VISUALIZATION SUPPORT FOR MANAGING INFORMATION OVERLOAD IN THE WEB ENVIRONMENT

By

OZGUR TURETKEN

Bachelor of Science
Middle East Technical University
Ankara, Turkey
1994

Master of Business Administration
Middle East Technical University
Ankara, Turkey
1999

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
In partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
May, 2001

VISUALIZATION SUPPORT FOR MANAGING

INFORMATION OVERLOAD IN

THE WEB ENVIRONMENT

Thesis Approved:

Thesis Adviser

Dean of the Graduate College

PREFACE

The promise of the information age is for the information user to access the highest quality information in the right form, at the right time, and right place. Information technologies have proliferated at an unprecedented rate and are serving the needs of information users better than ever. However, this proliferation has led to an information overload. The information overload problem has adverse impacts on the use of information and the quality of the decisions based on the available information. This research focuses on the information overload problem on the Internet, and proposes a potential remedy to the overload problem encountered while searching the Web.

In this study we developed a prototype system that makes use of clustering and visualization for browsing the results of a typical Web search. This prototype is based on the idea of visualizing Web search results by organizing them into a hierarchy according to their individual contents. This system presents a visual overview of the groups in this hierarchy, and lets its users focus (zoom) on specific groups of interest. We used two different zooming methods (full zoom vs. fisheye zoom), and empirically compared their success with each other as well as the traditional non-visual presentation method by means of an experiment. We hypothesized that the visual systems would lead to higher success than the text-based system, and that the fisheye zooming system would lead to higher success than the full zoom system. The results of our data analyses provide partial support to our hypotheses since the specific system(s) (visual vs. text-based, fisheye zoom vs. full zoom) that a user used caused a significant difference in the speed that (s)he performed the experimental tasks. The data analyses results as mentioned above and the comments made by the experimental subjects suggest that our design ideas were found promising by the users, and it is worthwhile to focus on improving the implementation.

ACKNOWLEDGEMENTS

I am sincerely grateful to my advisor, Dr. Ramesh Sharda, for his invaluable supervision and support that has been continuing since I first met him. Without his intelligent comments and guidance, this project would not have been successful. My appreciation also extends to the other members of my Ph.D. committee, Drs. Marlyn Kletke, Rick Wilson, and Manjunath Kamath for their insightful comments at various stages of this dissertation. I owe special thanks to my other committee member, Dr. Mark Gavin, for his efforts in guiding me through the methodological and statistical parts of this study.

I would also like to acknowledge the tremendous help that I received from two talented people, Mr. Someshwar Baldawa, and Mr. Vishal Jangala, in the development of the system reported in this dissertation. Thanks also go to Mr. Sanjay Vanketashvarullu for his contribution thereof.

Many people at Temple University helped in the later stages of this study. I would like to specifically thank Dr. David Schuff of the MIS Department for his help in the implementation of the system. School of Business professors, Dr. Anat Hovav, Dr. Kevin Dow, and Mr. John Livezey also deserve special thanks for their support in the experimental part of the study.

Finally, I would like to thank my family for their everlasting support, a special person, Miss Jessica Evans, for her understanding and continuous encouragement, and

those colleagues, who have been generous in providing me feedback throughout the countless presentations from this study.

TABLE OF CONTENTS

Chapter | Page

1. INTRODUCTION 1

1.1 Overview 1

1.2 Information Use and Overload on the WWW 10

1.3 Information Visualization 13

1.4 Organization of the Dissertation 17

2. REVIEW OF RELATED WORK 18

2.1 Clustering Based Information Browsing and Search 19

2.2 Visual Information Retrieval (IR) Systems 26

2.3 Distortion-Oriented Systems 43

2.4 Summary and Research Directions 48

3. METHODS 53

3.1 Prototype Design 53

3.1.1 Query Formulation and Search 54

3.1.2 Document Representation 54

3.1.3 Document Grouping (Clustering) 59

3.1.4 Visualization 62

3.1.5 Implementation 74

Chapter			Page
	3.2	Hypotheses	81
	3.3	Experimental Design	83
		3.3.1 Subjects	83
		3.3.2 Tasks	83
		3.3.3 Control Variables	90
		3.3.4 Measurements	90
		3.3.5 Procedure	93
4.	RESULTS		103
	4.1	Preliminary Analyses	103
		4.1.1 Data Representation	103
		4.1.2 The Experimental Design Revisited	110
	4.2	Analysis of the Revised Data	119
		4.2.1 MANOVA Assumptions	119
		4.2.2 Hypotheses Testing	126
	4.3	Post Hoc Analyses	133
5.	CONCLUSIONS AND DISCUSSION		141
	5.1	Brief Summary	141
	5.2	Contributions of the Study	142
	5.3	Discussion and Directions for Future Research	145
		5.3.1 The Prototype System	145
		5.3.2 The Empirical Study	149
REFERENCES			153

Chapter	Page
APPENDIX A	165
APPENDIX B	167
APPENDIX C	169

LIST OF TABLES

Table		Page
3.1	Roussinov's experimental questions	86
3.2	Shneiderman's extended fact-finding questions	87
3.3	The extended fact-finding questions	88
3.4	The experimental tasks	89
3.5	The Initial Questionnaire	92
3.6	The Satisfaction Survey	93
3.7	The modified list of the experimental tasks	95
3.8	The Experimental Design	102
4.1	The Repeated Measures Design	111
4.2	Summary of the Data Analyses Results	138

LIST OF FIGURES

Figure		Page
1.1	Information Search with Clustering and Visualization Support	4
1.2	The Conceptual Model	9
2.1	The Scatter/Gather Interface	20
2.2	Main Results Page in Grouper	21
2.3	The outline view of the current implementation of Cha-Cha search on the query "earthquake"	24
2.4	Search Results from "Sense Maker"	25
2.5	Search Results from "Sense Maker"	26
2.6	Details and Context Integrated in "WebTOC"	30
2.7	Details and Context Integrated in "WebTOC"	30
2.8	An example WebBook	32
2.9	The Web Forager	33
2.10	The PDQ Tree-browser	35
2.11	A Cone Tree	36
2.12	Zooming of a Cone Tree	37
2.13	The Cat-a-Cone Interface	38
2.14	The ET Map	40
2.15	3-D FSN-File System Navigator	41

Figure | Page

2.16	Webspace© Viewer	42
2.17	An organization chart	45
2.18	Changing focus on the organization chart in Figure 2.17	46
2.19	The perspective Wall	46
2.20	Information Search with Clustering and Visualization Support	50
3.1	The ET Category Map (Chen et al. 1997)	63
3.2	The Music Category	64
3.3	"Hyperbolic Tree" of Library of Congress Library	65
3.4	Zoomed View of the Copy Right Office	66
3.5	A simple hierarchy	68
3.6	The treemap of the hierarchy in Figure 3.5	69
3.7	The Original Treemap Drawing Algorithm by Johnson and Shneiderman (1991)	72
3.8	Extended Features	73
3.9	The input form	75
3.10	Flowchart of the Process for Visualizing the Hierarchy of Documents	76
3.11	The Visual Overview of Documents Returned in Response to the Query "Name five different kinds of music that Sony is publishing."	78
3.12	Flowchart of the Process for Fisheye Zooming a Part of the Visualization	79
3.13	The System Flowchart	80
3.14	The Research Model	91
3.15	Text-based Presentation of Search Results to Query 2	97

Figure		Page

3.16	Visual Overview of the Search Results to Query 2	98
3.17	"Full" zooming of the "populous world" section of Figure 3.16	99
3.18	"Fisheye" zooming of the "populous world" section of Figure 3.16	100
4.1	Reliability Calculations (SPSS Output)	106
4.2	Factor Analysis Results	108
4.3	Preliminary Analyses for the Comparison of the Groups	112
4.4	Comparison of the Groups with "Full Zoom"	114
4.5	Comparison of the Groups with "Fisheye Zoom"	116
4.6	Correlation between the Dependent Variables	121
4.7	Plots for "No Visualization"	123
4.8	Plots for the "Full Zoom" Visualization	124
4.9	Plots for the "Fisheye Zoom" Visualization	125
4.10	Hypotheses of the Study	126
4.11	The factors of the model	127
4.12	Descriptive Statistics for "Score"	128
4.13	Descriptive Statistics for "Time"	129
4.14	Descriptive Statistics for "Satisfaction"	129
4.15	Multivariate Tests	130
4.16	Univariate Tests	132
4.17	Paired Comparisons	133
4.18	Descriptive Statistics for the Modified Sample	134
4.19	Multivariate Tests for the Reduced Sample	135

Figure		Page
4.20	Univariate Tests for the Reduced Sample	136
4.21	Paired Comparisons for the Reduced Sample	137

1. INTRODUCTION

1.1 Overview

People look for the highest quality information possible for making good decisions. Today's computers are becoming gradually faster, and together with the communication infrastructure they facilitate the connectivity of a myriad of information sources across physical and geographical boundaries. However, this infrastructure is still not capable of presenting the "right information" to the "right person" at the "right time" and in the "right form". The challenge is that we have connectivity, but not interactivity and integration. This general issue has been observed by researchers from various disciplines, and led to a recent NSF (National Science Foundation) program on collaborative research, called "Knowledge Networking" (1997-1999).

The idea behind knowledge networking is to extend the traditional knowledge management objective of high connectivity for information sharing so that there is more emphasis on interactivity and integration for the creation, accumulation, and use/reuse of knowledge. The term "knowledge" is used rather loosely in this context where it may refer to information (content) as well as meta-information (high level information about content), procedural knowledge (i.e. know-how), and reasoning knowledge (know-why). We mainly refer to content in this study. Hence, we use the term "information". On the other hand, management (i.e. creation and accumulation) of information is not fully isolated from that of the other forms of knowledge, therefore we keep using the term "knowledge management".

A critically important issue in this broader definition of knowledge management is information overload. Information users' tendency to get the best information possible may lead them to retrieve all the information that is potentially related to a specific need. The combination of such aggressive information sharing effort and high connectivity to large knowledge repositories may jeopardize certain aspects of information quality such as relevance. This problem, commonly known as "information overload", occurs when an information user is exposed to more information than (s)he needs, and more importantly, is able to process.

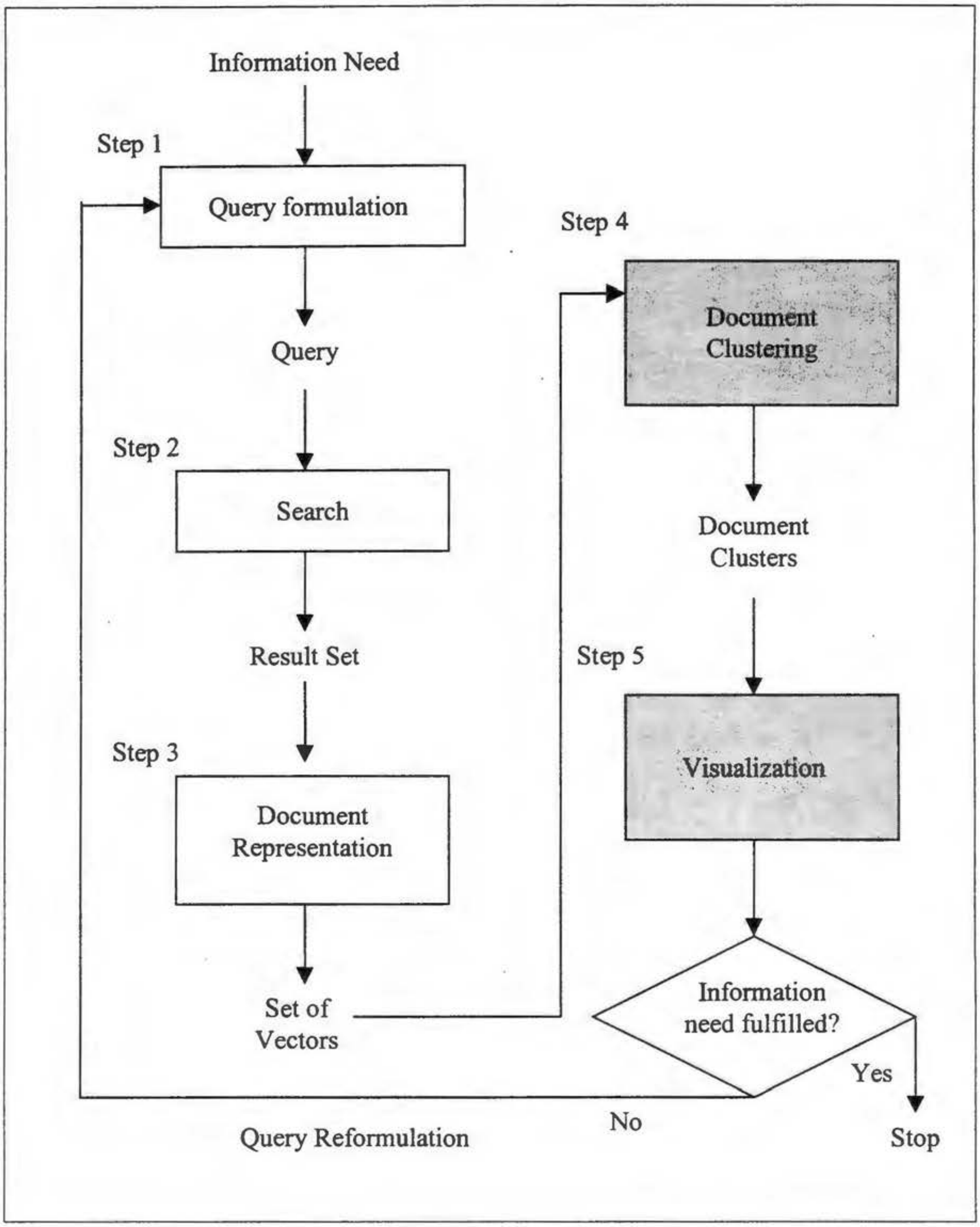
Information overload has adverse impacts on information use regardless of the type of information and decision making domain. The problem has been studied in accounting (Bright 1996; Chewning and Harell 1990), finance (Mattlin 1992; Setton 1997), hospital management (Hunt and Newman 1997; Johnsson 1991), banking (Johnson 1997), hotel management (Worcester 1997), and general e-mail management (Rudy 1996), among others. Early in the age of information support systems, Denning (1982) pointed out that "The visibility of personal computers, individual workstations, and local area networks has focused most of the attention on generating information – the process of producing documents and disseminating them. It is now time to focus more attention on receiving information -- the process of controlling and filtering information that reaches the person who must use it." Early research in marketing found that as the amount of information available to people increased, the accuracy of their decisions decreased (Malhotra 1982, Keller and Staelin 1987). Beyond a relatively low quantity of information, people will begin to filter out information that they use to make decisions (Jacoby 1984), increasing the probability of bypassing important information. This brief discussion makes it clear that knowledge management needs to address the information overload problem, especially considering that the problem will only be more serious with the fast growth in connectivity and access to information.

Recently, the Internet and particularly the World Wide Web (WWW) is emerging as a frequently used source of information. WWW is arguably the most prominent platform for global communications with the various kinds of hypermedia forms it supports. Consequently, the information collected from the Web could be in various formats. Due to this heterogeneity and the large size of the Web, management of Web-based knowledge, especially dealing with the problem of information overload deserves special attention. In this study, we address a specific information overload problem encountered during a Web search. Typically, Web search engines present their results as a ranked list. For broadly formulated search queries, such a list may contain thousands of documents. Research has suggested that the users of search engines are not likely to go beyond the top 20 to 30 documents on these lists before they get bored or frustrated, and subsequently quit the search (Roussinov 1999). In fact, according to a user study by Excite Corporation, less than 5% of the users looked beyond the first screen of documents returned in response to their queries (Wu 1997). Consequently the chances of reaching the relevant information are reduced when the searcher is overloaded with the irrelevant documents at the top of the list. There is an obvious need for research on the alleviation of this problem. This study reports on our research effort for that purpose.

We developed a prototype system that aims to address the above-mentioned information overload problem by using clustering and information visualization. The prototype design is based on a simple five-step model of information search with

Figure 1.1

Information Search with Clustering and Visualization Support



clustering and visualization support (See Figure 1.1). The focus of the study is the use of a specific visualization method that has not been fully studied for exploring Web search results before. More detailed discussion of the steps in this model is presented in later chapters.

One of the central themes of this research is the combined use of clustering and visualization, both of which can be used individually to reduce different aspects of information overload. Clustering is a well-known method commonly used to identify patterns in an unstructured group of objects. Clustering has its use in various application domains such as market segmentation in traditional market research and more recently in data mining. A very detailed discussion on the concept of clustering and a review of clustering techniques are beyond the scope of this research, yet we revisit the topic including more relevant detail in the later parts of the dissertation.

Information visualization is the common name for a group of techniques that use the idea of supporting the cognitive system by means of visual cues for better and quicker understanding of information (Shneiderman 1996). Some well-known applications of this idea in our domain of interest are the use of visual maps to represent directory structures (Johnson and Shneiderman 1991, Chen et al.1997) and the visual representation of documents returned by a search query (Hearst 1995).

The visual system that we have developed presents an overview of search result clusters instead of a linear ranked list of individual documents. The clustering (grouping) process is based on the semantic content of the documents, hence the resulting groups can be deemed semantically formed. The overview of these clusters summarizes the information space and lets its viewers recognize certain patterns. Based on such an understanding, the information searcher can focus on the document groups of more interest. Our proposition is that this approach would not only help in finding the relevant information in the whole collection, but also in finding this information fast. In other words, the proposition is that this approach would provide better information access with less overload. In that respect, our aim has been achieving high search success by means of increasing search effectiveness and search efficiency. This use of the success concept refers to the better organization of information rendering it possible to display more information with less information overload. At this point, it should also be clarified that our use of the term "search" assumes a user, rather than a system perspective. Although the system we are describing does not aim to improve the available search algorithms per se, it aims to enhance the outcomes of the users' search efforts. Hence, the terms effectiveness and efficiency are used regarding these outcomes.

It has been observed that the scarcity of empirical studies on the usefulness of information visualization is a weakness of research on information search, especially in the Web domain (Zamir 1998, Roussinov 1999). Accordingly, an important feature of our study is the empirical testing of our prototype to discover whether our design ideas fulfill the aim of high search success. As described before, this research aims to enhance the general understanding on the success of different information presentation methods in alleviating information overload that one faces in the exploration of Web search results. Because the Web is a general platform, we contend that the insights gained from empirical studies in the Web domain such as the one reported here are generalizable to narrower domains such as company Intranets or more structured database systems.

The effect of an information presentation method on the success of human-computer interaction in more traditional domains has been well studied. Early information systems research identified the problems with experimental research on human-computer interaction as the lack of strong theoretical models and research designs (Jarvenpaa et al. 1985). However, as Jarvenpaa et al. (1985) state, later efforts in discovering the superiority of a certain presentation method over others (e.g. tables vs. graphs) have suffered from the same limitations, and have been mainly inconclusive. Today's research on the topic makes use of more advanced computer graphics than those in the last two decades, yet this advance does not overshadow the importance of rigor in experimental studies. This study employed such rigor by basing the research model on past theory and conceptual studies.

General success of an information system can have many dimensions such as system quality, information quality, use, user satisfaction, individual impact, and organizational impact. Different researchers have used differing sets of these measures depending on the purpose of their study (DeLone and McLean 1992). From a user-oriented perspective, system success can be defined in terms of the success of the end-user performing certain tasks (performance) by using the system (individual impact), and their satisfaction with the system (user satisfaction). End-user success, in turn, has two dimensions: effectiveness and efficiency. Effectiveness of an information system user is a measure of how correctly (s)he performs, i.e. how desirable his or her outcomes are. On the other hand, efficiency refers to how well (s)he uses the available inputs (physical resources or time) in producing those outcomes. In other words, efficiency is the ratio of the outputs to the inputs used by a user.

