plot efficiency per user (min / 25%-quartile / median / 75%-quartile / max) 196
Figure 181: Mean task time per question 197
Figure 182: Box plot task time per question (min / 25%-quartile / median / 75%-quartile / max) 197
Figure 183: Mean effectiveness per question 198
Figure 184: Box plot effectiveness per question (min / 25%-quartile / median / 75%-quartile / max) 198
Figure 185: Average effectiveness and task time per question 199
Figure 186: Mean efficiency per task 199
Figure 187: Mean effectiveness per user interface condition 200
Figure 188: Mean effectiveness index and standard deviation 201
Figure 189: Mean task time per user interface condition and situation for the Visualization plus ResultTable conditions 202
Figure 190: Mean task time index and standard deviation 202
Figure 191: Mean effectiveness and mean task time per user interface condition 203
Figure 192: Mean effectiveness index (y-axis) and mean task time index (x-axis) per user interface condition 203
Figure 193: Mean efficiency per user interface condition (calculated by using efficiency per question) 204
Figure 194: Mean efficiency per user interface condition (calculated by using average task time in minutes and average effectiveness) 204

Figure 195: Mean effectiveness per user interface condition in the original sequence of the questions 207
Figure 196: Mean task time per user interface condition in the original sequence of the questions 208
Figure 197: Relative mean effectiveness of the user interface conditions per question 209
Figure 198: HTML-List and ResultTable of the "Danube"-question result set 209
Figure 199: SegmentView and user configured ScatterPlot of the "Danube"-question result set 210
Figure 200: Mean effectiveness index and standard deviation per user group 211
Figure 201: Mean task time index and standard deviation per user group 211
Figure 202: Mean effectiveness index and mean task time index per target user group 212
Figure 203: Usage of the components in the Visualization plus ResultTable conditions per user group 213
Figure 204: Mean effectiveness index and standard deviation per task type 213
Figure 205: Mean task time index for extended fact-finding tasks and standard deviation 214
Figure 206: Mean effectiveness index and mean task time index per task type 214
Figure 207: Usage of the components in the Visualization plus ResultTable conditions per task type 215
Figure 208: Mean effectiveness index and standard deviation per size of the document set 216
Figure 209: Mean task time index and standard deviation per size of the document set 216
Figure 210: Mean effectiveness index and mean task time index per size of the document set 217
Figure 211: Usage of the components in the Visualization plus ResultTable conditions per number of hits 218
Figure 212: Mean effectiveness index for one-keyword tasks and standard deviation 218
Figure 213: Mean effectiveness index for three-keywords tasks and standard deviation 219
Figure 214: Mean effectiveness index for eight-keywords tasks and standard deviation 219
Figure 215: Mean task time index for one-keyword tasks and standard deviation 220
Figure 216: Mean task time index for three-keywords tasks and standard deviation 220
Figure 217: Mean task time index for eight-keywords tasks and standard deviation 220
Figure 218: Usage of the components in the Visualization plus ResultTable conditions per number of keywords 221
Figure 219: Further integration of the components and of additional ideas 231
Figure 220: Box plot efficiency per task (min / 25%-quartile / median / 75%-quartile / max) 257
Figure 221: Box plot efficiency per user interface condition (min / 25%-quartile / median / 75%-quartile / max) 257
Figure 222: Box plot efficiency per user interface condition (full scale) 258
Figure 223: Task time and used components per user and question 259
Figure 224: Effectiveness and used components per user and question 260
Figure 225: Questions not solved and used components per user 261
Figure 226: Efficiency and used components per user and question 262
Figure 227: Planned INSYDER functions SOI, configuration of system, general requirements 264
Figure 228: Planned INSYDER functions: Formulation + Action 265
Figure 229: Planned INSYDER functions: Results + Refinement 266

7.2. Tables

Table 1: Four-phase framework of information seeking according to [Shneiderman, Byrd, Croft 1997] 12
Table 2: Task actions according to [Shneiderman 1998] 20
Table 3: Examples of Information-Seeking Strategies ISS according to [Belkin, Marchetti, Cool 1993] and [Belkin, Cool, Stein et al. 1995] 22
Table 4: Comparison of literature search and Web moves according to [Choo, Detlor, Turnbull 1999] Fig. 2. 23
Table 5: Comparison of the search models by [Marchionini 1992], [Marchionini 1997], [Hearst 1999], [Veerasamy, Heikes 1997], and [Shneiderman 1998] 26
Table 6: Tasks of the TTT data type by task taxonomy according to [Shneiderman 1998] p. 524 Box 15.2 27
Table 7: Main tasks supported by visualizations in data analysis according to [Rohrer, Swing 1997] 27
Table 8: User information seeking goals according to [Roth, Mattis 1990] 27
Table 9: Analysis techniques according to [O'Day, Jeffries 1993] 28
Table 10: User tasks and goals in interactive data exploration according to [Goldstein, Roth 1994] 28
Table 11: Operation classes in a visual environment according to [Wehrend, Lewis 1990], Table 2 28
Table 12: Navigation tasks in a document space taken as an assumption by [Spring, Morse, Heo 1996] to develop the CASCADE system. 29
Table 13: Explanation of terms from the reference model for visualization according to [Card, Mackinlay, Shneiderman 1999] Figure 1.23 47
Table 14: Data types of the TTT data type by task taxonomy from [Shneiderman 1996], [Shneiderman 1998] 49
Table 15: Metaphors used for the visualization of queries, search results or browsing 60
Table 16: Focus plus context: selective reduction of information for the context according to [Card, Mackinlay, Shneiderman 1999] 63
Table 17: Components for the visualization of queries or query attributes 74
Table 18: Components for the visualization of document attributes 88
Table 19: Components for the visualization of interdocument similarities 112
Table 20: Systems fully or partially used for the visualization of Web search results or hypertext 114
Table 21: Classical IR and library systems, including those with Web-based interface. 115
Table 22: Hypertext browsing systems, including Web browsers and their add-ons 116
Table 23: Other systems 116
Table 24: Rules when to use multiple views and areas of major impact on utility according to [Baldonado, Woodruff, Kuchinsky 2000], Table 1, Lines 1 to 4 119
Table 25: Rules how to use multiple views and areas of major impact on utility according to [Baldonado, Woodruff, Kuchinsky 2000], Table 1, Lines 5 to 8 120
Table 26: Overview of the document attributes used in the INSYDER system 142
Table 27: Data Table of the Documents 143