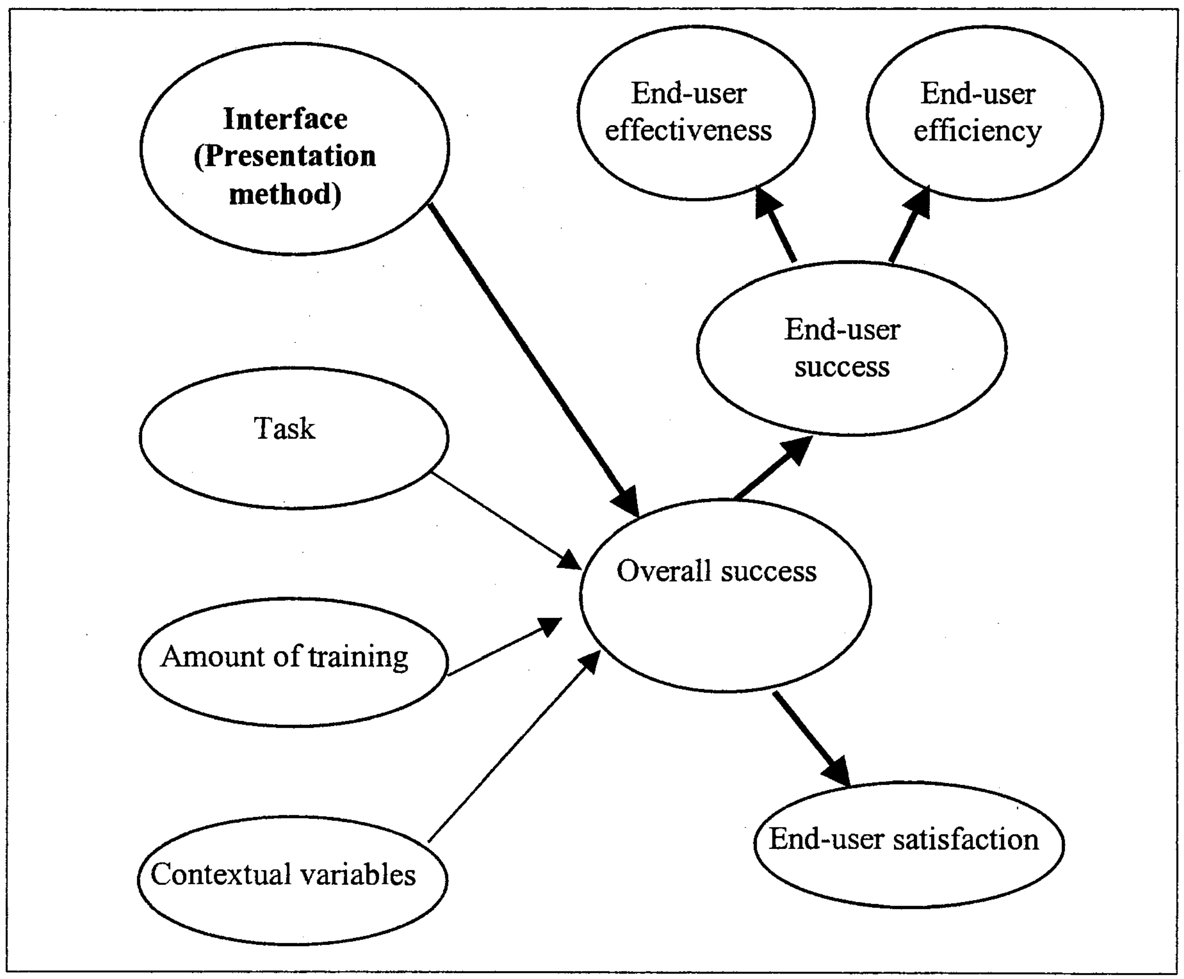
Previous work in human computer interaction has identified a number of factors affecting the success and satisfaction of the end users of an information system. Among these factors are the interface (Suh and Jenkins 1992, Santhanam and Sein 1994), the characteristics of the task such as its level of difficulty (Suh and Jenkins 1992), amount of training (Suh and Jenkins 1992, Santhanam and Sein 1994), contextual variables such as individual differences and experience (Santhanam and Sein 1994), and the interaction between some of these factors such as cognitive fit (Vessey 1991), or task and interface match (Tan and Benbasat 1990).

In this specific study, we take a user-oriented approach in defining the overall success in performing specific information search tasks using an information presentation system. Based on the main success factors that have been identified by past research that were mentioned in the above paragraph, we propose that the overall success in the performance of specific search-related tasks is determined by the following: the success of the interface in presenting the information content (with little information overload), the tasks themselves, the amount of end user training, and contextual variables such as individual differences and experience. Subsequently, the overall success leads to end-user success (effectiveness and efficiency) and end-user satisfaction. Figure 1.2 depicts these causal relationships that constitute the conceptual model of our study.

As mentioned previously, a major objective of this study is to test the effect of the presentation method on the overall success in Web search related task performance under one instantiation of the "task" and "amount of training" variables in the conceptual model. The discussion on the operationalization of the independent and dependent variables in the model is presented in Chapter 3. Similarly, the system development and testing efforts are further detailed in Chapter 3. Meanwhile, there is a need to better understand the problem addressed by the study. Accordingly, the following section presents a brief summary on the information retrieval activities on the WWW and related problems, and provides a background for a detailed discussion on our specific problem.

Figure 1.2

The Conceptual Model



1.2 Information Use and Overload on the WWW

The two basic information retrieval activities on the WWW are hypertext browsing and keyword-based information search (hereafter referred to as information search).[1] Hypertext browsing starts at a page that was previously known or is suggested by a reference (e.g. an expert), and continues by following the interesting links thereafter. People browse the Web when they are able to recognize their information needs but cannot comfortably describe them in appropriate terms. On the other hand, information search is performed when the information need is better defined; at least to a degree that a query based on a keyword, a phrase or a combination of these can be formulated. Both of these information-seeking activities have shortcomings leading to poor information sharing and high information overload as explained next.

The Web contains hundreds of millions of pages today, and the number of these pages is growing at a tremendous rate (it is argued that the size of the Web doubles every six months!). Hypertext browsing can only cover an extremely small portion of this whole Web space probably skipping a lot of relevant information. Moreover, due to occasional mismatches between the Web designers' organization of the information and the users' perception of this organization, an information seeker can reach a page that (s)he has not planned on. This causes confusion and disorientation problems, a phenomenon usually described as "being lost in the cyberspace".

Information search has its own problems leading to ineffective retrieval. Two metrics commonly used in information science to assess retrieval effectiveness are "precision" and "recall". The term "precision" refers to the ratio of the number of

---

[1] The term "information retrieval" refers to both browsing and search, and will be used accordingly throughout the text when there is no need to differentiate between the two activities.

relevant pages to the number of total pages retrieved. "Recall" is the ratio of the number of relevant pages retrieved to the total number of relevant pages in the whole data collection. These two performance measures, originally defined for structured and relatively small databases, are hard to precisely measure in the Web context, because the total number of relevant pages for a given information need is almost impossible to determine. Considering the enormous number of results in response to some search queries, the same is true even for the number of relevant pages within a group of search results. Hence, the use of these conventional metrics for the Web should be taken as general approximations rather than precisely measurable entities. Using these terms to this effect, Chen et al. (1998) point to the fact that Web search results in low precision and poor recall. Another problem is slow response due to the limitations of indexing and communication methods (bandwidth). Also, it is known that different people tend to use different words in expressing similar concepts whereas computer programs use controlled language-based interfaces assuming that every user will chose to use the same words for expressing the same information need and this leads to the so-called "vocabulary problem". Furthermore, the interfaces of most search engines require that the users put their needs in a computer-understandable format such as a boolean query. However, a lot of users are not sophisticated enough to fully articulate their needs in such a format, and the result is yet another information retrieval problem (Chen et al. 1998).

There is ongoing research on many aspects of the aforementioned problems. Limited bandwidth and low speed are general network problems independent of the specific form of communication. These problems are addressed by the broad research on telecommunication networks. Efforts on possible remedies for the information search problems focus on more efficient ways of information filtering (e.g. De Bra et al. 1997) to pursue higher precision in search results. Filtering is useful in increasing the precision of search results especially on relatively small-scale databases. However, by its very nature, it does not help in the improvement of recall on the Web. That is, filtering the irrelevant information from the search results increases the ratio of the relevant information to the irrelevant, but does not increase the chance of retrieving more of the potentially relevant information. This remaining problem is addressed by research on Web indexing, search algorithms (Chakrabarti et al. 1996, Fox et al.1999) and query formulation (Savoy 1997, French et al. 1997). A recent and detailed study on Web search algorithms, and a comparison of the major commercial search engines can be found in Gordon and Pathak (1999).

Our research focuses on improving the methods that are used to examine information search results. It can be argued that once the search results are collected, the exploration of this collection is a browsing task. Following this viewpoint, we propose that search success can be improved by means of superior browsing of search results.

The recent advances in information processing speed and the graphical capabilities of today's powerful computers have made it possible to support cognitive tasks such as scanning, sorting and selection by means of perceptual (visual) aids in a computerized environment. Visualization is a general name given to the use of these aids. The same cognitive tasks of scanning, sorting, and selection are also performed in post-retrieval information exploration (exploration of documents retrieved by a search engine) suggesting that visualization can be used as a support for the post-retrieval phase of information search. In the next section, we discuss the basics of information visualization in more detail in order to understand its applicability to our problem.

1.3 Information Visualization

Information visualization aims to present a collection of information by providing visual cues so that it will be possible to process the information by the (visual) perceptual system instead of solely depending on cognition. It is argued that the perceptual system operates in a time range of 10 to 100 milliseconds whereas this range for the cognitive system is from hundreds of milliseconds to a few minutes (Brautigam 1996). Visualization takes advantage of the fact that information assimilation will be faster by several orders of magnitude if the initial processing of information can be offloaded from the cognitive system to the perceptual system (Brautigam 1996). Humans can quickly understand the relative position of the different entities and their relationships in a picture. "Interface designers can capitalize on this by shifting some of the cognitive load of information retrieval to the perceptual system. By appropriately coding properties by size, position, shape, and color, we can greatly reduce the need for explicit selection, sorting, and scanning operations" (Shneiderman 1994). Two important classes of visualization paradigms have been studied: scientific visualization and information visualization. Both of these paradigms share the same basic principle of using perception in support of cognition. Information visualization is different from scientific visualization in that it aims at revealing the relatively abstract relationships in multidimensional data. An example of this would be the display of demographic trends in a certain part of the world. Scientific visualization, on the other hand, is about data that are already low dimensional, but still need elaboration to be clearly perceived, for example the molecular structure of an organic tissue where the entities physically exist (in three dimensions) but cannot be observed by the naked eye. By this token, information visualization focuses on understanding multi-dimensional and implicit relationships whereas the aim of scientific visualization is to make it easier to understand the relationships between physical entities. Information visualization is a dimension-reduction activity and hence is inherently involved with summarization of complex data. Research points to the usability of this approach in human computer-interaction, and as mentioned before, our own conviction is that information visualization is a potential remedy to the specific information overload problem we are studying.

Shneiderman (1996) points out to the under-utilization of people's perceptual abilities in the design of user interfaces and suggests that there is room for much improvement in this respect. He summarizes the basic principle of visual design as the "Visual Information Seeking Mantra", which is a sequence of the following basic tasks: "overview" first, "zoom" and "filter", then get "details on-demand". His well-known "task by data type taxonomy" that consists of seven tasks and seven data types is an amendment of this basic principle. The seven tasks that are at a high level of abstraction are:

<u>Overview:</u> Gain an overview of the entire collection

<u>Zoom:</u> Zoom in on items of interest

<u>Filter:</u> Filter-out uninteresting items

<u>Details-on-demand:</u> Select an item or group and get details when needed

<u>Relate:</u> View relationship among items

History: Keep a history of actions to support undo, replay and progressive refinement.

Extract: Allow extraction of sub-collections and the query parameters.

The seven data types are 1-dimensional, 2-dimensional, 3-dimensional, temporal, multi-dimensional, tree, and network.

The list of the data types presented in this framework is subject to variation and Shneiderman notes that many prototypes use a combination of these data types. This taxonomy covers both scientific visualization and information visualization, and its purpose is to facilitate discussion leading to useful discoveries. Other work on visualization suggests a good amount of evidence to the fulfillment of this purpose.

When our topic of interest, i.e. visual information browsing, is examined through this framework, the initial task can be defined as the reduction of the number of implicit dimensions (terms or concepts) in data to two or three thus making the relationships "visible". This results in a visible overview (summary) of the information. As will be seen by example of the systems in the next chapter, there are a number of different paradigms as to how such an overview can be presented. Yet, regardless of the kind of the overview, certain subsets of the information space will be of more interest to the viewers of the presentation, and these portions will be zoomed.

Visual systems also differ as to the kind of zooming capabilities they provide. Our study focuses on two alternative zooming methods, namely a full zoom system where the irrelevant (i.e. not immediately relevant) portions of the overview are filtered out and a fisheye zoom system where the irrelevant portions of the overview are summarized to provide context.

The full zoom approach provides the immediately needed information in sufficient detail but eliminates the less relevant information. Yet this "less relevant" information forms the global perspective in which the detailed information is useful. Hence information may be more useful if organized and presented accordingly. Furnas (1986) introduced the concept of generalized fisheye views based on a similar observation: "humans often represent their own 'neighborhood' in great detail, yet only major landmarks further away. This suggests that such views ('fisheye views') might be useful for the computer display of large information structures like programs, data bases, online text, etc."

Since their introduction in 1986, fisheye views have found applicability in displaying information structures such as hierarchical tables (Remde et al. 1987; Egan et al. 1989), computer graphs (Sarkar and Brown 1992), and hypertext (Feiner et al. 1982; Feiner 1988; Noik 1993, Collaud et al. 1995, Bederson et al. 1998). Some of the main application domains have been groupware (Greenberg et al. 1995, Gutwin and Greenberg 1997), and monitoring systems (Schafer et al. 1998). Schafer et al. (1998) and Leung and Apparley (1994) present a summary of selected fisheye visualization systems and the enabling methods.

Traditionally, most of the visual information retrieval systems have used full zoom[2]. However, the concept and applications (as briefly mentioned above) of fisheye views is promising thus motivating us to use the idea in the Web search domain. The basis of this motivation is elaborated in the next chapter where we review different visual systems and their characteristics.

---

[2] A detailed discussion takes place in the next chapter

1.4 Organization of the Dissertation

The organization of the rest of the dissertation is as follows: in Chapter 2 we review the literature addressing the problems that we identified in this chapter, and accordingly formulate our research questions. Chapter 3 discusses the methods used in the pursuit of the research objectives. Chapter 4 includes empirical results and their analysis while Chapter 5 is devoted to the discussion of the results, contributions of the study, conclusions, and directions for future research.

2. REVIEW OF RELATED WORK

In Chapter 1, we outlined the information overload problem that this study focuses on, and discussed some of the past efforts towards a solution for the problem. We also indicated that our approach to the alleviation of the problem is the use of clustering (grouping) and visualization. In this chapter, we review the literature on the use of grouping and visualization for knowledge management especially addressing the problem of information overload. Most of the studies covered here are about the Web while others are about traditional methods of information retrieval, yet are still applicable to the Web domain.

The growth in Web related research is almost commensurate to the growth in the Web itself. Especially with the new generation of fast PCs and the huge interest in Web based business activities and related software applications, methods for effective information presentation are needed. There is a fast growth in the number of systems including visualization based information retrieval systems that are developed in response to this need. For this reason, a comprehensive coverage of such systems is not feasible hence the review in this chapter is mostly limited to representative applications of the main ideas rather than a survey of the state of art in the field.

The chapter is organized as follows: Section 2.1 reviews previous work based on document clustering in information retrieval. Section 2.2 gives an overview of systems that aim to take advantage of the visualization idea, and share many of the visual principles discussed in Section 1.3. In Section 2.3, we look at distortion-based visual systems that aim to improve over the ones in Section 2.2 by means of the zooming capabilities they include. The last section of the chapter summarizes the previous literature and describes the research directions pursued in this study. Review of the relevant literature continues throughout the later chapters as necessary.

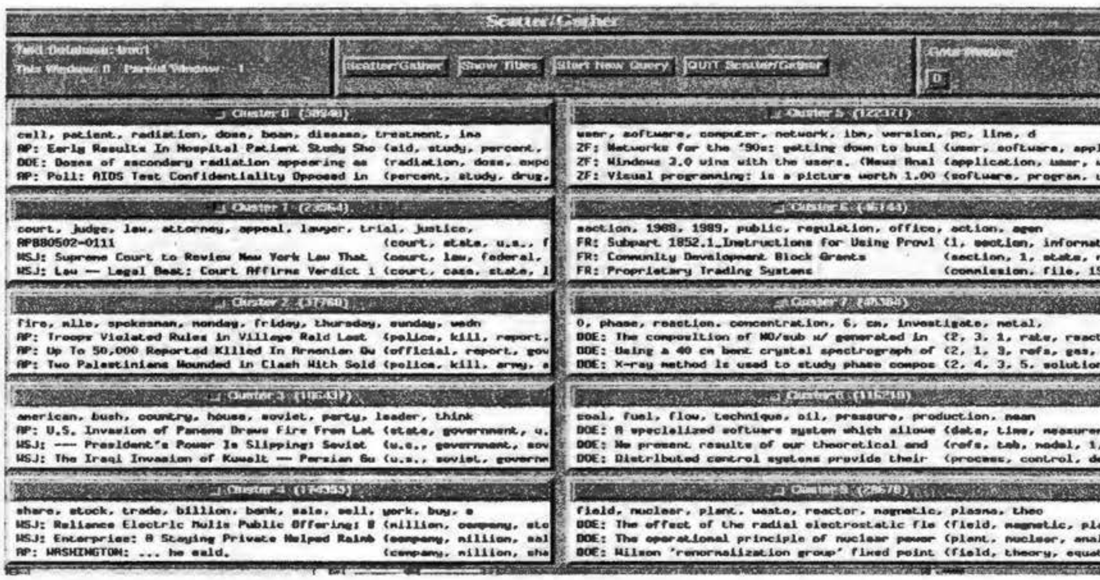
2.1 Clustering Based Information Browsing and Search

As mentioned previously, our study proposes a clustering-based approach to the organization of Web search results. In this section we review previous work that used clustering in information search. While doing this, we put special emphasis on the use of clustering for post retrieval exploration of search results since this is of specific interest to our study.

The cluster hypothesis (van Rijsbergen 1979) states that mutually similar documents will tend to be relevant to similar information needs. Hence, clustering can be used to increase search effectiveness. Cutting et al. (1992) were some of the earlier researchers to extend this idea and develop a document browsing method based on clustering. This browsing method known as Scatter/Gather is directed towards a focus set of documents potentially interesting to the user. The focus set is clustered into smaller subsets and summarized to form an outline from which the user can select a smaller focus set. The indicated subcollection becomes the focus set, and the process repeats (Cutting et al. 1993). Figure 2.1 depicts the Scatter/Gather interface in a typical session. Pirolli et al. (Pirolli et al. 1996b) tested the effectiveness of Scatter/Gather as a simple document retrieval tool, and studied its effects on the incidental learning of topic structure. Their basic conclusion was that the Scatter/Gather clustering method improved browsing effectiveness. Hearst and Pederson (1996) revisited the method and applied Scatter/Gather to retrieval results reaching a conclusion that clustering increased both search effectiveness and efficiency. The relevance of the Scatter/Gather studies to our specific study is that the corresponding results provide strong evidence to the successful use of clustering in information retrieval.

Figure 2.1

The Scatter/Gather Interface



The clustering algorithm used in Scatter/Gather is reported as being "nearly linear". For an on-line system that presents Web search results, the speed of clustering is critical. Recent efforts on clustering-based systems focus on this issue. Zamir and Etzioni (1998) introduced a linear time clustering algorithm called Suffix Tree Clustering (STC), and used the algorithm to build an interface called Grouper to display results of the HuskySearch meta-search engine. Figure 2.2 shows the main results page in Grouper for the query "Israel". "Each row in the table is the summary of a cluster – an attempt to convey the content of the documents in the cluster. It includes the size of the cluster, shared phrases – phrases that appear in many documents of the cluster, and up to three sample titles of documents in the cluster. The numbers appearing in parenthesis after each phrase indicate the percentage of the documents in the cluster that contain the phrase. In the example above only the first five clusters are shown" (Zamir and Etzioni 1999).

Figure 2.2

Main Results Page in Grouper

Query: israel
Documents: 272, Clusters: 15, Average Cluster Size: 15.1 documents

Cluster	Size	Shared Phrases and Sample Document Titles
1 View Results Refine Query Based On This Cluster	16	Society and Culture (56%), Faiths and Practices (56%), Judaism (69%), Spirituality (56%); Religion (56%), organizations (43%) • Ahavat Israel - The Amazing Jewish Website! • Israel and Judaism • Judaica Collection
2 View Results Refine Query Based On This Cluster	15	Ministry of Foreign Affairs (33%), Ministry (87%) • Publications and Data of the BANK OF ISRAEL • Consulate General of Israel to the Mid-Atlantic Region • The Friends of Israel Gospel Ministry
3 View Results Refine Query Based On This Cluster	11	Israel Tourism (36%), Comprehensive Israel (36%), Tourism (64%) • Interactive Israel tourism guide - Jerusalem • Ambassade d'Israel • Travel to Israel Opportunites
4 View Results Refine Query Based On This Cluster	7	Middle East (57%), History (57%); WAR (42%), Region (42%), Complete (42%), Listing (42%), country (42%) • Israel at Fifty: Our Introduction to The Six Day War • Machal - Volunteers in the Israel's War of Independence • HISTORY: The State of Israel
5 View Results Refine Query Based On This Cluster	22	Economy (68%), Companies (55%), Travel (55%) • Israel Hotel Association • Israel Association of Electronics Industries • Focus Capital Group - Israel

The STC algorithm creates its clusters based on the short snippets returned by the search engines. The creators of the algorithm observed that the quality of the clusters produced based on these snippets was almost as good as that of the clusters based on whole documents. The authors also empirically tested the interface. The three metrics that were used to compare the Grouper and the ranked list display of the search results are the "number of documents followed", "the time spent traversing the results", and "the distance between successive user clicks on the document set". Analyzing the log files of these two systems, the authors found that the Grouper users followed more documents on the average than the ranked list users. However, the results on "the time spent traversing the results" and "the distance between successive user clicks on the document set" were not in favor of any specific interface.

The contributions of the Zamir and Etzioni (1999) study were the much-needed improvement in clustering speed that the Grouper system provides by means of the STC algorithm and the use of document snippets instead of whole documents. Also the further empirical evidence reported as to the usefulness of clustering in post retrieval document exploration is valuable.

Numerous kinds of information can be used to classify or organize collections of WWW pages. Among such information are the textual content, the connectivity (hyperlink) structure, and various characteristics of the pages including file-system attributes and access statistics, usage statistics (Pirolli et al. 1996b), Web sites that they come from, author, and time of publication (Baldonado 1998). The two systems that we have discussed so far use clustering algorithms based on the similarities of the textual content of the documents. Our prototype followed this approach as well. The next two systems in this section are representative examples of other clustering approaches for us to assess the importance and common use of clustering in information retrieval.

The Cha-Cha System (Chen et al. 1999) uses the hyperlink structure of documents within an Intranet to cluster them. In this approach, an outline of the documents is created by first recording the shortest paths in hyperlinks from the root page to every other page within the Intranet. After a query is issued, these shortest paths are dynamically combined to form a hierarchical outline of the context in which the results reside. Chen et al. (1999) compared their display system to a traditional ranked list system on the basis of performance and user-preference. Their results showed no significant difference on performance yet users tended to prefer the Cha-Cha system to the list display.

The SenseMaker (Baldonado 1998) is another system that helps its users to cluster (bundle) search results along multiple dimensions such as Web site and author. The system has tools to dynamically organize results. Therefore, one can look at different aspects of the document collection. Users of SenseMaker can also add on the structures they have created (add more results in the context of a few specified bundles), or sculpt the structures (eliminate large numbers of results from consideration by removing their enclosed bundles). Figures 2.4 and 2.5 show the results to the search query “volume rendering” “ray tracing” organized into bundles by author and Web site respectively.

Figure 2.3

The outline view of the current implementation of Cha-Cha search on the query "earthquake"

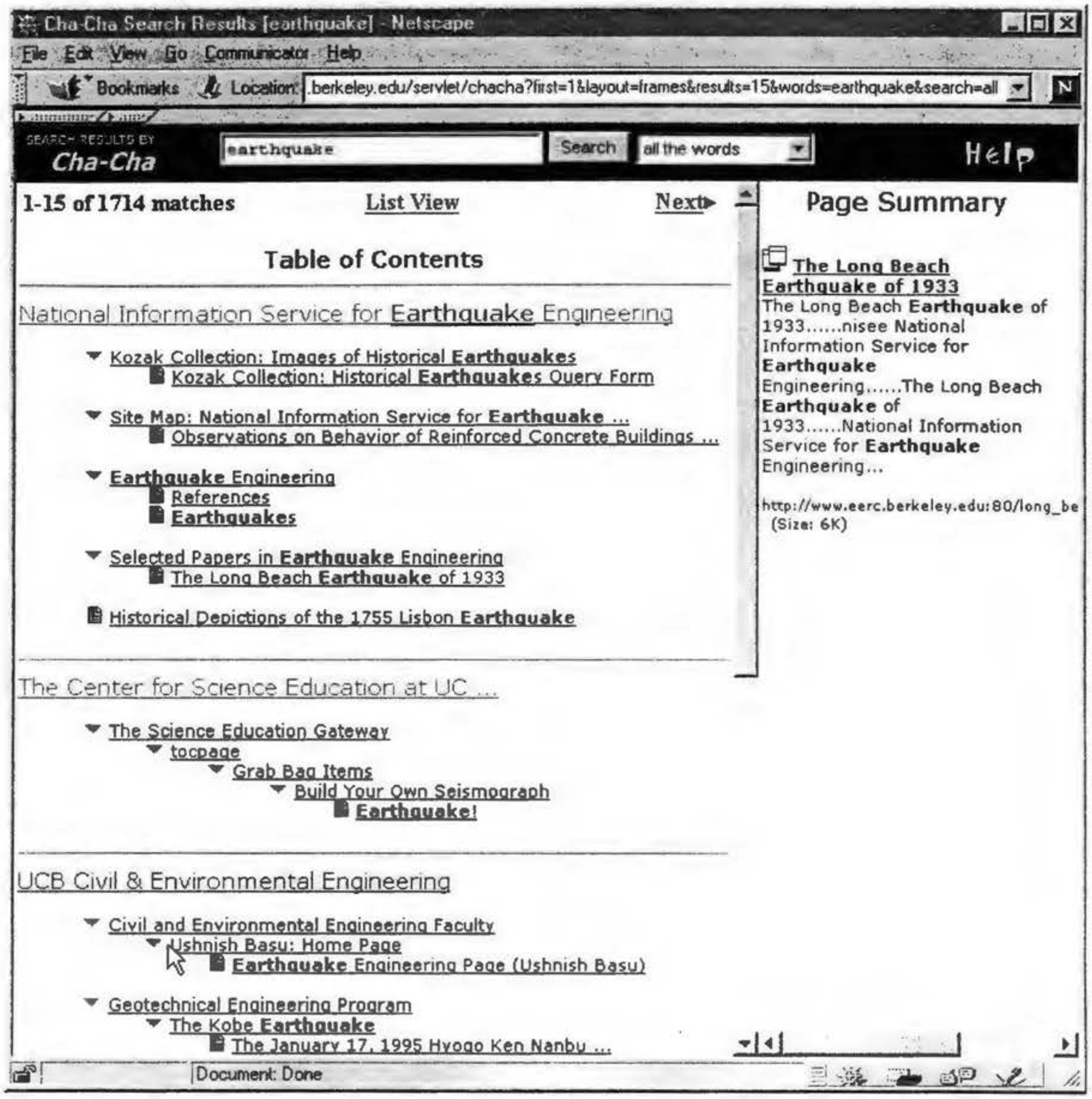
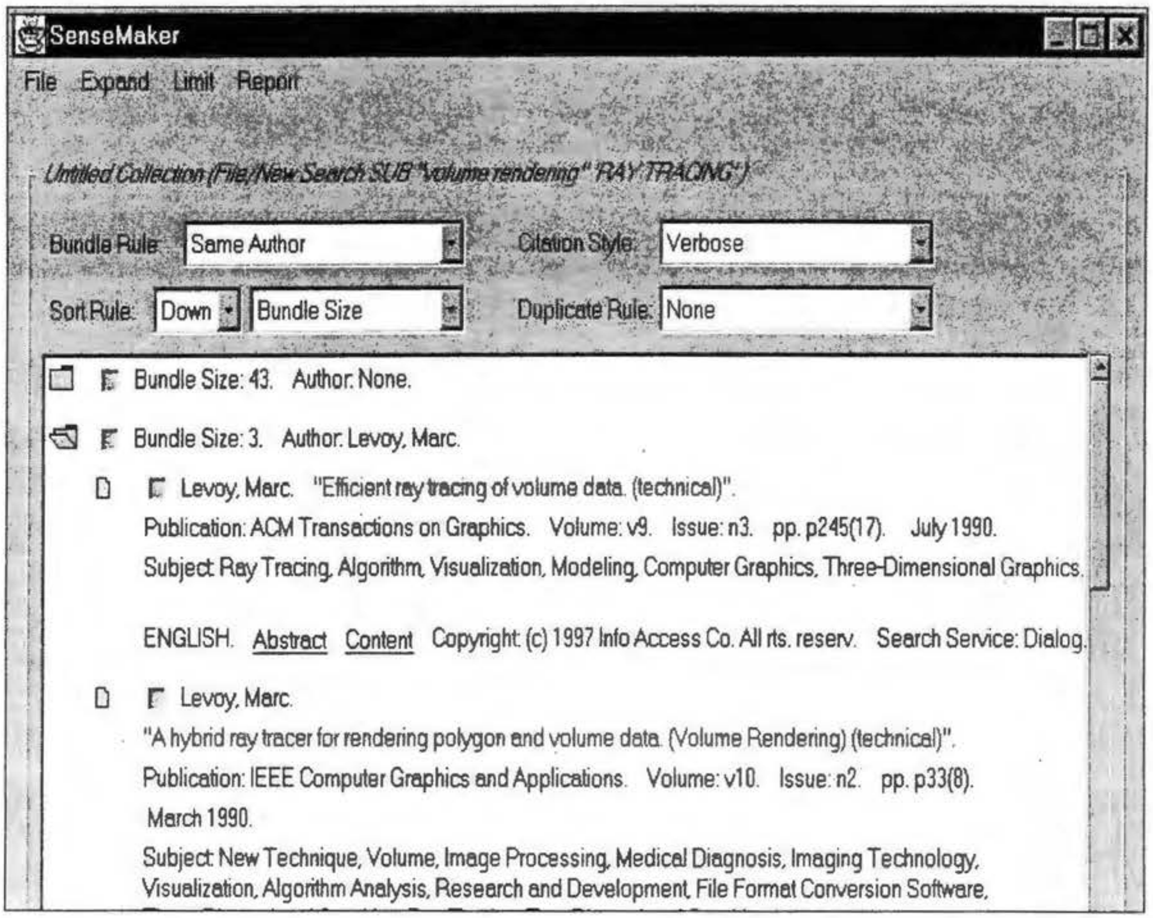


Figure 2.4

Search Results from "Sense Maker"



As seen by example of these representative systems in this short review, clustering is a promising method that is commonly used for reducing information overload.

Figure 2.5

Search Results from "Sense Maker"

2.2 Visual Information Retrieval (IR) Systems

The prototype system that we have developed is based on the visualization of Web search result clusters. Accordingly, our aim is to provide not only summarized information on context, but also visual representations of this context and the relationship between the entities therein. The systems in the previous section only fulfilled the former one of these objectives. In this section, we review visual systems that aim to fulfill both objectives.

The systems in this review can be divided into two main groups according to the contextual information they provide. In one group are those systems that provide contextual information by displaying the hyperlink structures in a Web site or in a collection of Web documents collected otherwise. In the second group are those systems that follow our approach and provide contextual information by displaying clusters of Web documents that are formed according to the document contents (or possibly some meta-information). As explained in the previous chapter, we treat post-retrieval document exploration as a browsing task. Consequently, the review herein does not differentiate between the systems designed for the browsing of an organization of Web pages collectedas a result of a search from those designed for the browsing of Web pages residing in aWeb site. The visual techniques used in both types of systems for supporting the visual tasks of overview and zoom are basically indifferent. Consequently, our prototype system borrowed ideas from both types of systems, and our review is inclusive of both types of IR systems.

Another point that needs clarification is that the use of visualization has certainly not been limited to the Web search problems that we are addressing in this study. For example, the TileBars (Hearst 1995) system was designed to help users make first judgments about the potential relevance of documents. This system addresses the problem with the opaqueness as to how query terms are relevant to search results. As a remedy, TileBars provides information about relative document length, query term overlap, frequency of the query terms and their distribution in each document. Although the value of such a system in enriching one's understanding of individual documents is indisputable, it does not show relationships between these documents, and does not summarize the document collection as a whole. In that respect, TileBars is not a system that is designed to reduce information overload. In accordance with the research problem we are addressing, the rest of this section focuses on systems that are built to alleviate the overload aspect of the Web-based IR problems.

Our review of visual systems is based on the different approaches that the systems bring to the first visual task in Shneiderman's task by data type taxonomy: "overview". Then in the next section, we focus on the second and third tasks, i.e. "zoom" and "filter" in this taxonomy paying special attention to systems that adopt distortion-based zooming and filtering techniques since these techniques are of interest to us as well. The review in this section starts by the systems that adopt real life metaphors for providing visual overviews of Web-based information.

The world we live in is a visual world. People have no difficulty in understanding the objects that they are used to seeing all the time and can easily interpret the relationships between these objects. Consequently, a natural first thought in visual system design is the representation of abstract objects by familiar physical entities. In their 1993 paper, Dieberger and Tromp note that a strong spatial interface metaphor that supports orientation within and between hyperlinked documents is needed. One such metaphor is the city in which a house with open doors shows a document in strong relation to the topic looked for, and a half-closed door represents a weak relation. Similarly, the exterior of a house conveys information about internal complexity, age, and functionality of the house and hence the document that the house represents. A worn doormat shows a house that is entered very frequently.

In the information city, hypertexts are represented by houses where walking insidea house means navigating the hypertext and traveling the city is navigating between hypertexts. Similar information objects are grouped together to form "districts of interest", and navigation between different groups is supported by the subway metaphor.

The information city is one of the first attempts to use the idea of visualization in the Web domain. The study presents a list of interesting ideas, rather than a complete implementation. The lack of usability studies and empirical evidence on the usefulness of these ideas is a drawback. Nevertheless, the information city idea is inspirational for Web-based visualization applications.

WebTOC (Nation et al. 1997) is a method to summarize the contents of a Web Site by another familiar structure, a hierarchical table of contents. The automatically generated expand/contract table of contents provides graphical information indicating the number of branches as well as individual and cumulative sizes of these branches. WebTOC uses two different strategies in the automatic generation of a hierarchical table of contents: following existing links, or using the underlying directory structure. Figure 2.6 depicts an example WebTOC session where the left part of the figure is the table of contents of the original homepage shown on the right. When the same hierarchy is summarized, the visualization in Figure 2.7 results. The WebTOC system has other similar presentation modes such as using the length of bars to represent the number of documents in a certain branch of a hierarchy.

Figure 2.6

Details and Context Integrated in “WebTOC”

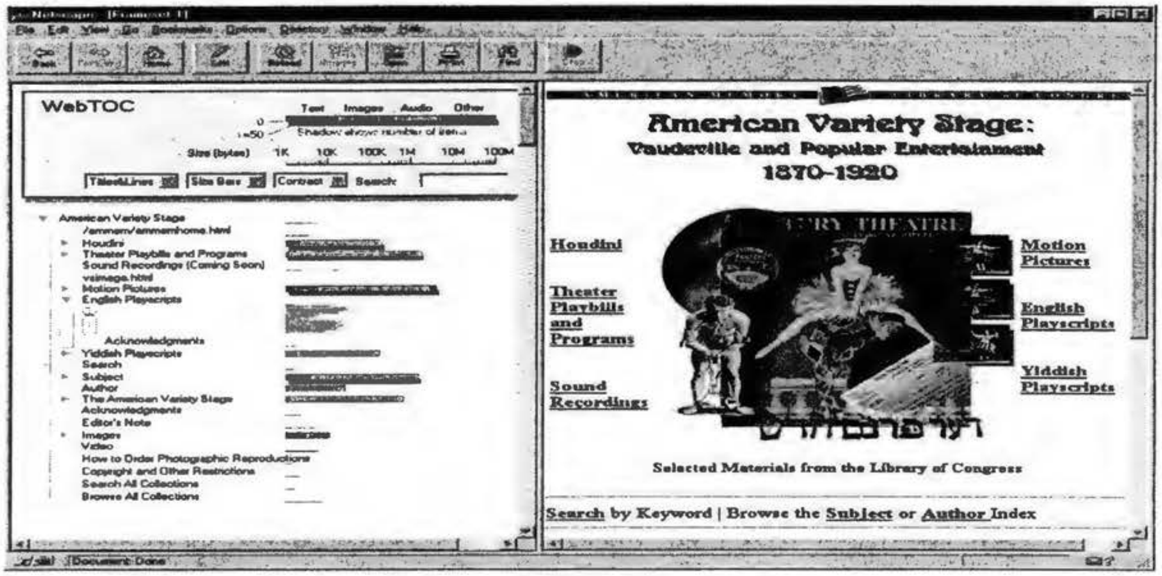
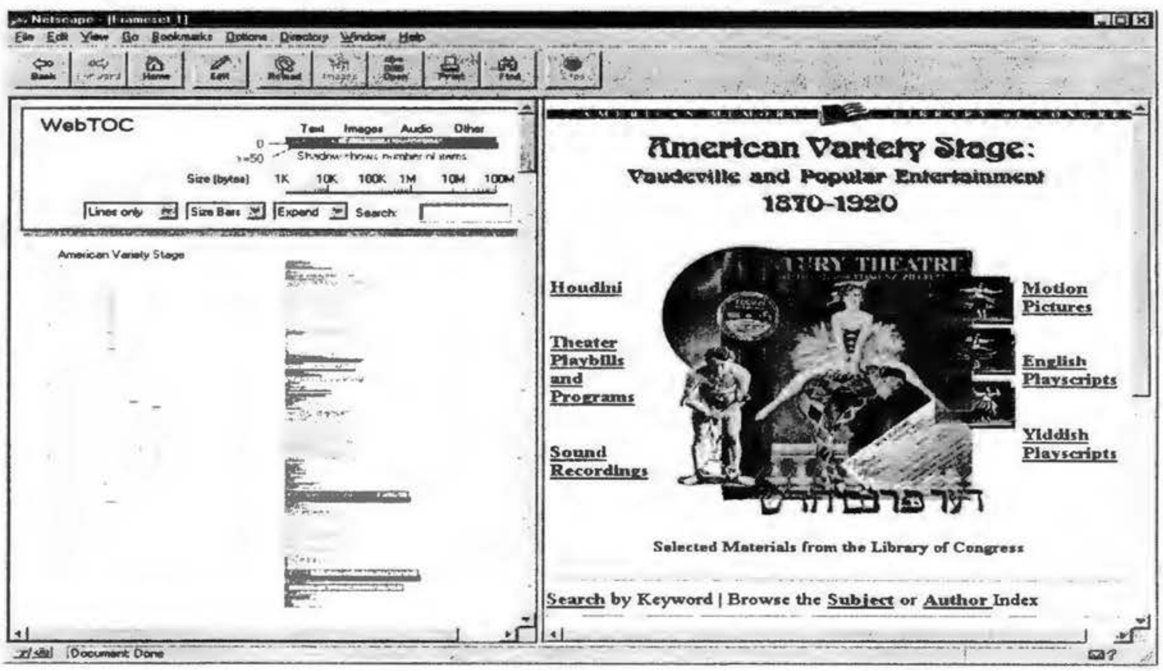


Figure 2.7

Details and Context Integrated in “WebTOC”



WebTOC is a good representative of a group of systems built to present context and details simultaneously (Figures 2.6 and 2.7). A problem with this approach in general is the inefficient use of screen space and the lack of visual cues to smoothly connect the two graphs to each other. Due to these problems, we adopt an alternative method of presenting details in context. The details of this alternative approach are discussed in the next section, and the later chapters of the dissertation.

A similar real-life metaphoric approach to in providing a visual overview is presented in Golovchinsky and Chignell (1995). They argue that the newspaper metaphor provides strong clues to the reader about the relatedness and the relative importance of articles. For example, important articles tend to appear closer to the front of the newspaper, and related topics tend to be found on the same or adjacent pages. The front page gives an overview and has links to various topics. Consequently, the authors argue that newspaper-like interfaces should be appropriate for hypertext interfaces, even if the hypertext does not contain news-related information.

Inspired by similar traditional forms of information organization, Card et al. (1993) propose two moves from the traditional "one page at a time" display of Web pages:

- A move from the single Web page as the unit of interaction to a higher aggregate entity (the WebBook)
- A move from a work environment containing a single element to a workspace in which the page is contained with other entities, including WebBooks (the Web Forager).

Examples of a Web Book and a Web Forager can be seen in Figures 2.8 and 2.9 respectively. The Card et al. (1993) study is an interesting application of the idea of organizing Web documents into groups and presenting a visual overview of these page groups. This application differs from our application in that the groups in these systems are formed manually by a person whereas we are using an automatic clustering method after the documents are collected (by means of a Web search).

Figure 2.8

An example WebBook

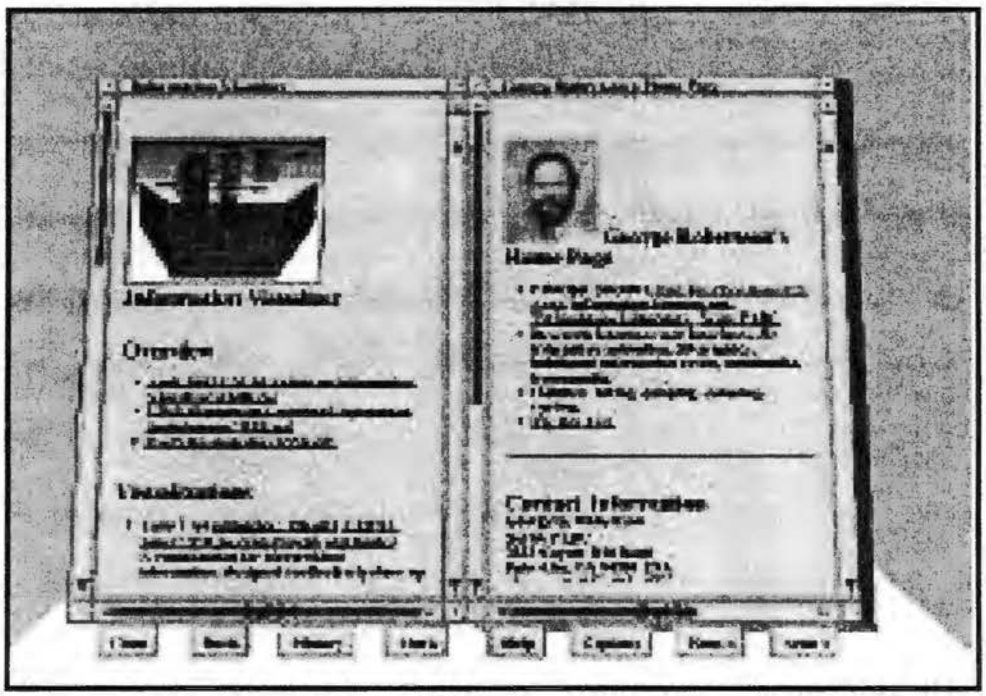
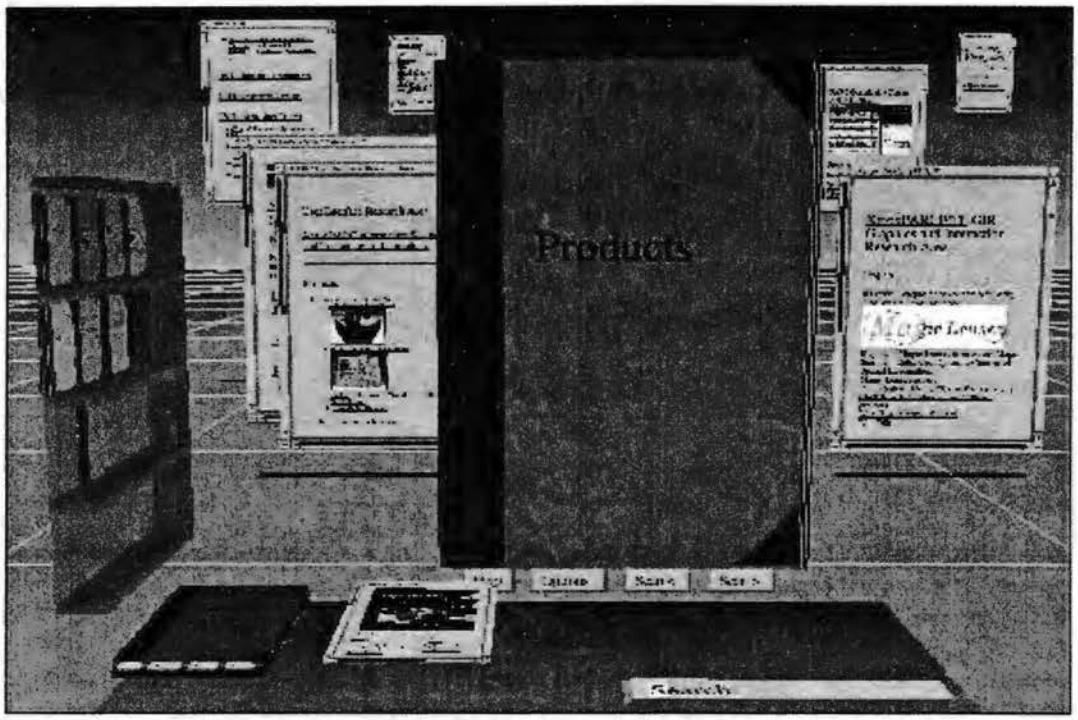


Figure 2.9

The Web Forager



Real life metaphors are intuitive and hence easy to understand visual paradigms. On the other hand, their implementations are relatively complicated and computing intensive. There are numerous alternative structures to the real-life metaphors to present visualizations of Web-based information spaces. Most of these structures have been in use for more traditional information spaces for a fairly long time.