Table 28: Sample of document extracts and corresponding queries from the INSYDER system 144
Table 29: Data Table of the Queries 145
Table 30: Main Hypothesis of the INSYDER visualization evaluation February – April 2000 157
Table 31: Experimental conditions 158
Table 32: Dependent Variables 158
Table 33: Specific fact-finding tasks 167
Table 34: Extended fact-finding tasks 167
Table 35: Task times of the pre-test 171
Table 36: Combination of test tasks by variables 173
Table 37: Combination of test tasks by question 174
Table 38: Questions grouped by category and question type 175
Table 39: Two-tailed paired t-test: user satisfaction index 191
Table 40: Two-tailed paired t-test: user satisfaction index per user group 192
Table 41: Two-tailed paired t-test: effectiveness indexes for the user interface conditions 201
Table 42: Two-tailed paired t-test: task time indexes for the user interface conditions 203
Table 43: Task 5 (San Francisco), mean task times and standard deviation per user interface condition for all cases with 100% effectiveness 205
Table 44: Task 9 (Exxon Valdez), mean task times and standard deviation per user interface condition for all cases with 100% effectiveness 205
Table 45: Task 11 (Titanic), mean task times and standard deviation per user interface condition for all cases with 100% effectiveness 206
Table 46: One-way ANOVA of the task times for tasks 5, 9, and 11 206
Table 47: Task 5 (San Francisco), Bonferroni analysis user interface condition / task time 206
Table 48: Task 9 (Exxon Valdez), Bonferroni analysis user interface condition / task time 206
Table 49: Task 11 (Titanic), Bonferroni analysis user interface condition / task time 207
Table 50: Two-tailed paired t-test: effectiveness index per user group 211
Table 51: Two-tailed paired t-test: task time index per user group 212
Table 52: Two-tailed paired t-test: effectiveness index per task type 214
Table 53: Two-tailed paired t-test: task time index per task type 214
Table 54: Two-tailed paired t-test: effectiveness index per size of the document set 216
Table 55: Two-tailed paired t-test: task time index per size of the document set 217
Table 56: Two-tailed paired t-test: effectiveness index per number of keywords 219
Table 57: Two-tailed paired t-test: task time index per number of keywords 220
Table 58: Specific factfinding tasks and answers. German original 256
Table 59: Extended factfinding tasks and answers. German original 256
Table 60: Two-sample t-test (homoscedastic), effectiveness per user interface condition, beginners versus experts. 263
Table 61: Two-sample t-test (homoscedastic), task time per user interface condition, beginners versus experts 263

8. Appendix

8.1. Tasks

#	Task	Keywords	Hits	Answers
1	Wie lang ist die Donau?	danube	30	1.170 miles
3	Wer verlor 1997 die 2. Partie beim Schach-wettkampf Gari Kasparow gegen Deep Blue?	gari kasparow deep blue ibm chess result game	30	Kasparow
5	Wieviele Einwohner besitzt San Francisco zur Zeit?	san francisco inhabitants	30	724.000 / 725.000
7	Wie groß ist die Masse des Mondes?	moon	500	7,349e22 kg
9	Welche Menge Öl (t, l, Barrel) verlor der Tanker „Exxon Valdez“ bei seiner Havarie?	exxon valdez oil pollution cata-strophe tanker average spill	500	42 Mio l / 232.000 Bar-rel / 11 Mio Gallons
11	An welchem Tag sank die Titanic?	titanic sinking iceberg	500	1912-04-14 / 1912-04-15

Table 58: Specific factfinding tasks and answers. German original

#	Task	Keywords	Hits	Answers
2	Nennen Sie die Namen von Nati-onalparks (National Parks, NP) in Kalifornien!	national park california	500	9: Redwood NP, Death Valley NP, Kings Canyon NP, Sequoia NP, Yosemite NP, Joshua Tree NP, Lassen Volkcanic NP, Channel Islands NP, Mojave NP
4	Nennen Sie Städte, deren Museen Werke des venezianischen Malers Tizian ausstellen!	tizian	500	16: Berlin, Dresden, Kassel, München, Köln, Nea-pel, Rom, Venedig, Paris, Madrid, Wien, London, Washington, Prag, St. Petersburg, Edinburgh
6	Nennen Sie Berge, die höher als 8000 Meter (26248 feet) sind!	mountain hima-laya altitude height top peak reinhold messner	500	14: Mt. Everest, K2, Kangchenjunga, Lhotse, Makalu, Cho Oyu, Dhaulagiri, Manaslu, Nanga Parbat, Annapurna, Gasherbrum I (Hidden Peak), Broad Peak, Shisha Pangma, Gasherbrum II
8	Nennen Sie Bücher des Autoren John Irving!	john irving book	30	12: A widow for one year, A prayer for Owen Meany, The cider house rules, The world according to Garp, A son of the circus, The 158 pound mar-riage, Setting free the bears, The water method man, My movie business, Imaginary Girld Friend, The hotel New Hampshire, Trying to save Piggy Sneed
10	Was alles (Firmen, Projekte usw.) trägt heute den Namen des Philo-sophen Platon?	platon	30	7: Real time teknik analiz / MS Windows, Distrib-uted language based on asynchronous message pass-ing, QMS /Business Improvement / Lotus Notes, Waterproof, Hotel, A Multipurpose Crystallographic Tool, Platon Flow Management / Roxspur
12	Zu welchen Preisen werden die Karten der Kategorie „upper level end-zone“ für den 34.Superbowl bei den verschiedenen Online-Shops im einzelnen angeboten?	superbowl nfl national football league ticket xxxiv atlanta	30	5: 1,750 $, 1,850 $, 1,850 $, 1,850 $, 2,100 $