A graph is one such familiar structure that can be useful in representing similarities within a group of objects. A recent study by Liu et al. (2000) reports on a system that is designed to cluster the results to a Web query according to the contents of the documents, and to visually display these clusters and their similarities by means of a graph where the vertices represent the clusters and the edges represent the relationships between them. This system, being developed for the Florida Center for Library Automation, provides insightful representations, but still lacks visual cues to smoothly connect the context and details to each other.

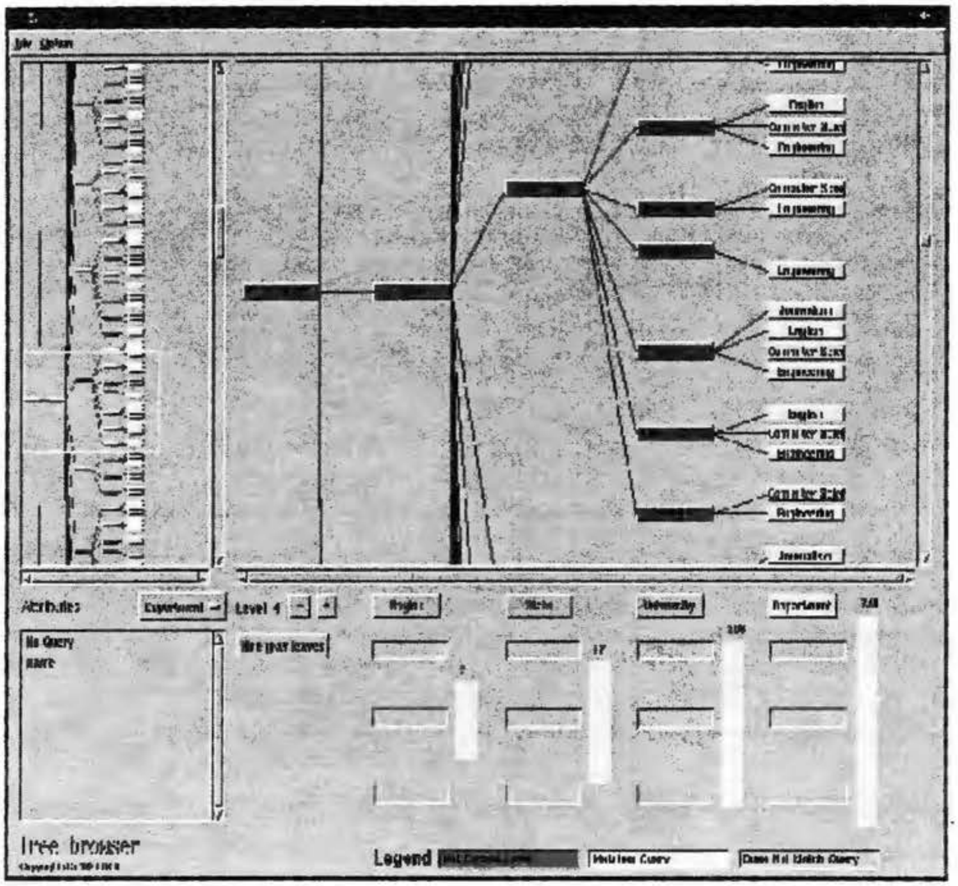
A well-known way to represent hierarchies (including hypertext) is by means of another familiar structure: a tree. In a tree display, the overall hierarchy is overviewed by means of a tree, and areas of more interest (i.e. branches) are zoomed as needed. Individual information objects are located at the leaves of the tree. The PDQ Tree-browser is a representative system based on this principle, and was designed to help information users to browse hierarchies in searching for the nodes of most interest to them. PDQ stands for "Pruning with Dynamic Queries" where the purpose of pruning is to reduce the set of alternatives for a decision. In the 1995 paper, Kumar and colleagues describe the basic PDQ requirements as follows:

- Browse the entire tree and view at different levels
- Query nodes at all levels on the basis of attribute values.
- Hide uninteresting nodes and branches rapidly, and thus reduce the data set progressively.

The PDQ Tree-browser is displayed in Figure 2.10. As seen in Figure 2.10, the PDQ tree is designed to provide summarized context and detailed information on two separate graphs on the same screen. Similar to the problem in the Liu et al. (2000) system, the problem with the PDQ tree is that the visual cues to smoothly connect the two graphs to each other are still not very strong.

Figure 2.10

The PDQ Tree-browser



3-dimensional (3-D) trees are not as common as their 2-dimensional (2-D) counterparts, yet they have the advantage of being able to display a larger portion of the information space than that a 2-D tree can. Like 2-D trees, 3-D trees are used to visualize hierarchies where certain branches of the hierarchical tree can be brought into focus by rotating the whole tree or by stretching a branch of interest.

By far, the best-known example of 3-D trees comes from Xerox PARC and is known as Cone Trees (Robertson et al. 1991). Figure 2.11 shows a cone tree where Figure 2.12 depicts the same tree through rotational and elastic (stretched-out) zooms.

The advantage with such a system is the ability to display more information on one screen yet an important problem is the occlusion of a certain portion of the tree by the focused branch. Nevertheless, it is a promising structure to visualize hierarchies, and has the potential to be more effective when the transitions between the views are supported by animation.

Figure 2.11

A Cone Tree

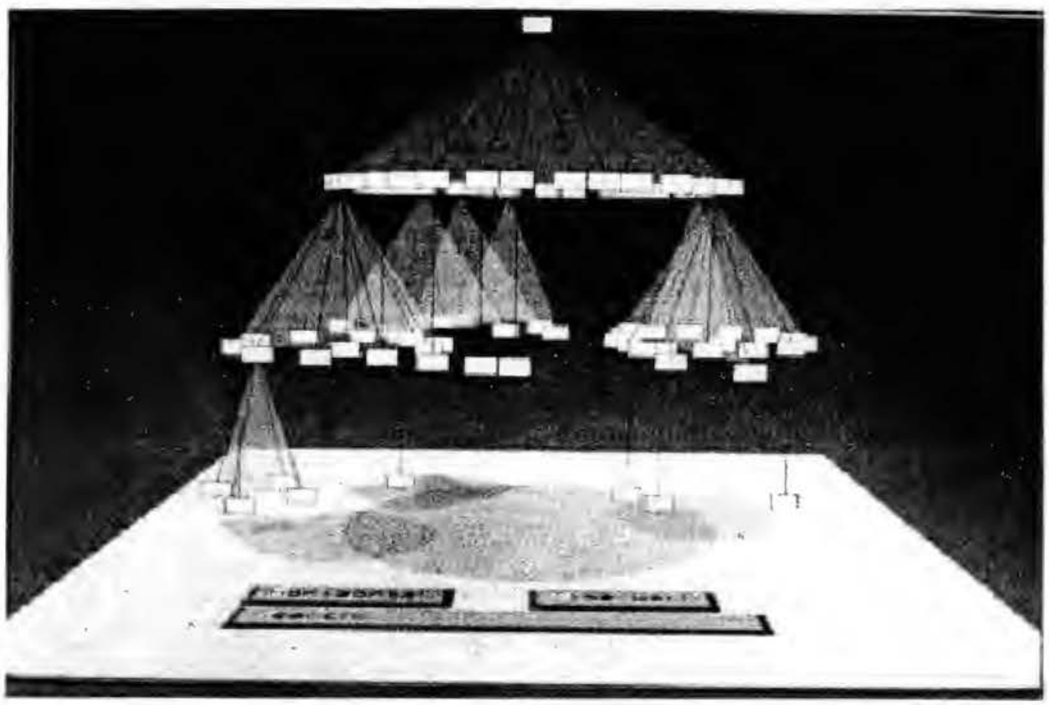
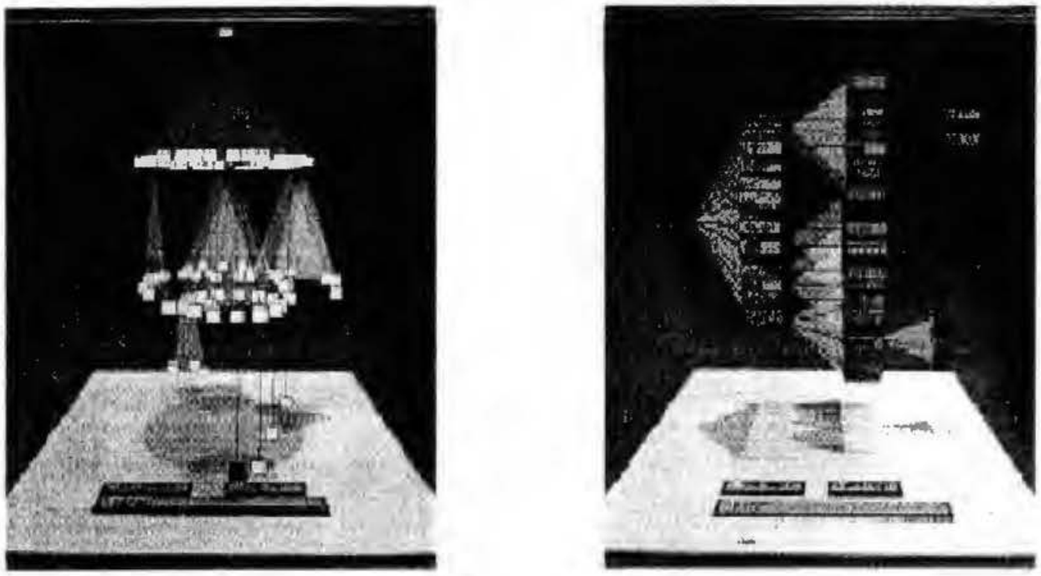


Figure 2.12

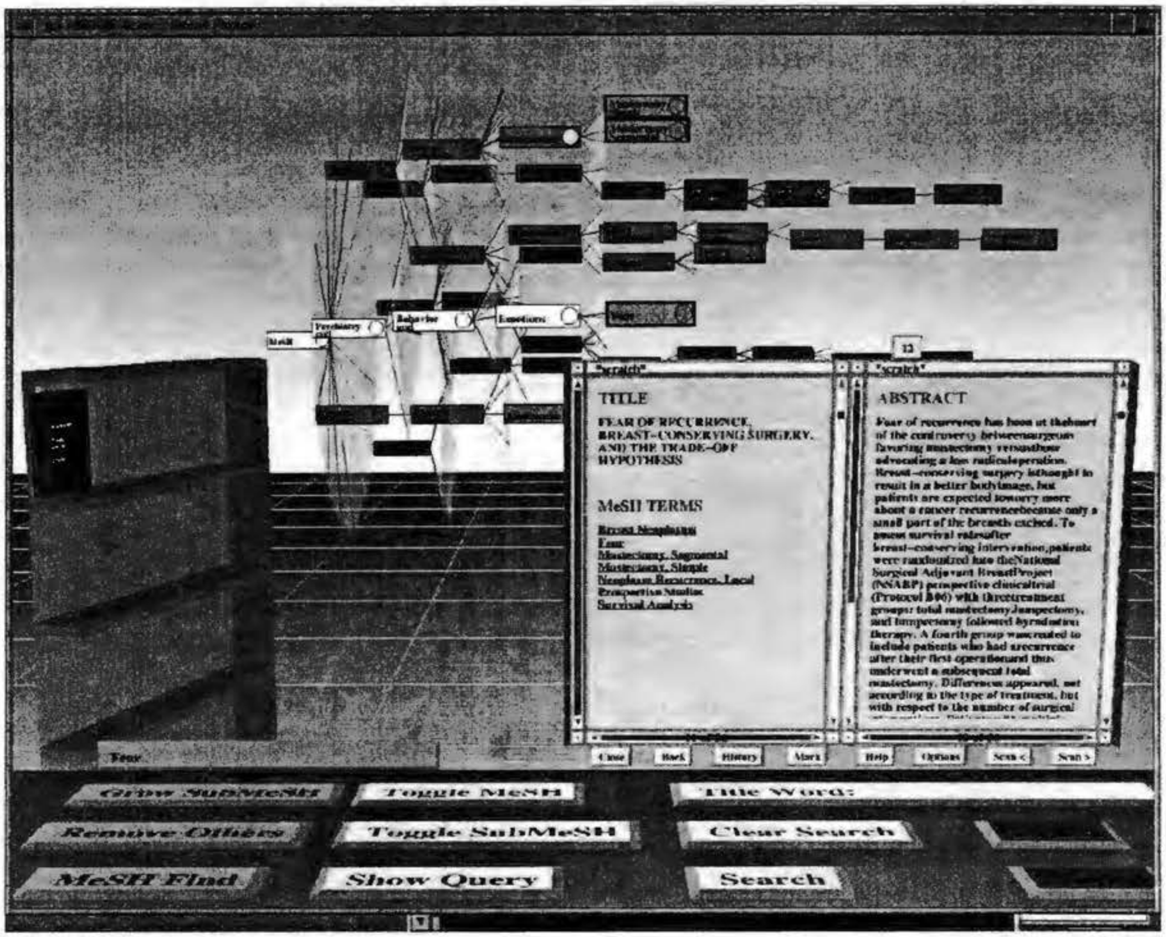
Zooming of a Cone Tree



Cat-a-Cone (Hearst and Karadi 1997) is a more recent application that uses the Cone Tree structure that integrates search and browsing of very large category hierarchies with their associated text collections by using existing 3D+ animation interface components. (Cat-a-Cone integrates *category* hierarchies into *Cone Trees*). The prototype system is designed to separate the graphical representation of the category hierarchy from the graphical representation of the documents allowing a fluid, flexible interaction between browsing and search, and between categories and documents. However, this separation also causes the same problems (i.e. difficulty to conceptually connect the two views) that the WebTOC and PDQ systems have.

Figure 2.13

The Cat-a-Cone Interface



Geographic maps have a long history of providing spatial clues to people by means of an overview of the geographical area of interest. A map can similarly be used to present an overview of an information space. On such a map, different regions represent different groups (clusters). The proximity of regions means the underlying concepts have close semantic contents while the size of a region is an indicator of the size of the corresponding cluster. This idea has been extensively used in information retrieval.

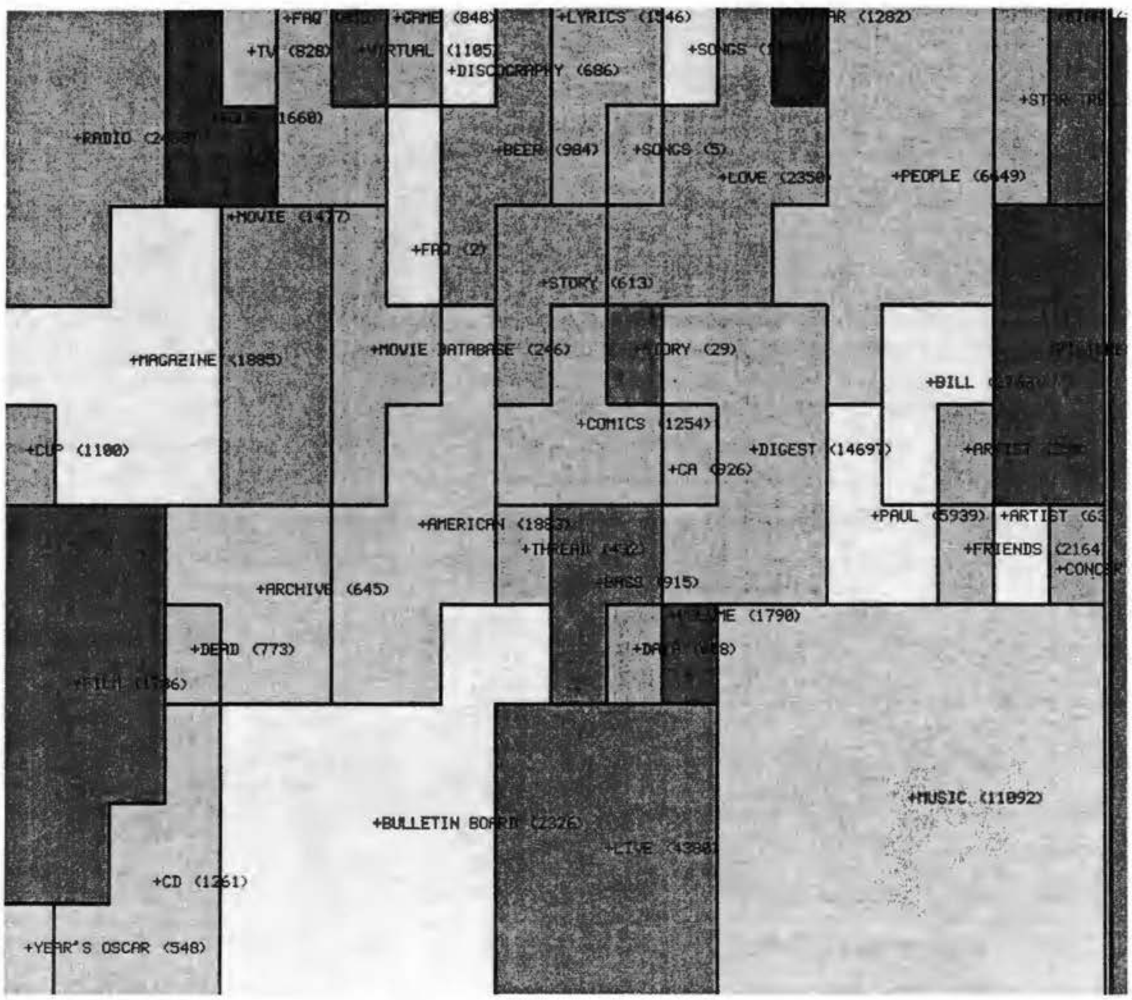
Some common examples of 2-D maps used for the Web are WebSOM from the Helsinki University of Technology (Lagus et al. 1996*),* the CategoryMap from University of Arizona (Chen et al. 1998, Roussinov 1999), the Visual Site map from the University of Kentucky (Lin 1997), the Galaxies visualization in SPIRE (Wise et al. 1995), and Cartia's Themescape. These systems are developed using Kohonen's self-organizing maps. Figure 2.14 displays one such system, the ET (Category) Map. On this map, different regions correspond to different entertainment topics. In a recent application, Roussinov (1999) used a variant of this approach to create an overview of results to a typical Web search. The basic idea in this application was to take the clustering approach one step further, and make the clusters visible. The particular application was based on the partitioning (non-hierarchical clustering) of the search results. Roussinov empirically showed that his approach increased search speed and it was preferred by most of its users.

2-D maps have the characteristic of being relatively simple and easy-to-implement from the application developer's perspective, and intuitive and easy-to-understand from the user's perspective. For these reasons, they are fairly popular representations and are attractive for many practical applications including the one that we have developed.

Similar to 2-D maps, 3-D maps (landscapes) may be the visualizations of a number of clusters or they can be used to visualize a hierarchy such as a directory structure or a Web site. The VxInsight™ from Sandia National Laboratories is a system that displays relationships within large databases by means of clustering the database entities and mapping similarities to 3D terrain maps. It is designed as a mining tool for very large databases to recognize implicit structures such as a collection of papers in an academic discipline.

Figure 2.14

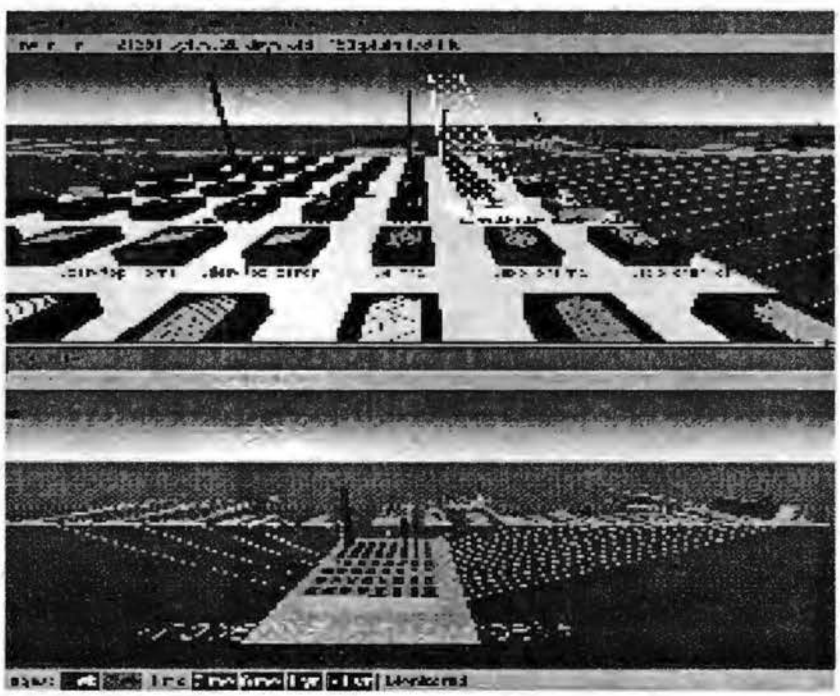
The ET Map



The 3-D FSN-file System Navigator© from Silicon Graphics is another example of visualizing a hierarchy. It is a system created by 3-D bar charts that are connected by a topology on an extended landscape plane (Figure 2.15). The purpose of the system is the managing of large collections of hierarchically structured data such as computer file systems. In Figure 2.15, there are two panels that show the same file system from different perspectives. The virtual landscape is built by cells (directories) containing data blocks (files). The volume of data blocks represents the size of the files and the volume of the pedestals represents the cumulative size of the files within a directory. The spotlight in the upper panel marks a selected file and moves the object of interest in the foreground.