Table 59: Extended factfinding tasks and answers. German original

8.2. Additional figures from the hard facts

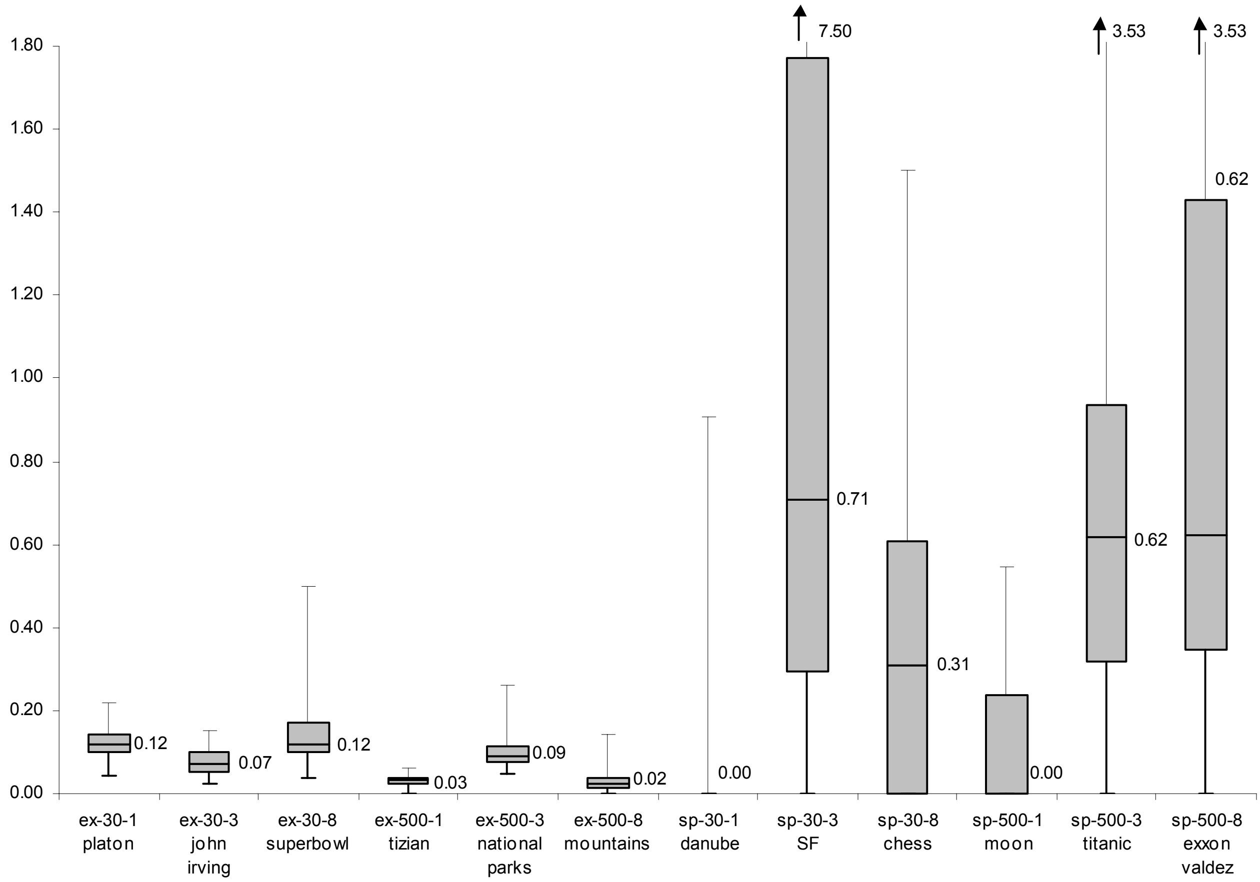


Figure 220: Box plot efficiency per task (min / 25%-quartile / median / 75%-quartile / max)

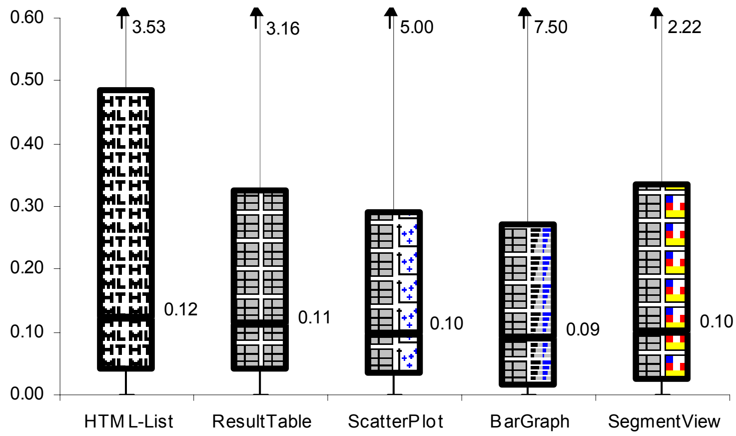


Figure 221: Box plot efficiency per user interface condition (min / 25%-quartile / median / 75%-quartile / max)