Figure 2.15

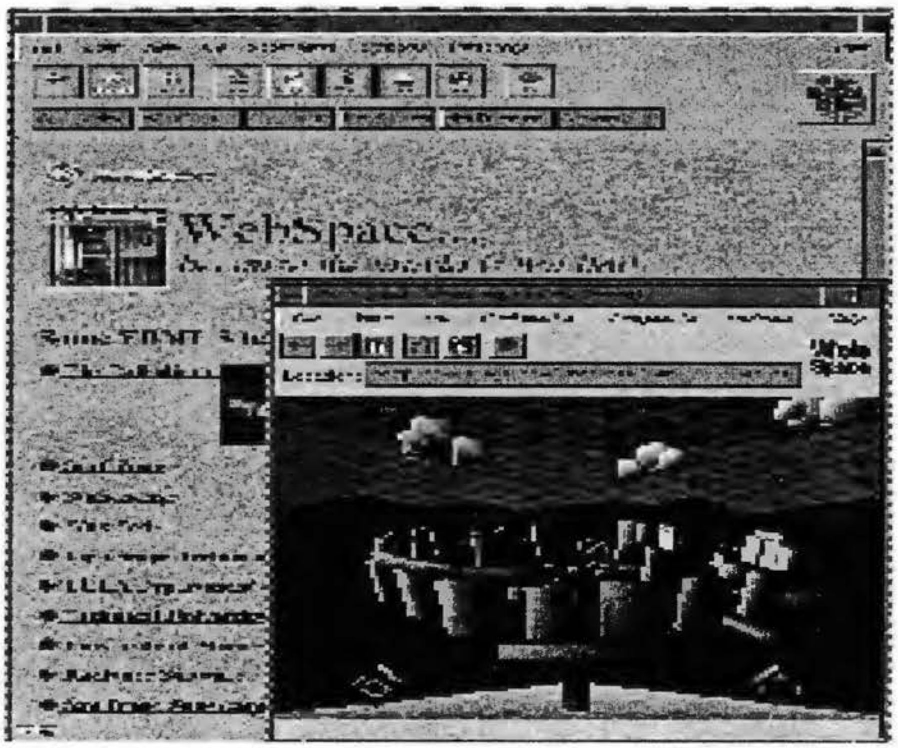
3-D FSN-File System Navigator



WebScape© (Netscape) is an interactive 3-D user interface for popular Web browsers such as Netscape© (Figure 2.16). The 3-D viewer is similar to the FSN, and it supports the Virtual Reality Modeling Language (VRML), an open architecture, platform-independent file format for 3-D graphics on the Internet.

Figure 2.16

Webspace© Viewer



Although attractive, 3-D maps are not as practical as their 2-D counterparts since they are more computing-intensive and harder to implement. There are other 3-D visualization systems that are based on sophisticated clustering algorithms and/or more complex visual paradigms such as the 3-D Nirve (Sebrecths et al. 1999), AspInquiry, 32D Hypercube (Miller et al. 1997), Cosmic Tumbleweed (Miller et al. 1997), and Rainbow (Hetzler and Miller 1998) details of which are beyond the scope of this review. An important point about 3-D systems in general though is the lack of empirical evidence as to their usefulness and superiority over their relatively simple 2-D counterparts (Swan and Allan 1998, Cugini and Sebrecths 1999).

2.3 Distortion-Oriented Systems

A common feature in the visual systems that were reviewed above is that they give undistorted views of an information space. There are two main groups of such systems according to the zooming capabilities they provide. One group of such systems display the zoomed-in area of the visualization in full detail and prune the area that is not in the zoom (strict filtering) while the others provide separate views of the context and details. In this section, we discuss an alternative (distortion-oriented) zooming method that wehave adopted for this study. The main promise of this method is the smooth integration of context and details. A majority of distortion-oriented systems are based on Furnas' (1986) fisheye view approach that we mentioned in Chapter 1.

In the original paper by Furnas (1986), the basic motivation for fisheye views was described and the "degree of interest (DOI) functions" concept was introduced to formalize generalized fisheye views. According to his formulation:

$$DOIfisheye(x,y) = API(x) - D(x,y) \quad \text{(Eqn 2.1)}$$

where DOIfisheye is the user's degree of interest in a given point, x, given that the given point of focus is y,

API(x) is the given a priori importance of x, and

D(x,y) is the distance between x and the current point y.

Using this formulation, fisheye views could be defined in a number of different structures. Furnas demonstrated an application for tree structures and a specific example for tree structured text files.

As explained in Leung and Apparley (1994), there a number of ways a fisheye view can be created. Sarkar and Brown (1992) used a formulation similar to the original one introduced by Furnas and applied the fisheye view technique for viewing and browsing computer graphs. They introduced layout considerations into fisheye formalism, and built a framework to incorporate arbitrary structures by redefining the "distance" concept.

Lamping and Rao (1996) describe an implementation for presenting a 2-D graph through a fisheye zoom. The hyperbolic browser provides a smoothly varying "focus + context" view where the display space allocated to a node decreases continuously with the distance from the focus, yet does not disappear abruptly. Display of a specific node in the graph within the context of the other nodes is shown in Figure 2.17. Figure 2.18 shows the effect of carrying a node on this graph to the focus. The Site Lens™ system from Inxight Software, Inc. is based on this principle and it produces maps of Web sites which display "details + context" of a Web site similar to that illustrated in figures 2.17

and 2.18. The fractal projection of an information space as described in Miller et al. (1997) is another system working on similar principles.

The Perspective Wall (XEROX PARC, XSoft) uses a 3-D technique that integrates detailed and contextual scale-reduced views of an information space (Figure 2.19). The wall moves a selected item into the center panel with a smooth animation. The ratio of the context and detail can be adjusted where a selected item is explored in detail within the context.

Figure 2.17

An organization chart

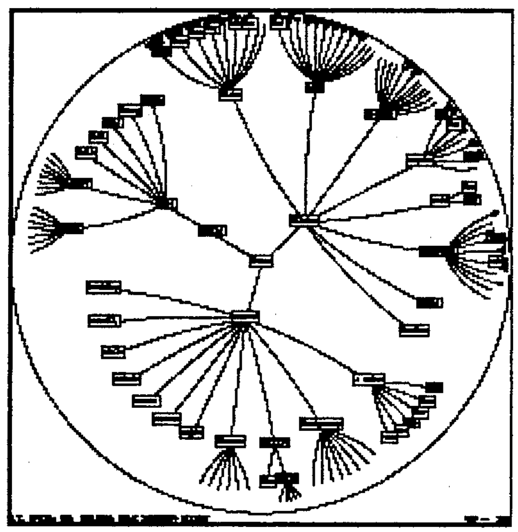


Figure 2.18

Changing focus on the organization chart in Figure 2.17

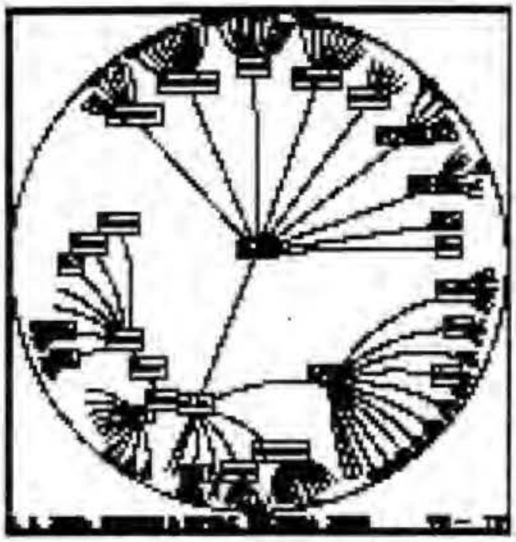
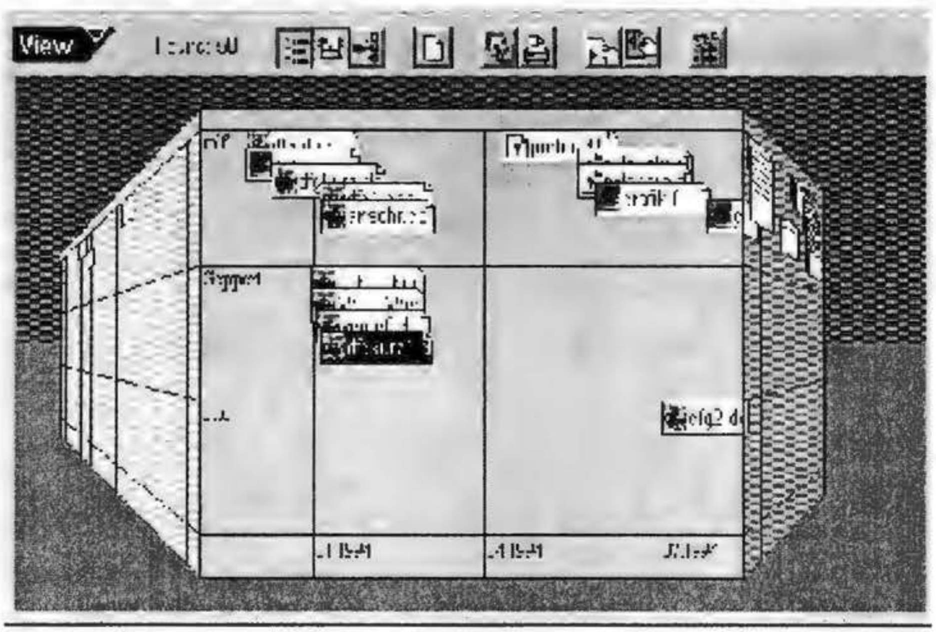


Figure 2.19

The perspective Wall



Collaud et al. (1995) introduced their CZWeb tool that makes use of two techniques, fisheye views and continuous zoom to help users navigate the Web. The prototype displays a network in a rectangular 2-D display space. A hierarchy of nested triangles is created by recursively subdividing this area into smaller rectangular areas. This hierarchy is used to display some Web pages in great detail and the others in less detail or not at all.

Bederson et al. (1998) developed a Web-browsing prototype in Pad++, a multiscale graphical environment. This prototype displays multiple Web pages and the links between them instead of showing one page at a time. A fisheye view approach is used in this display method where the page in focus is clearly readable whereas the others are shown in smaller scales to provide context. This approach was compared to the traditional display method of Netscape in several different scenarios, and the authors found that after some changes to the prototype, subjects using Pad++ answered questions 23% faster than those using Netscape did.

There have been other approaches to the visualization of the hierarchical data structure on the Web by embedding details and context. Two such systems that use a graph data structure as their visual representations are the Navigational View Builder (Mukherja et al. 1995, Mukherja and Foley 1995), and the Auditorium Visualization system by Terveen and Hill (1999). Terveen and Hill introduced a novel data structure called "clan graphs" that work on the Web *site* instead of the Web *page* as their basic data unit.

As evident from the brief discussion in this section, distortion-oriented, or more specifically fisheye view systems have found applicability in information visualization especially in the visualizations of complex spaces. The idea to smoothly integrate the context and details is promising, but needs to be modified for specific implementations. Our implementation of the technique is one such modification while being committed to the original design principle shown by means of the examples covered here.

2.4 Summary and Research Directions

The previous literature on Web search agrees that the search methods used presently are not successful enough in the sense that commercial search engines usually overload their users with the large number of results they return. The linear display of search results is not very efficient unless the information that the user is seeking is among the first few pages of the list. There is very little structure in the list display of Web search results. In this respect, summarization of search results by grouping content-wise similar pages together is helpful since it imposes a structure on the information space. Northern Light (www.northernlight.com) -one of the most popular search engines today- has already adopted the clustering approach in presenting search results as an addition to the traditional list display. Northern Light's patented custom search folders organize the search results into non-overlapping groups. For example a search for "fractal" on Northern Light returns 182,592 results, and this list is obviously very difficult to explore in a linear fashion. However, the search engine also presents these results organized into fourteen groups among which are "fractals (mathematics)", and "programming algorithms". This approach is intuitively promising as a first attempt in reducing overload.

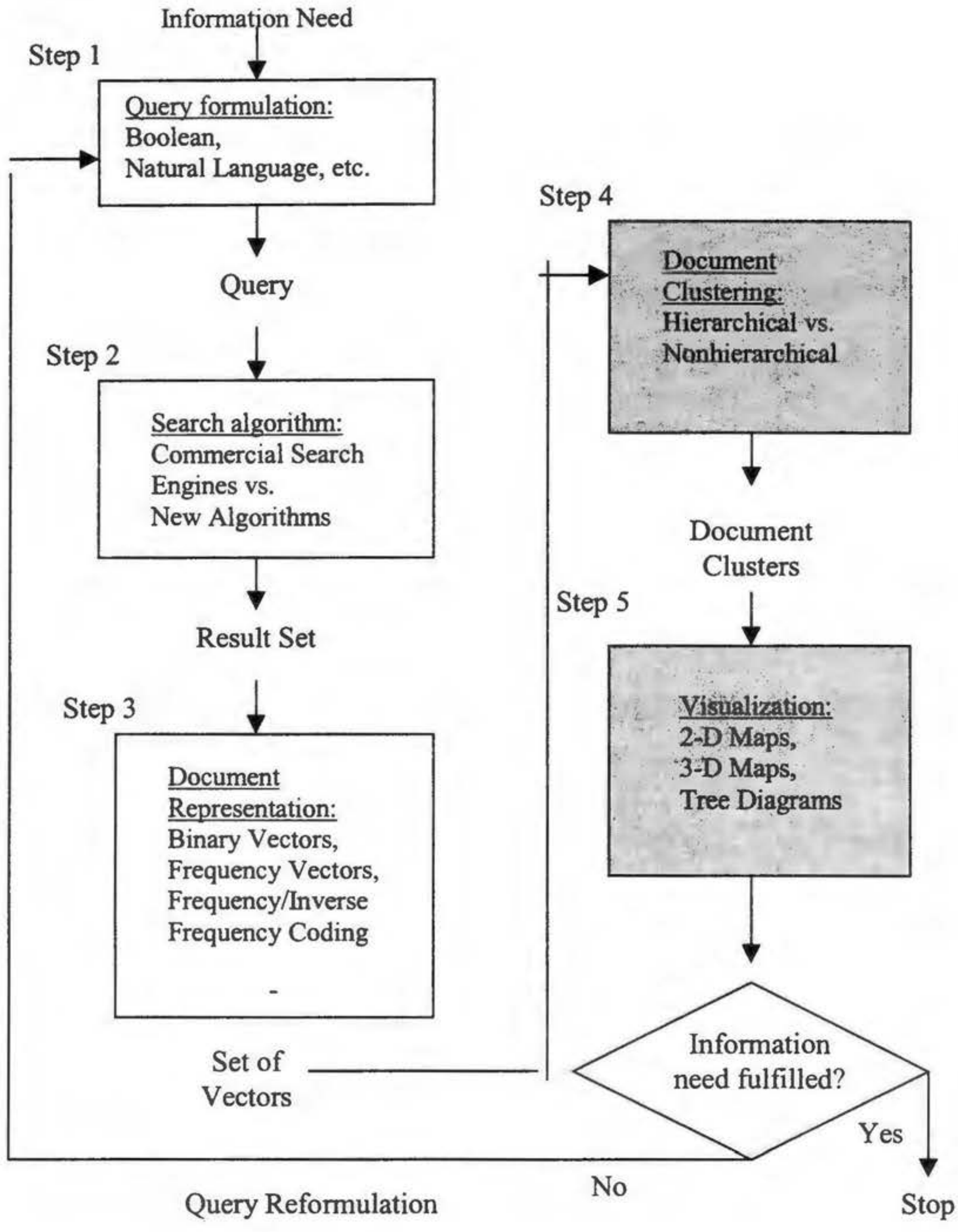
Past research has also focused on the use of visualization as an effective way of presenting summary information since visual cues speed up people's understanding of information. The different visual paradigms introduced for browsing complex information spaces are promising. Today's cheaper and powerful computers together with the recent positive experience make visualization a viable method for reducing information overload. Observing the promise of clustering and visualization in the successful support of information search (by reducing overload), we state our first research question as follows:

Can a (clustering-based) visual presentation system improve search success over one without such a support?

For answering this question, we developed a system that is based on the previous work on visual information search systems as well as a slight modification of the four-phase framework for clarifying information search proposed by Shneiderman et al. The four phases in this framework are "formulation" (identification of information sources, query formulation, etc.), "action", "review of results", and "refinement of the formulation" respectively. Figure 2.20 depicts an extended version of "the model of information search with clustering and visualization support" previously displayed in Figure 1.1. The model suggests a number of research questions in each of its steps, and a large amount of research is already addressing some of these possible questions. The aim

Figure 2.20

Information Search with Clustering and Visualization Support



of our study is to apply previously untried design guidelines in the pursuit of the clustering and visualization tasks and testing their success.

The previous literature suggests that a major problem in browsing a large information space is the disorientation problem. Especially in the visualization of search results where the boundaries between different areas in the visual space are rather imposed than natural, it may be desired to examine local detail without losing awareness of global context. A fisheye view is a potential way to provide such a global context. To our knowledge, the only system that has attempted a fisheye view approach for the visualization of Web search results is VITESSE (Nigay and Vernier 1998). This system uses a visual display of search results without any content-wise or link-based organization, i.e. clustering, hence, the benefit of this system in reducing information overload is limited. This leads to the formulation of our second research question:

Can a (clustering-based) visual presentation system supporting fisheye zoom improve search success over one with full zoom only?

For answering this question, we built a system that supports the fisheye zooming of the (clustered) information space, and empirically compared its success to that of the full zoom system.

The review of the past literature reveals that a good number of the studies on system development are usually not supported by empirical user tests. For this reason, the usability of most of the approaches is not very well known. Accordingly, there is an obvious need to perform such user studies for better understanding of the concepts we are studying. The next chapter displays the details of our plan for the prototype implementation and for the empirical testing of the system design ideas stated within the research questions and implemented within the prototype.

3. METHODS

This chapter discusses the details of our approach towards the treatment of the problem we identified in the first part of the dissertation. Section 3.1 discusses the methods employed in the prototype design. Section 3.2 explains the operationalization of the conceptual variables (constructs) that are used in the formulation of the research questions. The hypotheses of the research are formulated according to this operationalization. Section 3.3 discusses the experimental design.

3.1 Prototype Design

The design of a prototype for (clustering-based) visual information search entails making decisions on the design parameters (i.e. methods) in each step of the model depicted in Figure 2.20. As mentioned before, this study proposes a fisheye zooming alternative to the full zoom visualization of Web search results. We chose to use hierarchical clustering to better facilitate the fisheye zooming by means of aggregating lower level clusters to provide a contextual summary. Our aim is to test the effectiveness of the combined use of hierarchical clustering and visualization (with the two alternative zooming methods). To be able to do that, we have built a proof-of-concept prototype system, which requires the implementation of methods for performing the tasks in all five steps of the model. For this reason, we briefly describe all these steps hence presenting a more complete picture of our prototype.

3.1.1 Query Formulation and Search

For this module of our system, Digital Corporation's AltaVista, a popular commercial search engine, is used. AltaVista can handle Boolean as well as natural language queries and hence gives its users a relatively flexible interface for expressing their information needs. The AltaVista engine has an indexed database of over 130 million pages. It provides the opportunity to search by specific language, and has features such as rudimentary language translation and sophisticated techniques for refining searches and ranking results. Being a compact product, AltaVista includes the indexed database, and the search algorithm with the query interface. As a result, it supports query formulation, and performs the search (the first and second steps of our model).

The completion of these tasks results in a ranked list of (Web) documents potentially relevant for the expressed information needs.

3.1.2 Document Representation

As mentioned in Chapter 2, the major methods for clustering a collection of Web documents are based on either the documents' textual content, or the connectivity (hyperlink) structure, or various other characteristics including file-system attributes, access statistics, usage statistics (Pirolli et al. 1996b), Web sites that they come from, author, and time of publication (Baldonado 1998). In our model, we are using textual contents of the documents to cluster them. Accordingly, the document representation method is based on the documents' (semantic) contents.

We adopt a method commonly used in information science for this purpose. This method requires that each document be represented as a vector where each element in the vector corresponds to a term in the total collection of documents (Salton 1989). An element in a vector represents the weight of the corresponding term in the specific document that the vector represents. The calculation of weights is performed on a list of terms (words or phrases) in each document that remain after the elimination of noise words, those that are used frequently in the English language but do not carry unique meanings in each document. Some such words are "is", "the", "a", "of", "but" and "an".

In some applications (e.g. Roussinov 1999), the term weight is limited to two values, namely a 0 to represent nonoccurrence, and a 1 to represent the occurrence of the term. To understand this idea by means of an overly simplifiedexample, consider the following paragraph:

"Many people believe that information technology is the key source in MIS. Indeed, information technology is a critically important set of tools for working with information and supporting the information and information processing needs of your organization." (Haag et al. 2000, p.5)

Indexing this paragraph, the following terms would be extracted: *"people"*, *"information"*, *"technology"*, *"key"*, *"source"*, *"MIS"*, *"set"*, *"tool"*, *"support"*, *"process"*, *"need"*, and *"organization"*.[3] Now, let's look at the following paragraph:

[3] the list of index terms may change depending on the *noise words list*, *stemming algorithm*, etc. used

"But information technology is not a panacea. You have to realize that the success of information technology as a set of tools in your organization depends on the careful planning for, development, management, and use of information technology with the twoother key business resources -people and information" (Haag et al. 2000, p.5).