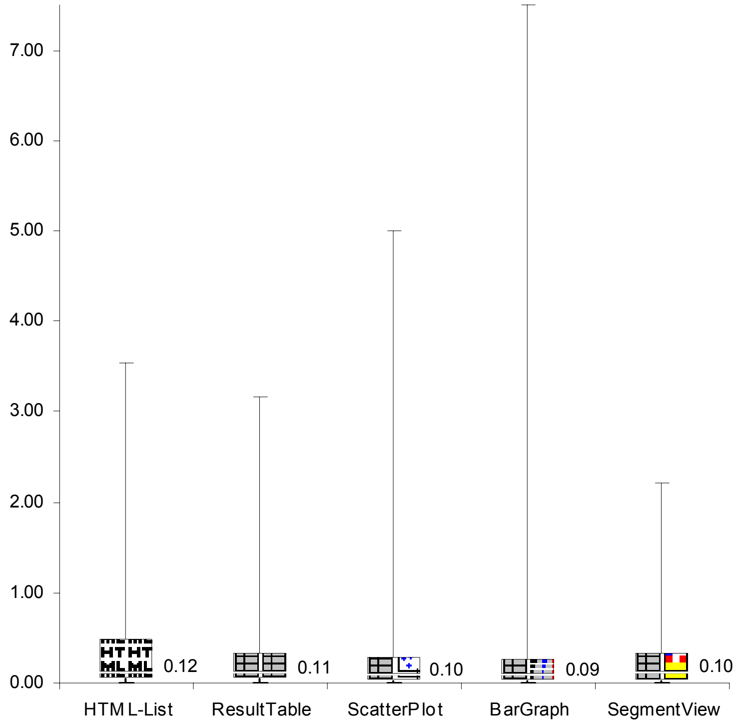


Figure 222: Box plot efficiency per user interface condition (full scale)

For the next figures the mapping of the used components to visual structures is a little bit different compaed with earlier figures. The reason to dos this was to show more details on less space. The so far used for the scatterplot condition is broken up into for users which used only the ScatterPlot component, for subjects which used ScatterPlot plus ResultTable, and for subjects which used only the ResultTable in the ScatterPlot condition. The same is done with in the BarGraph condition broken up to , , and , and with in the SegmentView condition broken up to , , and .