The terms *"information", "technology", "panacea", "success", "set", "tool", "organization", "planning", "development", "management", "use", "key", "business", "resource", "people"*, and *"information"* would be extracted as a result of indexing this paragraph.

If these two paragraphs were the only ones of interest[4], then we would only use the terms *"people", "information", "technology", "key", "source", "MIS", "set", "tool", "support", "process", "need", "organization", "panacea", "success", "planning", "development", "management", "use", "business"*, and *"resource"* for representing both pieces of text. Since there are 20 index terms, every document would be coded as a 20 dimensional vector. For example, using the above index terms, the second paragraph would yield a vector representation of *[1 1 1 1 0 0 1 1 0 0 0 1 1 1 1 1 1 1 1 1]* since the terms *"people", "information", "technology", "key", "set", "tool", "organization", "panacea", "success", "planning", "development", "management", "use", "business"*, and *"resource"* exist in the paragraph while the terms *"source", "MIS", "support", "process", and "need"* do not.

Although appealing because of its simplicity, this coding scheme does not reflect the importance of the terms in the documents, because there is no distinction between frequently and infrequently used terms. As an improvement, term frequencies can be used

[4] As mentioned before, this is an oversimplification.

instead of binary values to represent the importance of terms in documents. Using the above example, the secondparagraph would be coded as *[1/20 4/20 3/20 1/20 0 0 1/20 1/20 0 0 0 1/20 1/20 1/20 1/20 1/20 1/20 1/20 1/20 1/20]* since the total number of terms is twenty, the term *"information"*, and *"technology"* appear four and three times respectively while the other terms appear only once. This approach is an improvement over the binary valued vectors, yet is still incapable of representing the unique importance of a term in a document. A third approach to document coding is based on this observation, and calculates the weight as a function of the term frequency and inverse frequency. The inverse frequency is defined as the ratio of the number of documents that include a given term to the total number of documents in the collection. This third approach of vector representation has a conceptual appeal.

The vector representation algorithms described above are adopted from previous work of Salton (1989). Below is a summary of the general algorithm:

- Identify the individual words occurring in the documents of a collection
- Use a stop list of common function words (noise words) (is, the, a, of, but, an, etc.) to delete the frequent but too general words from the text
- Use an automatic suffix stripping routine to reduce the remaining words to word-stem form (e.g. analysis, analyzer, and analyzing are all reduced to analyze)
- For each word stem j in document i calculate a weight w_{ij} in one of the three ways (binary, frequency, both frequency and inverse frequency) as described above. One common formula to calculating this weight as a function of frequency and inverse frequency is:

$$w_{ij} = f_{ij} * \log (N/df_j) \qquad \text{(Eqn 3.1)}$$

where f_{ij} is the frequency of word j in document i,

df_j is the number of documents that includes the word j,

and N is the total number of documents in the collection.

- Represent each document j with a vector composed of w_{ij}'s as calculated before.

Tkach (1998) observed that using single words or concepts for extracting representative terms for the document collection is a very tedious task. An alternative to using singe words for document representation is the use of lexical affinities. "A lexical affinity is a correlated group of words, which appear frequently within a short distance of one another. Lexical affinities include phrases like "online library" or "computer hardware" as well as other less readable word groupings. They are generated dynamically, thus they are specific for each collection. A set of semantically rich terms can be obtained without a need to hand-code a specialized lexicon or a thesaurus" (Tkach 1998). IBM's IntelligentMiner uses this principle in indexing a group of documents, and in this study, we use this tool for indexing and document representation.

As a result of indexing and vector representation, the documents are in a form that is suitable for further mathematical processing. Our system uses the same tool (IntelligentMiner) for indexing, document representation and clustering. Nevertheless, we contend that it is worthwhile to cover the basics of the clustering idea before moving to the next step hence the next section includes the details thereof.

3.1.3 Document Grouping (Clustering)

Document grouping is essential for summarizing the document collection and identifying patterns. Clustering is performed to build uniform groups of documents that are significantly different from each other. This group structure is an important element in visualizing the similarities between the documents within a group and the differences between the documents in different groups. Our visual information search model depends on the visualization of search result groups to provide an overview for information searchers to get necessary details if and when demanded. In our model, visualization is independent of what technique is used for grouping the documents. Rather, it takes the document clusters as input and gives a visual representation of the grouped structure. For this reason, document clustering in our prototype is an activity that is separate from visualization, unlike the case in the previously discussed (also see the next paragraph) Self Organizing Maps (SOM) based systems where the clustering and visual representation are done simultaneously.

There are two main types of clustering algorithms differing according to the final groupings that they create. Nonhierarchical techniques divide (partition) a data set into a series of subsets, where these subsets are comprised of similar objects, and there is no hierarchical relationship between them. An example of such a partitioning is the "custom folders" structure that the Northern Light search engine created in response to a search on "fractals" as mentioned in Chapter 2. The cluster structure resulting from a nonhierarchical technique is dependent on a number of parameters specified by the researcher such as the desired number of clusters, and for that reason tends not to be very stable (Willet 1988). Recently, artificial intelligence techniques, particularly Self-

Organizing-Maps (SOM's) have also found applicability as alternative (nonhierarchical) clustering techniques (e.g. Lin 1997, Chen et al. 1998, Rousinnov 1999). SOM's are unsupervised networks inspired by the organization capabilities of the human brain, and are known to yield relatively stable cluster structures.

A hierarchical cluster scheme is composed of clusters within larger clusters where the largest cluster is the whole collection of objects that are being clustered. This is a tree-like structure where the individual objects reside on the leaves and larger and larger clusters are reached when one goes higher in the tree towards its root. A very well known example of a hierarchically clustered collection of objects would be the directory structure on a computer's hard disk. In such a structure, there are a number of directories (starting from the root) within which files and/or other directories are stored.

For the specific application we have developed, a hierarchical clustering algorithm is preferred to a nonhierarchical one for the following reasons:

- Nonhierarchical algorithms require some parameters to be specified in advance assuming that there is some prior knowledge about the vector space. In an automated system, this kind of prior knowledge is not (always) available.
- When the number of objects to be clustered is high, nonhierarchical clustering algorithms either create too many clusters, or they group too many objects in one cluster. This kind of a structure may not reduce the overload to a desired level.
- In a fisheye zoom system, as one looks at the information objects (pages) in one cluster, the other clusters need to be summarized to provide a context. This requires combining similar clusters at a higher level and hence assumes a hierarchy.

Two main strategies for the construction of hierarchical clusters are the agglomerative and divisive strategies. An agglomerative strategy works bottom-up by considering each object as a cluster of its own and then iteratively joining clusters to form larger ones until there is only one cluster. Conversely, a divisive strategy starts by one cluster containing all the objects and divides this cluster into smaller and smaller pieces based on some measure of dissimilarity or distance. Divisive strategies have some inherent theoretical disadvantages (Willet 1988). As a consequence of these disadvantages (details of which are beyond the scope of this study), agglomerative clustering strategies are by far the most popular.

Agglomerative methods in hierarchical clustering can be described by the following general algorithm:

```
FOR i = 1 to N-1 do
  FOR j = i+1 to N do calculate SIM (i, j)
REPEAT
   search similarity to identify the most similar pair of clusters;
   combine this pair, K and L into one and form a new cluster KL and
   update SIM (i, j) by calculating the similarity between KL and every remaining
   cluster
UNTIL there is only a single cluster
```

The various agglomerative clustering methods differ in the definition of similarity (e.g. the Euclidean distance or the cosine of the angle between the vectors) that is used for the selection of the most similar pairs of clusters and for the updating of SIM in the algorithm above (Willet 1988).

As mentioned in the recent two sections, we use IntelligentMiner's hierarchical clustering module, which is based on the principles that are discussed in this section. This clustering module integrates document indexing, representation, and clustering functions. The output of this module is a hierarchy of Web documents.

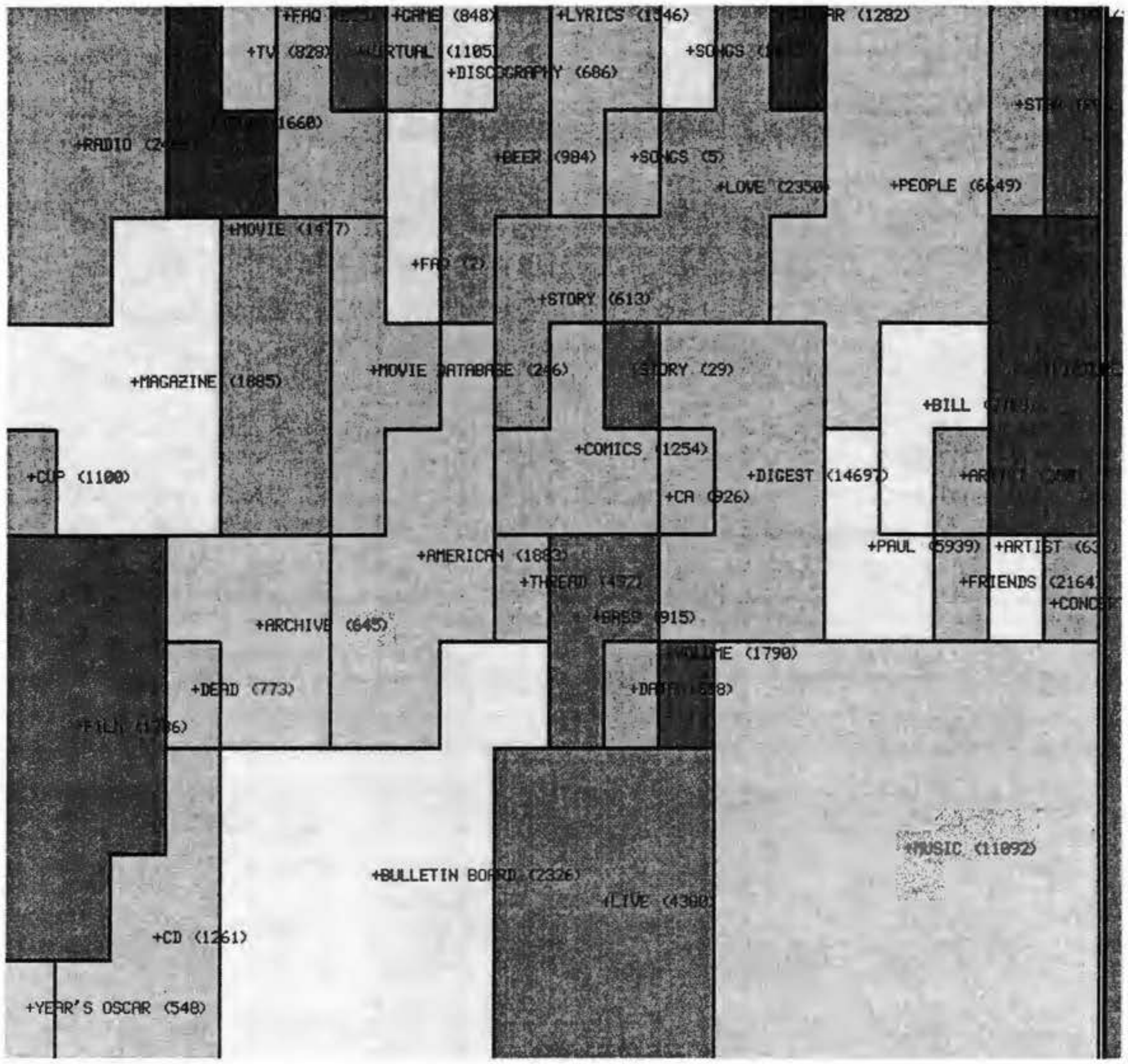
3.1.4 Visualization

A common way of document cluster visualization is displaying an overview of the clusters at a certain level of the hierarchy and facilitating the zooming on one of these clusters when needed. Figure 3.1 gives a 2-D map overview of the entertainment directory from Yahoo (Chen et al. 1997). In Figure 3.3 we can see another overview this time that of the Library of Congress. This visualization is created using the SiteLens software from Inxight. The regions of more interest on both of these overviews can be explored in further detail by zooming-in. Depending on the zooming method, the area that is out of the zoom can either disappear from view (full zoom) or can be kept in view to provide a context (fisheye zoom). The prototype systems that have been developed so far for visualization of Web search results have used the full zoom method (Lin 1991, Lagus et al. 1996, Roussinov 1999). Our study compares a full zoom method to our fisheye view approach.

As an example of the full zoom method, one can examine the ET-map displayed in figures 3.1 and 3.2. Figure 3.2 is the resulting map when one focuses on the music related documents (bottom right corner) on the map of Figure 3.1. Figure 3.2 displays the information objects under the cluster labeled as "music" and every other cluster in the overview of Figure 3.1 is eliminated from the view.

Figure 3.1

The ET Category Map (Chen et al. 1997)



As an alternative to this approach, we can examine Figure 3.3 and Figure 3.4. When one focuses on the Copyright Office section of Figure 3.3 to examine that specific part of the figure in greater detail, the visualization in Figure 3.4 results. This is an example fisheye view where the context and details are smoothly integrated. Our study uses both of these visual paradigms, and different methods of zooming, and applies them to the Web search results space. The remainder of this section explains the details of our approach to creating the visual overview of document clusters and the two different methods of zooming.

Figure 3.2

The Music Category

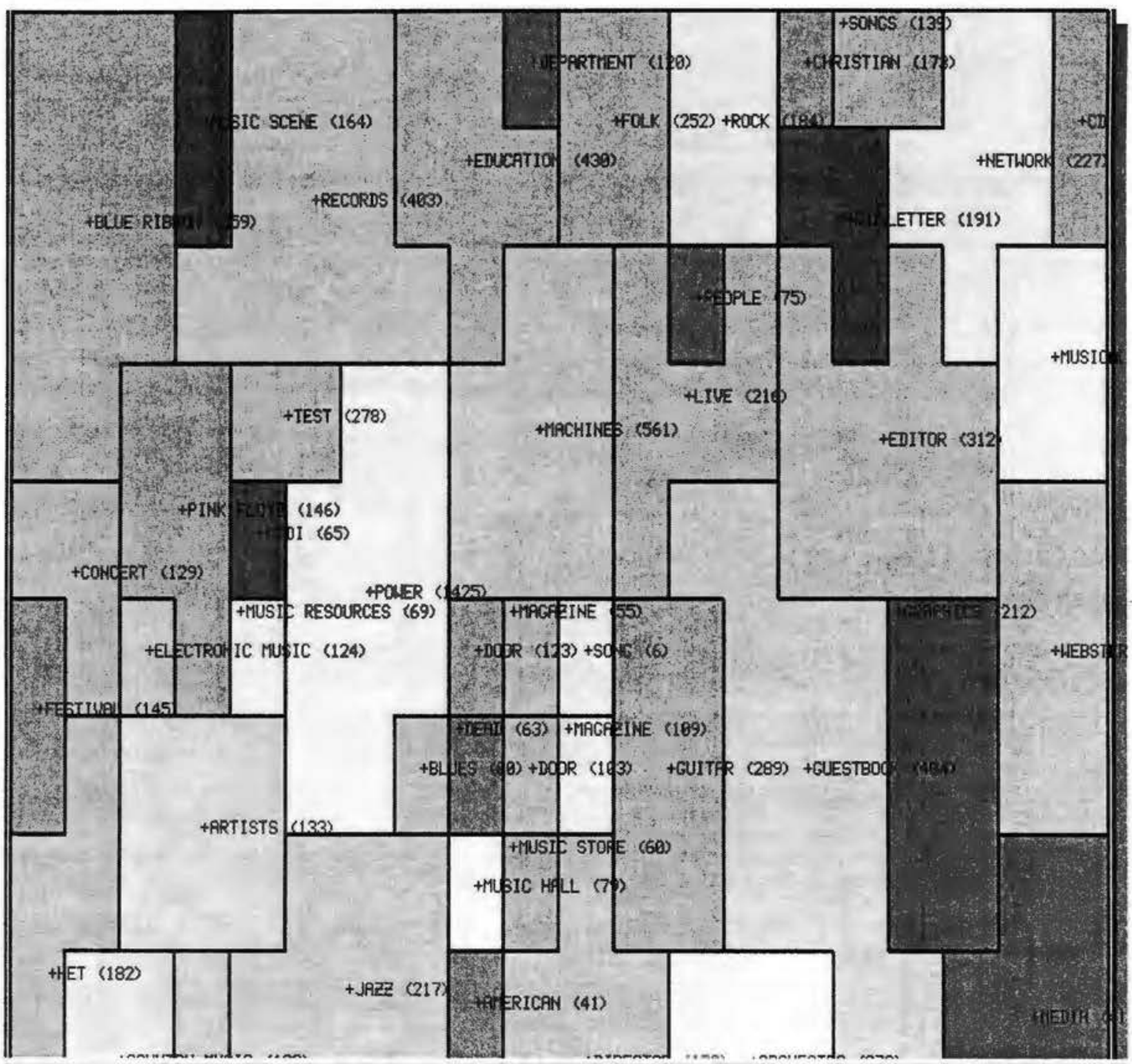


Figure 3.3

"Hyperbolic Tree" of Library of Congress Library

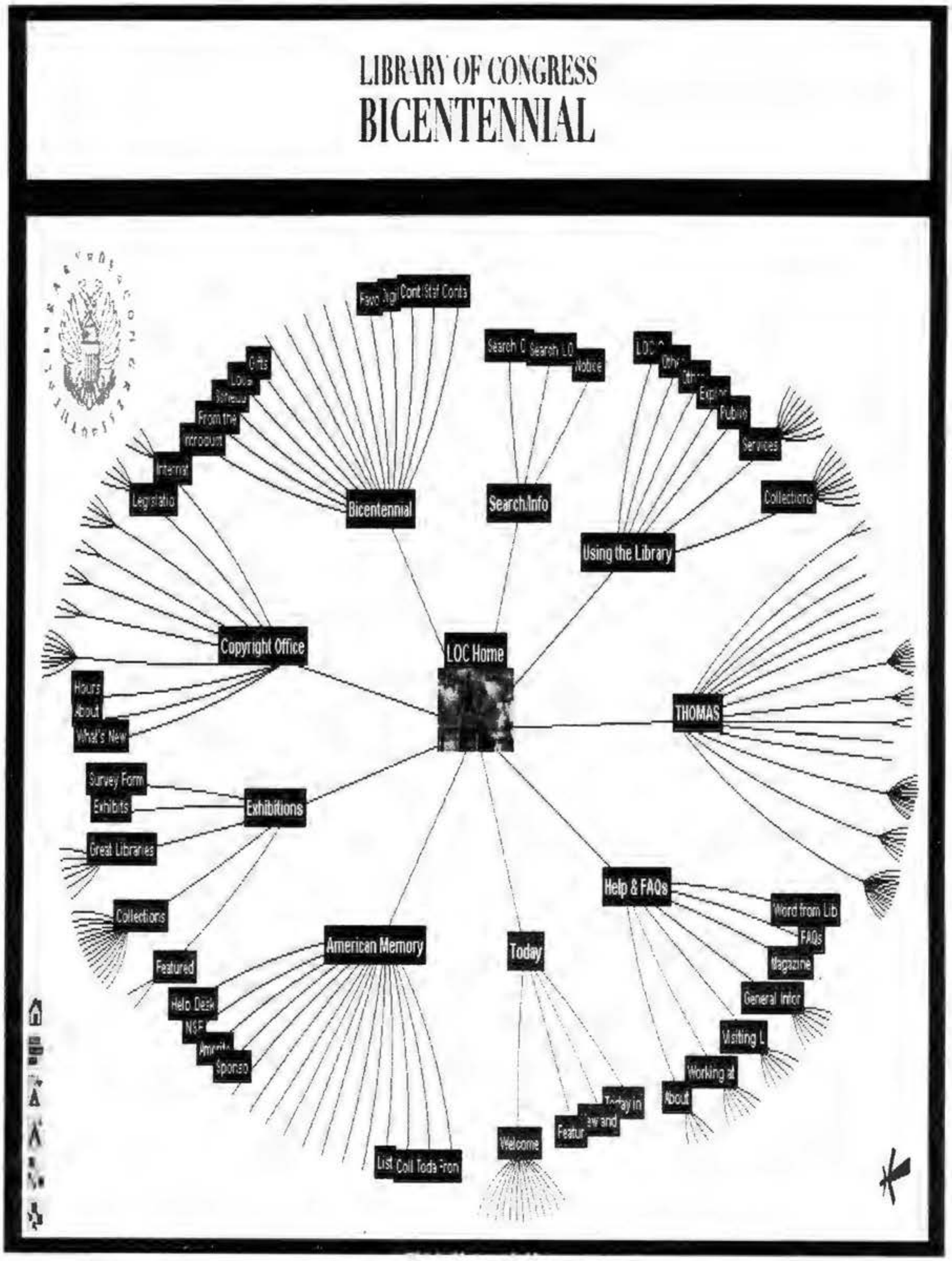
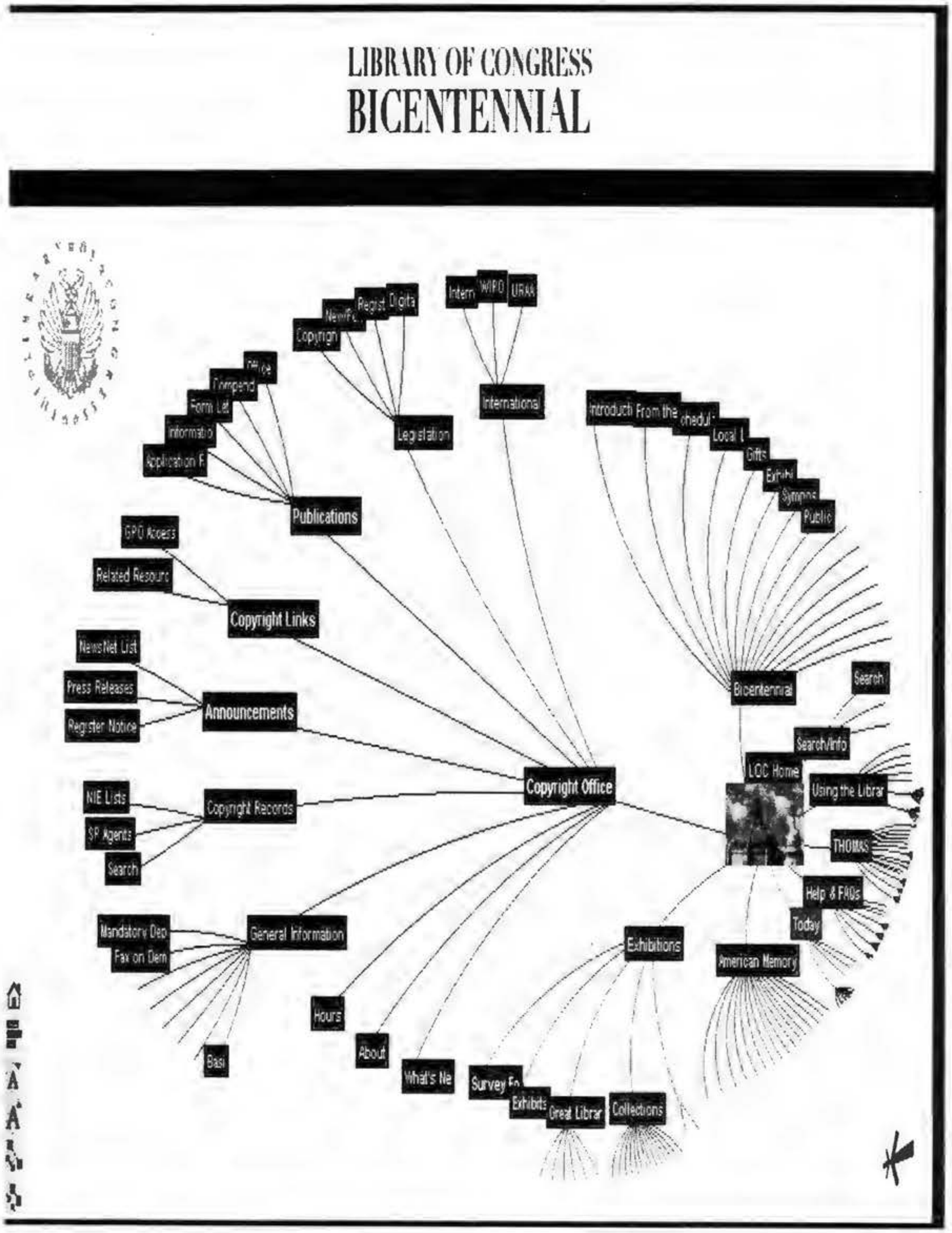