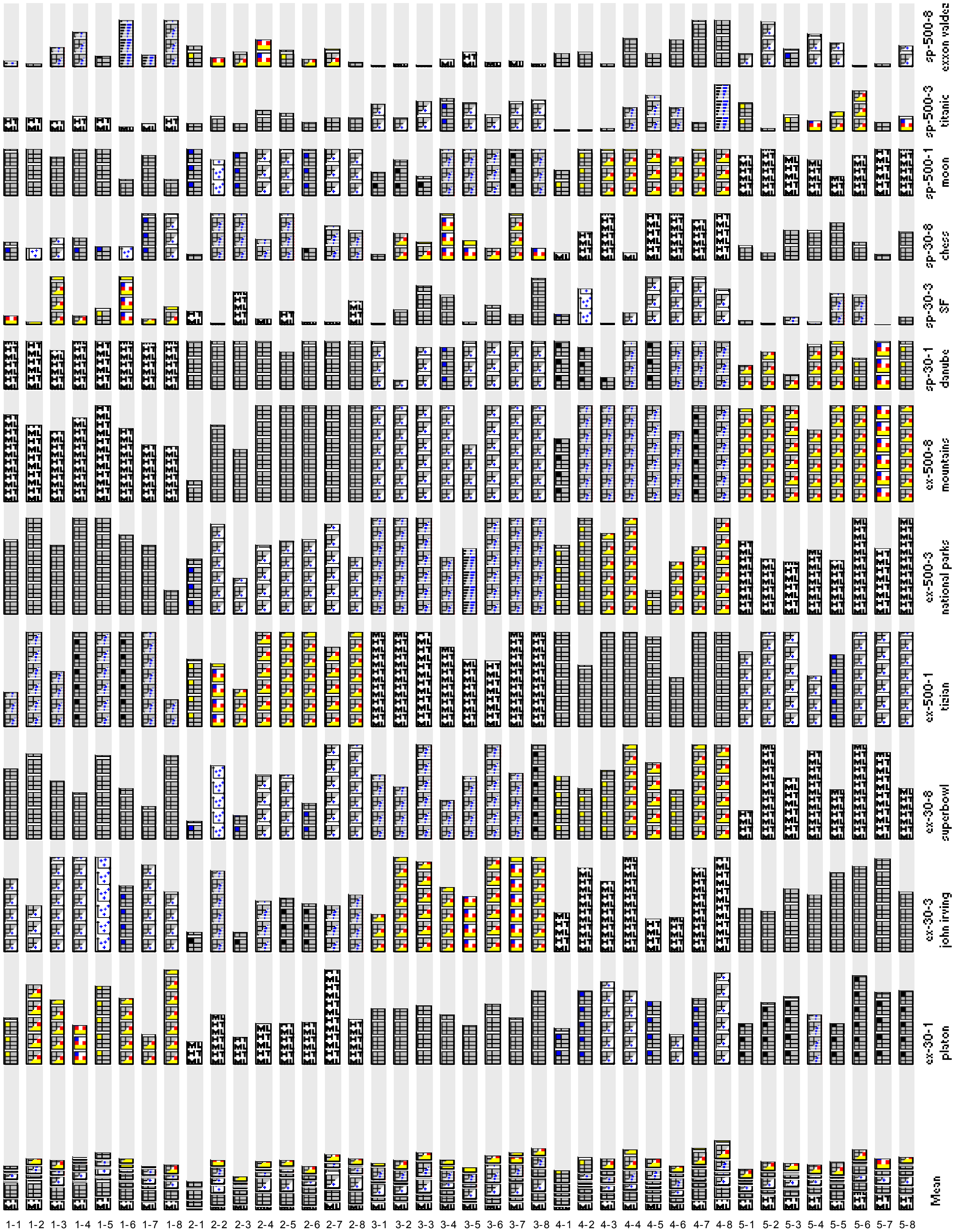


Figure 223: Task time and used components per user and question

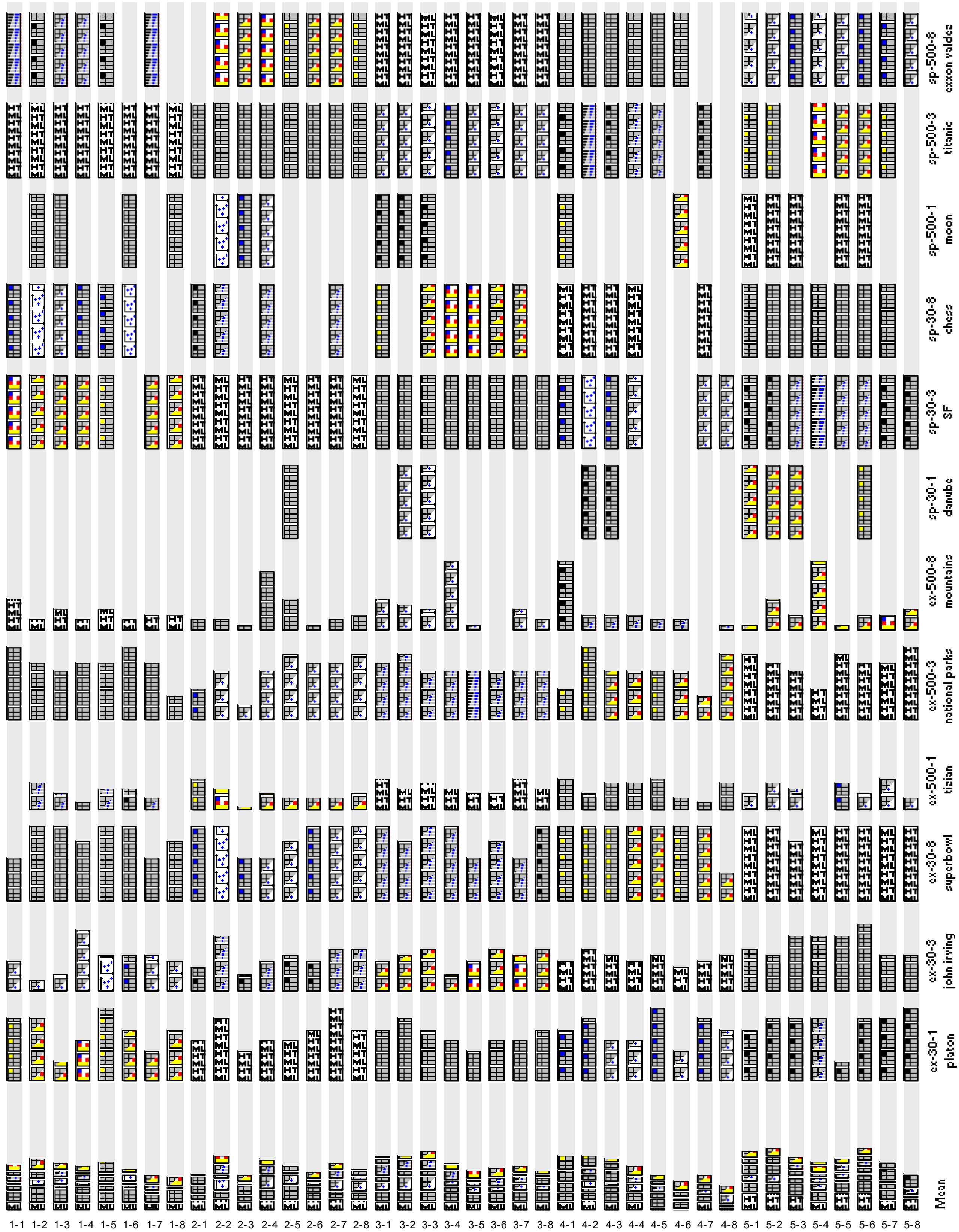


Figure 224: Effectiveness and used components per user and question

Due to the figure's mapping from data tables to visual structures, it cannot always be identified which components were used in conditions where the users had an effectiveness of 0% (i.e. where they did not answer the question). This was the case in 83 of the 480 evaluation cells. Figure 225 shows the cases where users failed to solve questions and which components were used.