Figure 3.4

Zoomed View of the Copy Right Office



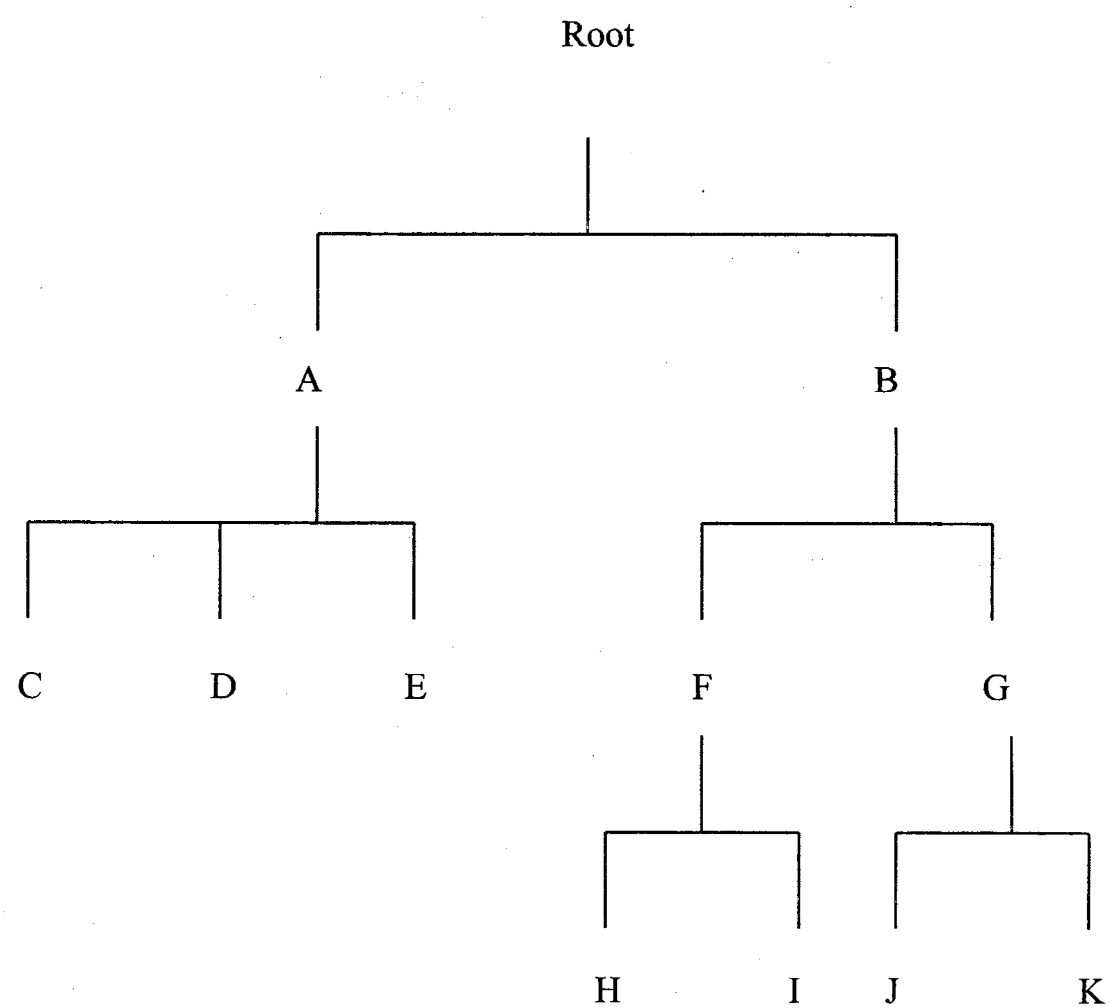
In this study, we adapted the treemap algorithm of Johnson and Shneiderman (1991) for creating an overview of the document clusters that are obtained as described in the previous section. The treemap algorithm is a space preserving technique for visualizing large hierarchies. It is based on the observation that a tree representation of a hierarchy is space-inefficient, therefore, is not appropriate for large hierarchies despite its intuitive appeal. As discussed in Chapter 2, maps are popular representations of an information space, and are easily understood by many information users. Hence, the treemap technique uniquely integrates a popular way of organizing information (hierarchical clustering) with a popular way of representing it (2D maps).

The treemap algorithm stems from a simple "slice-and-dice" idea. The available map space is split into vertical slices where every slice has an area proportional to the weight of the first level cluster it represents. These weights can be defined differently according to the requirements of the specific application. In our application, the initial weight of a cluster is defined as the total number of documents in the cluster and this weight is adjusted while zooming. After the first level slicing, a similar partitioning is applied to each vertical slice this time creating horizontal slices of lower level clusters where again the areas of these slices are proportional to the weights of the corresponding subclusters.

For a simple illustration of this technique, let's look at the hierarchy depicted in Figure 3.5. Assuming that the lowest level nodes C, D, E, H, I, J, and K are all individual files with a weight of 1, the weights of the other clusters will be 3 for A, 4 for B, and 2 for each of F and G. According to this weight structure, a treemap of the hierarchy is formed as follows: initially the map space is divided into two since there are only two first-level clusters: A, and B. The ratio of the area allocated for A to the area allocated for B will be 3 to 4 since A has a weight of 3 and B has a weight of 4. After the first slicing, the area for clusters A and B are partitioned similarly where the area of the A is divided into three equal portions (between C, D, and E), and the area for B is divided into two equal portions (between F and G). Finally, the areas for F and G are both divided into

Figure 3.5

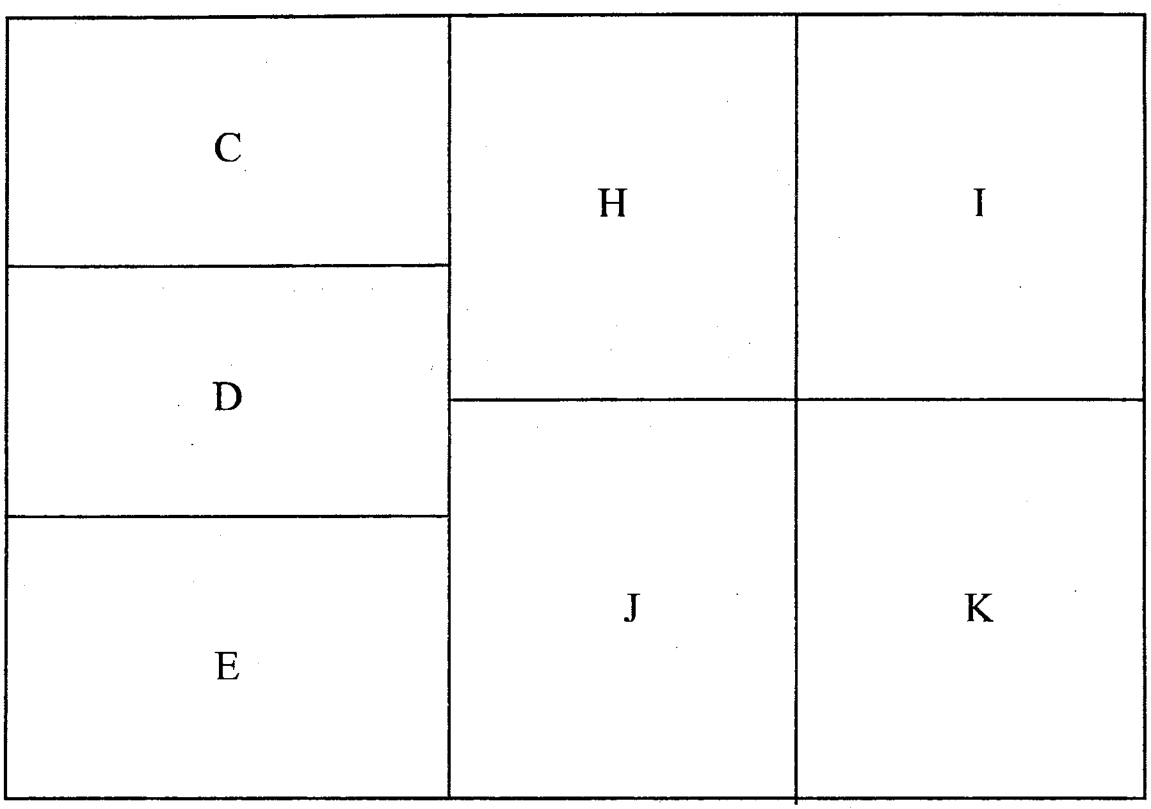
A simple hierarchy



two equal portions to represent H and I, and J and K respectively. The resulting treemap is shown in Figure 3.6.

Figure 3.6

The treemap of the hierarchy in Figure 3.5



In the original treemap algorithm by Johnson and Shneiderman (1991), the partitioning of the available map space continues until each individual document is represented on the map. In our application, we stop the partitioning before the area of a slice drops below a certain visibility threshold that we define. One caveat to this approach is that some clusters will always be too small, and will never be visible unless this threshold is ignored for a first-level cluster or a cluster that is one level below the zoomed-in cluster. Accordingly, our approach ignores this rule for those first-level clusters and the clusters that are one level below the zoomed-in cluster.

Since the original treemap algorithm shows every detail of the information structure on the overview, it does not require further zooming. As explained in the above paragraph, our implementation makes the clusters visible only if they are large enough (according to a predefined visibility threshold) hence there is a need for zooming to visualize the details of a cluster that are not available on the overview. For this reason, we propose an original approach to the design of the zooming capabilities in the visualization. Our implementation of the full zoom method simply entails redrawing the treemap taking the zoomed-in node (cluster) as the root. This approach is equivalent to assigning a weight of zero to all out-of-focus clusters.

Our approach to fisheye zooming is one out of many possible implementations of the idea. We propose a method of zooming that requires that the area of every cluster is redefined by increasing the weight of the zoomed-in cluster and all of its descendents such that all sub clusters of the zoomed-in cluster become visible i.e. exceed the visibility threshold. This entails that the size of the smallest sub cluster of the zoomed-in cluster would determine the scaling factor. In case this smallest sub cluster is too small to be visible even with the highest possible scale factor (a factor that would effectively result in full zoom) the threshold is ignored for that cluster, and it is allowed to be displayed as it was done in the case of the first level clusters. The scale factor is then calculated in a way to make the next to the smallest sub cluster exceed the visibility threshold if possible. This process is repeated until a large enough sub cluster is found, and the scale factor is calculated accordingly. Meanwhile, the out-of zoom clusters are allocated less space since the ratio of their weights to the weight of the overall hierarchy decreases. By this decrease, the areas of some of the out-of zoom clusters may drop below the visibility threshold. In case these clusters do not have a sibling in zoom, the algorithm modifies the view such that only the parents of the clusters below the visibility threshold are displayed. This way, the out-of zoom area is summarized to provide a context within which the zoomed-in details are visible. On the other hand, if an out-of zoom cluster is below the threshold and one of its siblings is in zoom, it can not be combined with its siblings since that would not allow the zoomed-in sibling to be visible. In that case, the cluster below the threshold is simply left as it is.

After a cluster is zoomed as described, the viewer of the visualization may be interested in one of the sub clusters thereof and want to further explore that sub cluster, or may focus on an out-of-zoom cluster. In the former case, the weights are adjusted exactly similar to the way it was done in the previous zoom. In the latter case, the weights of the zoomed-in clusters from the previous zoom are reset (i.e. set to their originals) before the procedure of weight adjustment is performed for the new zoomed-in cluster. The reason for this extra step is the need to make the visualizations independent of the previously visited clusters unless the browsed area is still inside those clusters.

The original treemap algorithm and the pseudo-code for some of our most important extended features are listed on figures 3.7, and 3.8.

Figure 3.7[5]

The Original Treemap Drawing Algorithm

by Johnson and Shneiderman 1991

DrawTree() *The node gets a message to draw itself*	*The Root node is set up prior to the original recursive call*
{	doneSize = 0; *The percent of this subtree drawn so far*
PaintDisplayRectangle();	*The node sends itself a Paint Message*
Switch (myorientation) {	*Decide to slice this node horizontally or vertically*
case HORIZONTAL:	
startSide = myBounds.left;	*Set start for horizontal slices*
case VERTICAL:	
startSide = myBounds.top;	*Set start for vertictal slices*
}	
if (myNode Type = = Internal{	*Set up each child and have it draw itself*
ForEach (childnode) Do {	
Childnode->**Setbounds**(startSide, doneSize, myOrientation);	*Set child's bounds based on the parent partition taken by previous children of parent*
Childnode->SetVisual;	*Set visual display properties (color, etc.)*
Childnode->**DrawTree()**;	*Send child a draw command*
}}}	
Setbounds(startSide, doneSize, parentOrientation)	
{ doneSize = doneSize + mySize;	*How much of the parent will have been allocated after this node*
switch (parentOrientation) {	*Decide which direction parent is being sliced*
case HORIZONTAL:	
myOrientation = VERTICAL;	*Set direction to slice this node for its children*
endSize = parentWidth * doneSize / parentSize	*How much of the parent will have been sliced after this node*
SetMyRect (startSide + offset,	*Left side, offset controls the nesting indentation*

[5] We appreciate the cooperation and access to Treemap97 code from Ben Shneiderman and the Human-Computer Interaction Lab at the University of Maryland

Figure 3.7 (ctd.)

```
                parentBounds.top + offset,              Top
                parentBounds.left + endside - offset,   Right
                parentBounds.bottom - offset);          Bottom
                startSide = parentBounds.left + endside;        Set start side for next child
case VERTICAL:
                        myOrientation = HORIZONTAL;             Set direction to slice this
                                                                node for its children
                        endSize = parentHeight * doneSize / parentSize

                        SetThisRect (parentBounds.left + offset,   Left side,
                        startSide + offset,                         Top
                        parentBounds.right - offset,                Right
                        parentBounds.top + endSide - offset);       Bottom
                startSide = parentBounds.top + endside;             Set start side for next child
```

Figure 3.8

Extended Features

```
Combine()                                   For only drawing the large enough clusters starting
                                            at the second level and below
        {if mynode.size / totalsize < threshold     If a node size is below the visibility
                                                            threshold
        parent.visible = true;              Make the parent visible
        visible = false;                    Make the node not visible
ForEach (sibling) Do                        Make the siblings not visible so that only their parent
                                            is displayed

        Sibling.visible = false;
}
}

Zoom()                                              Zooming in fisheye view
{
Smallest = FindSmallestchild(all children);         Find the small est child among all
                                                            children
        SmallestSize = Smallest.size;
If SmallestSize < mynode.size * threshold           If there is no scale big enough to
                                                      help this cluster size to exceed the
threshold
        {
        Repeat
        {
```

Figure 3.8 (ctd.)

```
smallest.visible = true;                                        Make the smallest child visible
                                                                regardless of the threshold
Smallest = FindSmallestchild(all but the previous smallest );   Find the next to the smallest
                                                                child and check if that can
                                                                exceed the threshold
        SmallestSize = Smallest.size;
        }
        until SmallestSize >= mynode.size * threshold;       until a big enough one is found.
        }
        Scale = (threshold * (totalsize - mynode.size)) / (SmallestSize - (threshold * mynode.size));
        Now calculate the scale to make the big enough child exceed the threshold
        Foreach (descendent) do
                {mynode.size = mynode.size * scale;}                Modify the weights
        Drawtree()                                           Redraw the tree with the new weights
        }
```

3.1.5 Implementation

We implemented a working prototype system as a proof of concept for the design ideas discussed in the previous parts of this section. The system facilitates the processes displayed in Figure 2.20 and performs them sequentially. A simple HTML form (see Figure 3.9) was designed to let the Web searcher enter a query. This form sends this query to an Active Server Pages (ASP) document[6] that communicates with the Alta Vista search engine. The query is sent to AltaVista in a format to receive the first 100 search results, and the "AltaVista Results" page including these 100 results is received by the ASP document. Next, the "AltaVista Results" page is parsed using a nonstandard ASP component (ASPTear) that has the built-in functionality to parse (tear) a given HTML document. This way, the title and URL of each search result is extracted from the "AltaVista Results" page. The name (see the following paragraphs for an explanation),

[6] We are thankful to Mr. Vishal Jangla for this part of the system development

the title, and the URL (obtained as explained above) of each of the 100 search results is stored in an index file. Next, the ASP document retrieves each of these results by calling their URLs. The resulting documents are in html format; therefore they are first converted to text files by stripping the html tags. The resulting text files are locally saved under their unique names, and they are already in the index file. The unique file names for the search results are for internal use by the system's clustering component. The clustering component, (IntelligentMiner's hierarchical clustering routine) works only with these names for creating a hierarchical cluster scheme. This clustering scheme is saved to a file, and is inputted to the visualization procedure.