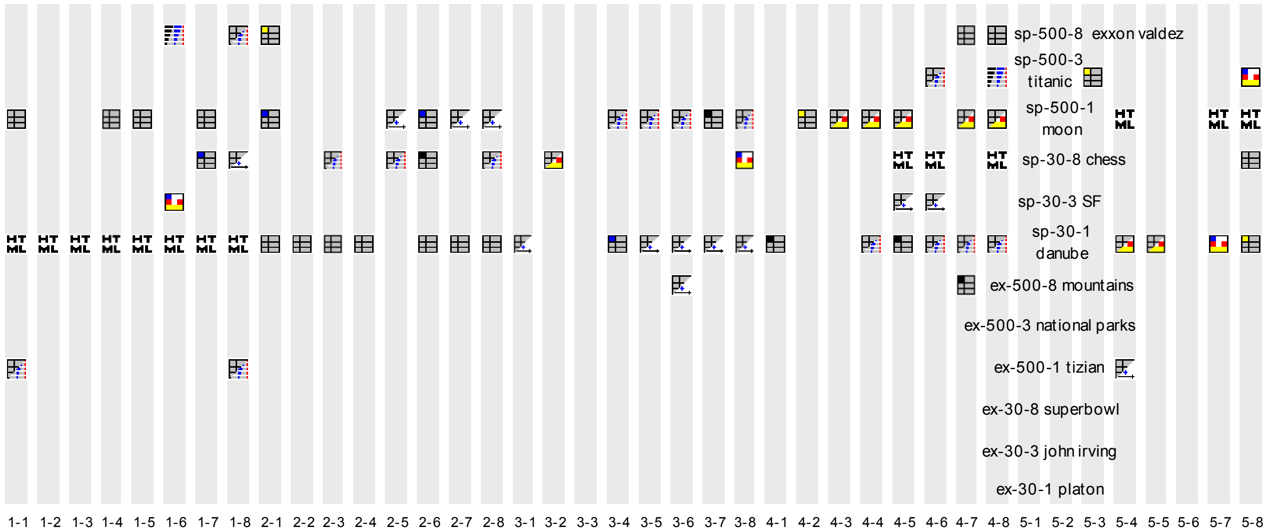


Figure 225: Questions not solved and used components per user

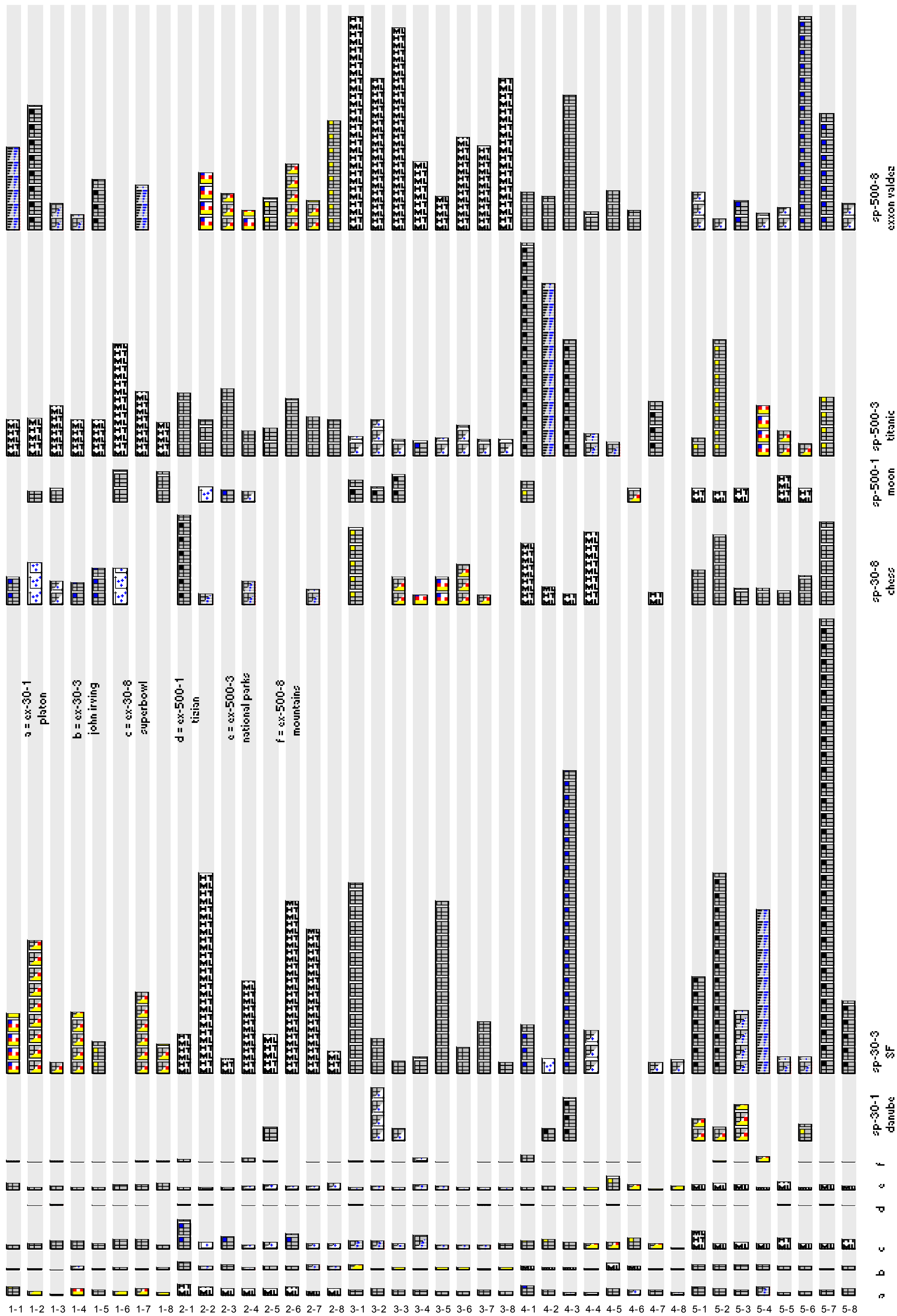


Figure 226: Efficiency and used components per user and question

8.3. Additional inferential statistics

Examination of the question: Are differences in the effectiveness of the user interface conditions between Beginners and Experts?

The effectiveness indices which had been calculated for testing the first hypothesis can also been used to examin tis question. In a t-test for independent samples the mean values of the experts were then compared with those of the Beginner. Since in all comparisons the variances of both groups did not differ significantly from each other, a two-sample t-test assuming equal variances (homoscedastic) were performed. Table 60 shows that there could be found for no user interface condition a significant difference in the effectiveness between beginnern and experts.

Variable	Method	Variances	DF	t value	Pr > |t|
HTML-List	Pooled	Equal	38	-1.51	0.1406
ResultTable	Pooled	Equal	38	0.85	0.4002
ScatterPlot + Result Table	Pooled	Equal	38	-1.67	0.1037
BarGraph + Result Table	Pooled	Equal	38	-0.88	0.3819
SegmentView + Result Table	Pooled	Equal	38	0.00	1.0000

Table 60: Two-sample t-test (homoscedastic), effectiveness per user interface condition, beginners versus experts.

When studying the task times with the same methods Table 61 shows that on 5% significance level, experts have significantly longer task times for the ScatterPlot than beginners. This is not coupled with a higher effectiveness which suggests the conclusion that experts work in the ScatterPlot condition more inefficiently than beginners.

Variable	Method	Variances	DF	t value	Pr > |t|
HTML-List	Pooled	Equal	38	-1.50	0.1424
ResultTable	Pooled	Equal	38	0.37	0.7122
ScatterPlot	Pooled	Equal	38	-2.53	0.0156*
BarGraph	Pooled	Equal	38	-0.40	0.6898
SegmentView	Pooled	Equal	38	-0.94	0.3555

Table 61: Two-sample t-test (homoscedastic), task time per user interface condition, beginners versus experts

* significant (p<.05)

8.4. INSYDER function Mindmap

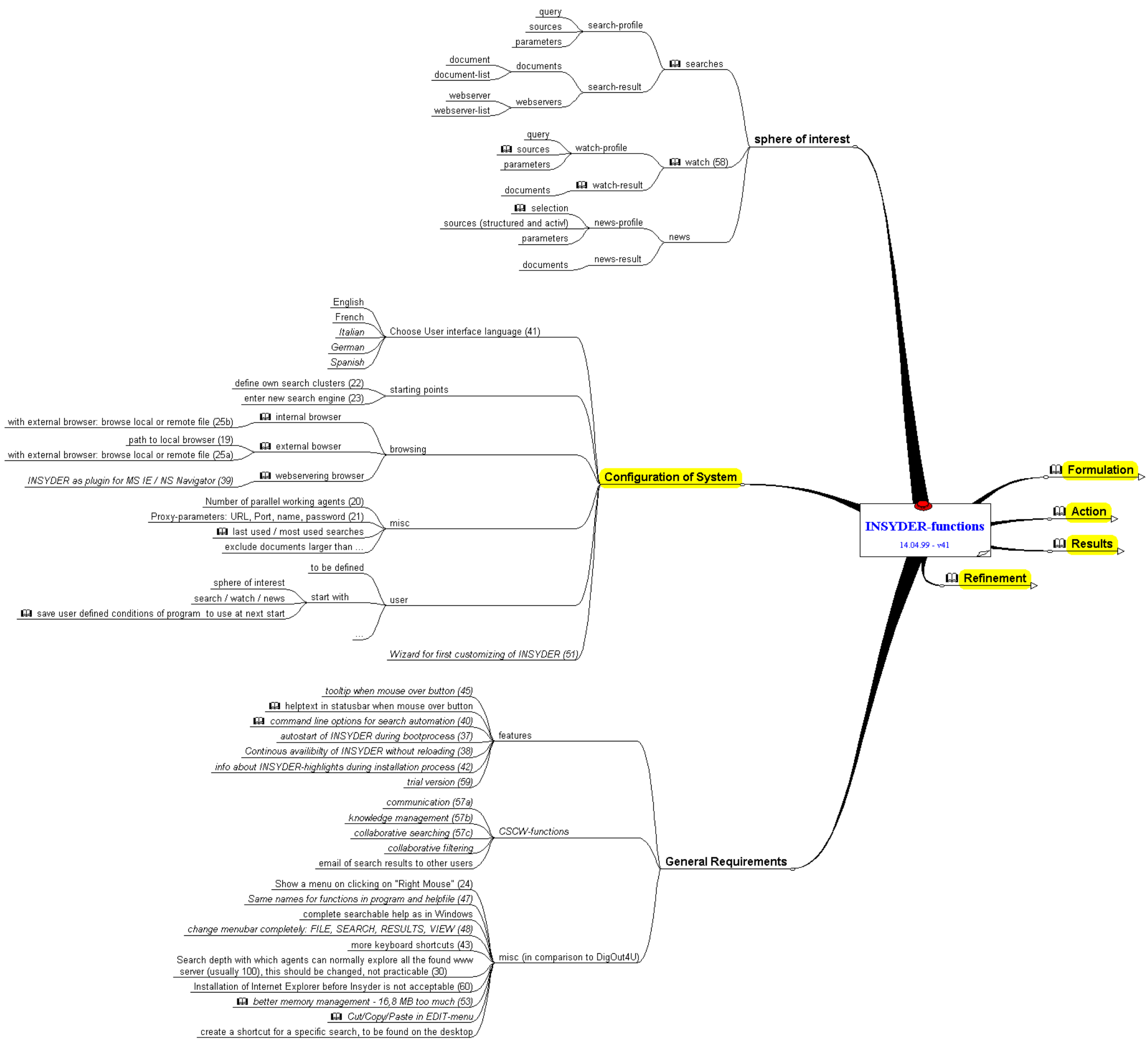


Figure 227: Planned INSYDER functions SOI, configuration of system, general requirements

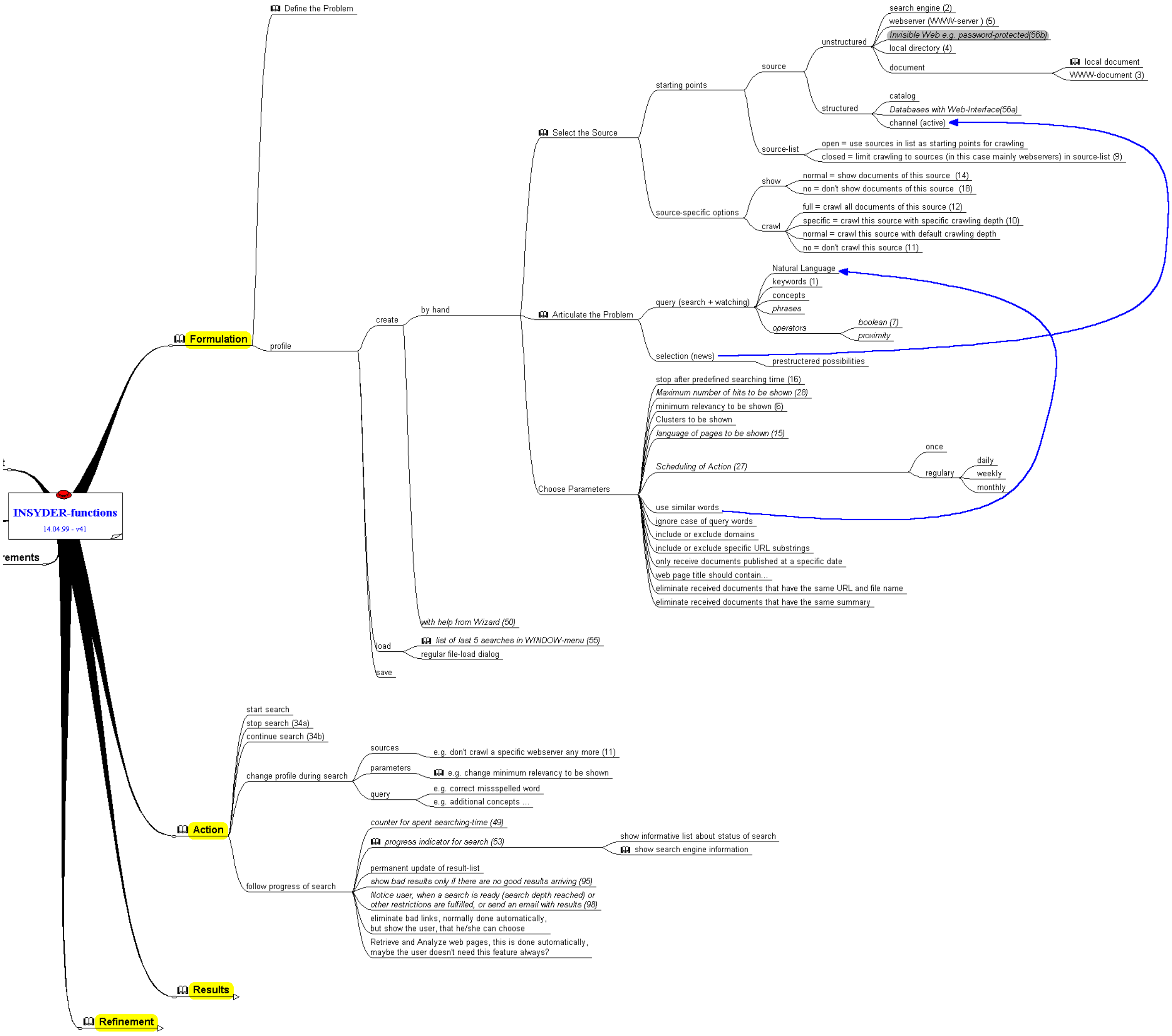


Figure 228: Planned INSYDER functions: Formulation + Action

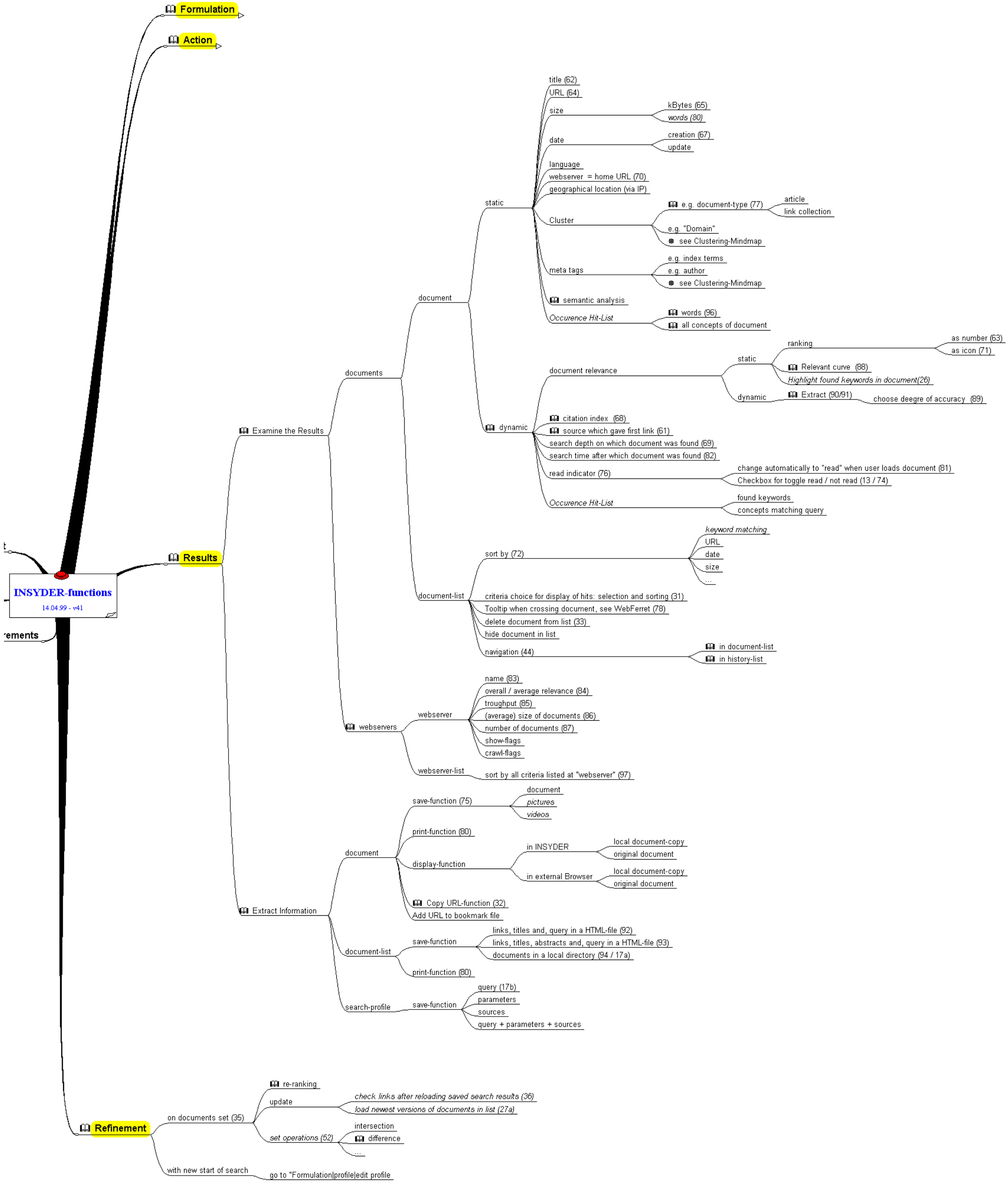


Figure 229: Planned INSYDER functions: Results + Refinement