Figure 3.9

The input form

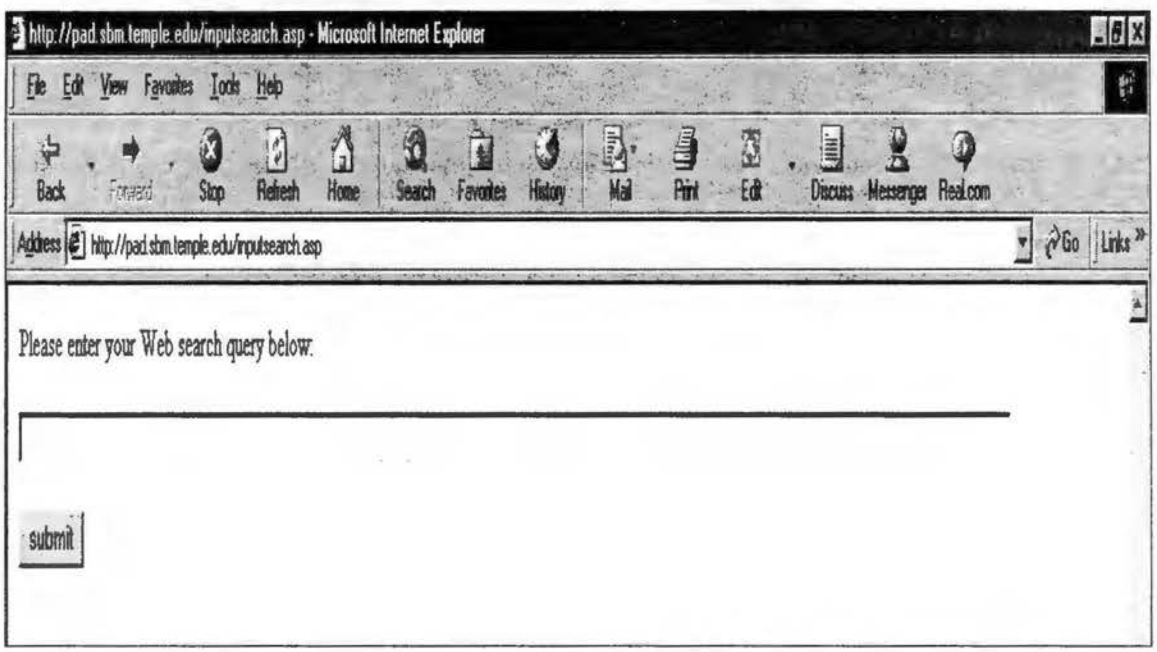
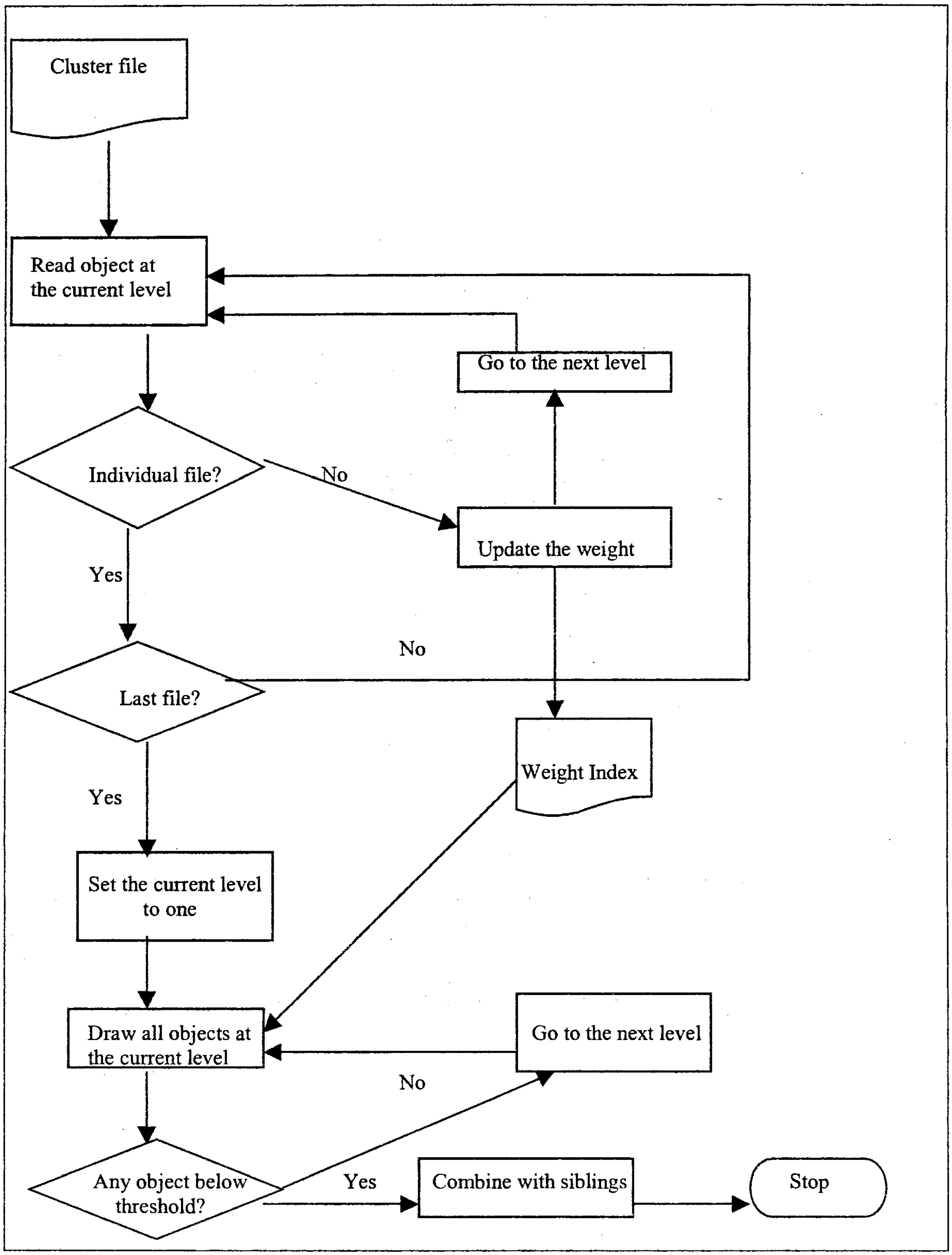


Figure 3.10

Flowchart of the Process for Visualizing the Hierarchy of Documents



We implemented a Java applet[7,8] that reads the hierarchical scheme information from the output of the clustering routine and the index file that is created by the ASP document. The applet visually displays the documents of the hierarchy based on our modification of Johnson and Shneiderman's (1991) TreeMap algorithm as discussed in the previous section (see figures 3.7 and 3.8). The applet uses the output of the clustering routine to label the clusters. Leaves of the hierarchy, i.e. individual files, are labeled by looking up the corresponding title for each file name in the index file that was created as explained before. Figure 3.10 displays the flowchart for the algorithm that performs this functionality.

As an example of this implementation, Figure 3.11 displays the visual overview of documents returned in response to the query "Name five different kinds of music that Sony is publishing".

Users of our system interact with this applet by focusing on a part of the displayed hierarchy. The algorithm for the zooming functionality of the visualization was explained in Figure 3.8. Figure 3.12 displays a flowchart of the algorithm that performs the (fisheye) zooming function of our system. When a user reaches a leaf of the hierarchy, i.e. an individual file, and then clicks on that part of the visualization to further explore the individual page, the applet looks up the URL of the file from the index file and retrieves the page accordingly.

Figure 3.13 summarizes the functionality of our system. This summary includes the main processes, the inputs and outputs produced, and the specific program segments

---

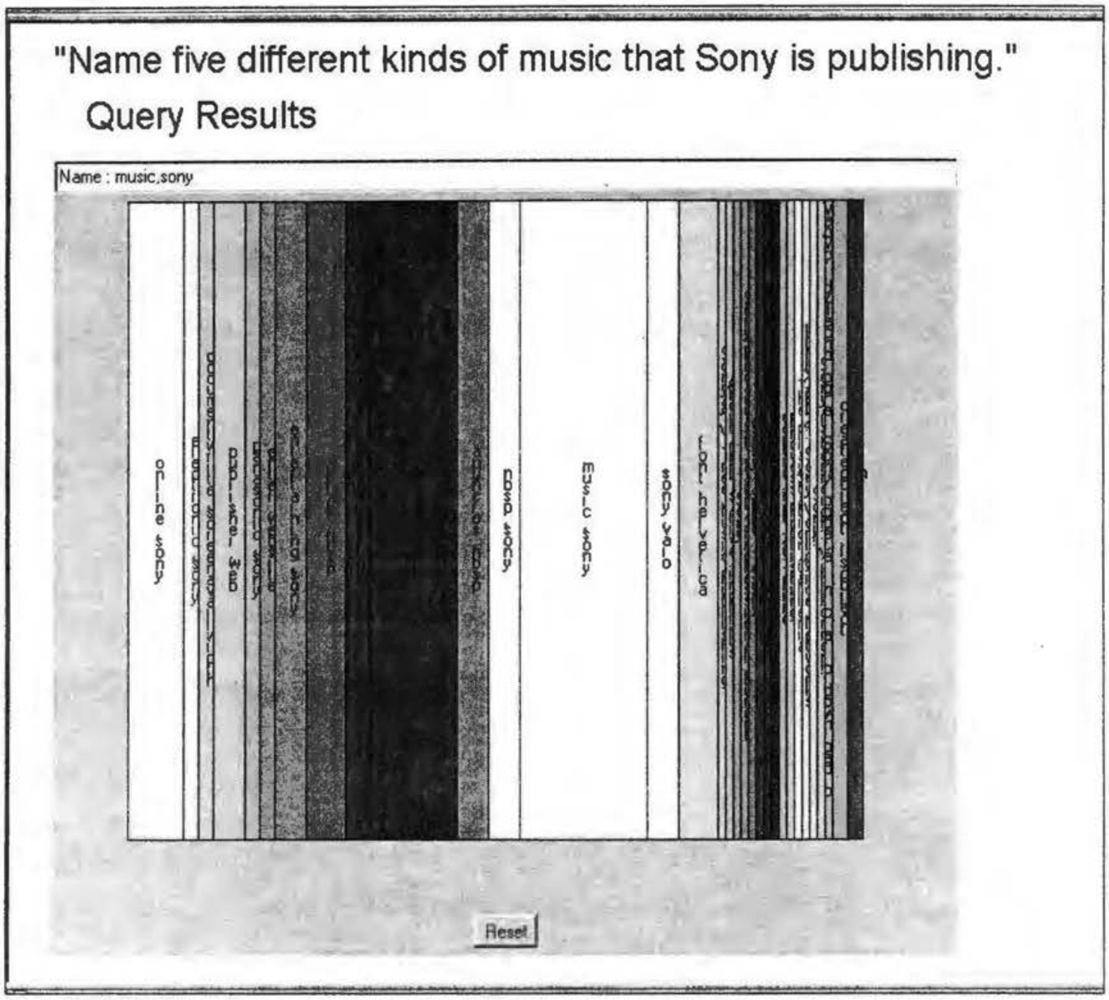
[7] Mr. Someshwar Baldawa is highly appreciated for his contribution in the coding of the algorithm.

[8] For the sake of convenience, we refer to this applet as applet.java in the following discussions

Figure 3.11

The Visual Overview of Documents Returned in Response to the Query

"Name five different kinds of music that Sony is publishing."



that execute each of the processes. Two of these main processes, namely "Visualize hierarchy" and "Zoom" are performed by means of the Java applet, the implementation of which is based on the algorithms of figures 3.7 and 3.8, and is explained by means of the charts in figures 3.10 and 3.12 respectively. Figure 3.13 is a

Figure 3.12

Flowchart of the Process for Fisheye Zooming a Part of the Visualization

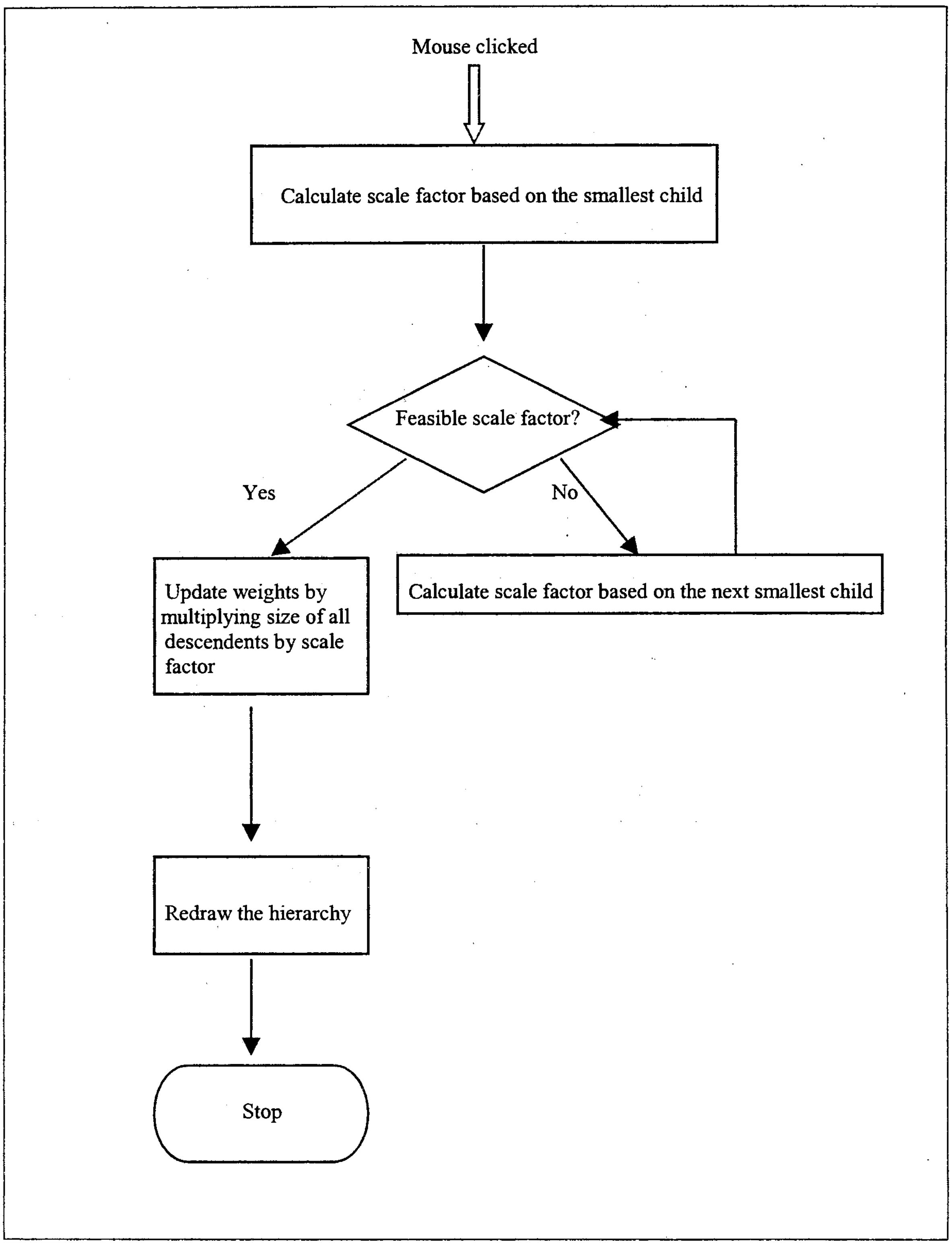
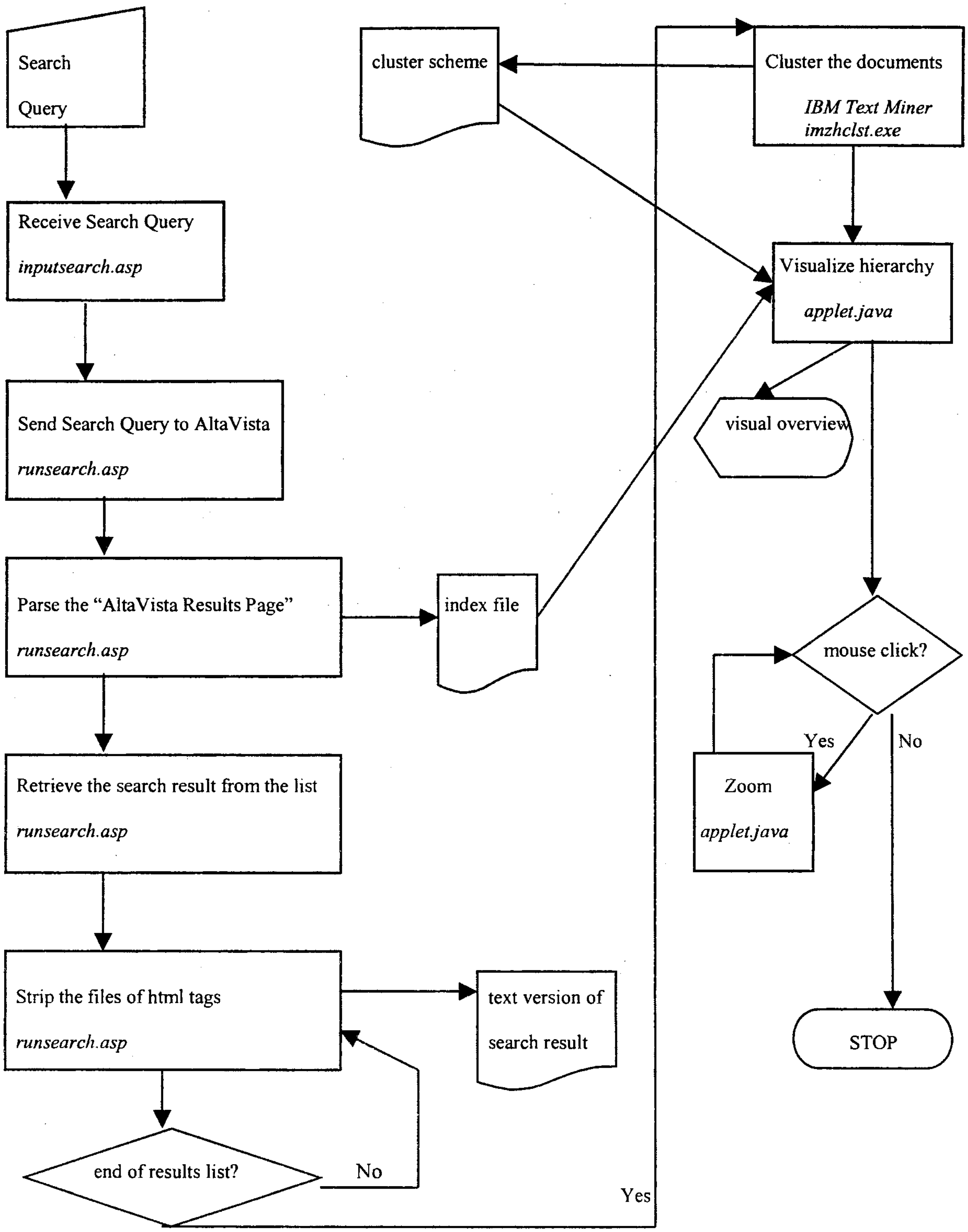


Figure 3.13

The System Flowchart



more detailed version of the system chart in Figure1.1. In Section 3.3.3, we describe how we used this prototype in our user study. Before that, we discuss the details of the conceptual research model and the statement of research hypotheses in the following section.

3.2 Hypotheses

As displayed by the model diagram in Figure 1.2, four main independent variables, namely the presentation method (interface), task, amount of training, and contextual (individual) variables have an effect on end-user success and satisfaction. The only independent variable that is manipulated in the study is the presentation method of Web search results, and it can take three levels: fisheye view visualization, full zoom visualization and no visualization. The other independent variables are either held constant (amount of training and task) or controlled (contextual variables).

Of the dependent variables, end-user success has two dimensions: effectiveness and efficiency. We operationalize effectiveness and efficiency as suggested by Roussinov (1999), Morse (1998), and Tan and Benbasat (1990) among others. The number of correct answers given in a limited time to a set of objective questions that have their answers within the search results is a surrogate measure for effectiveness while the time tocomplete the task of answering these questions is a surrogate for efficiency. End-user satisfaction is measured by the users' satisfaction survey scores for each display. According to this operationalization, the (alternative) hypotheses of the study are:

H1a: Existence of visualization increases the number of correct answers.

H1b: Existence of visualization reduces the time to complete the task of answering the questions.

H1c: The visual systems result in higher user satisfaction than the text-based system does.

The hypotheses in this first group aim to measure the success of visualization. The second group hypotheses aim to measure the success of the fisheye zoom method in comparison to the full zoom method, and are formulated similarly as follows:

H2a: Use of the fisheye zoom instead of the full zoom increases the number of correct answers.

H2b: Use of the fisheye zoom instead of the full zoom reduces the time to complete the task of answering the questions.

H2c: The fisheye zoom system results in higher user satisfaction than the full zoom system does.

As we mentioned before, few studies on visual information retrieval include and report on usability of the approaches and related empirical tests. This study addressed this issue by the empirical testing of the above hypotheses by means of a controlled experiment. The details of the experimental design are presented in Section 3.3.

3.3 Experimental Design

3.3.1 Subjects

The subjects of the experiment are college students enrolled in upper level undergraduate or graduate business (mostly MIS) classes in a large northeastern university.

3.3.2 Tasks

Finding the right tasks to test the usability of a design idea is a challenging endeavor, especially in Web-based empirical studies. Our main motivation in this study is to support the browsing of the results of a search query that retrieves a large number of pages (hits). Accordingly, our first criterion for the search tasks for the experiment is that they result in a large number of hits. Similarly, we want the tasks to have multi aspects and to produce a large number of clusters.

The purpose of our experiment is to test the success of our design principles hence we do not want the personal traits or backgrounds of the subjects to influence the results. For this reason, another criterion for the search tasks is that they be on general topics so that a group of subjects will not be systematically more knowledgeable about the search tasks compared to the other subjects.

Searching for information on the Web is a cognitive task. To that effect, it is worthwhile to briefly review the major groups of cognitive tasks to gain a deeper understanding of the experiment's objectives. According to an old taxonomy by Bloom (1956) there are six major areas in the cognitive domain.

1. Knowledge: The ability to recall or recognize ideas, facts, etc. in a situation.

2. Comprehension: The ability to receive ideas, etc. and make use of them without relating it to other materials or seeing its implication.
3. Application: The ability to use abstraction, rules, methods, and principles in concrete problems or situations.
4. Analysis: The ability to breakdown a communication or concept into its constituent elements.
5. Synthesis: The ability to combine pictures, parts, and elements to form a new pattern or arrangement.
6. Evaluation: The ability to make quantitative and qualitative judgments about the extent to which materials and methods satisfy criteria.

We argue that depending on the search purpose, information search tasks can require one or a combination of knowledge, analysis, synthesis, and evaluation. Along the same lines, Marchionini (1995) defined the common goals of information search in a range from finding a narrow set of items in a large collection that satisfy a well-understood information need (known-item search) to developing an understanding of unexpected patterns within the collection (browsing). Based on these general guidelines, Shneiderman (1997) classified information search objectives into the following general groups:

1. Specific fact-finding (Searching *directly* for a readily identifiable outcome)
2. Extended fact-finding (Searching *indirectly* for relatively uncertain but replicable outcomes)
3. Open-ended browsing (Gaining an understanding of a general subject area)
4. Exploration of availability (self explanatory)

In the Web domain, the availability of material is subject to continuous change. Hence, it is not easy to pursue the fourth search objective. Open-ended browsing is invaluable in real-life information search activities. However, in a controlled experiment it is very difficult to measure the outcomes of a loosely defined objective such as "finding new work on voice recognition in Japan", or "possible relationships between carbon monoxide levels and decertification". Subsequently, the search tasks in this study were mainly on (specific or extended) fact finding.

Roussinov (1999) used questions discussed by the panel on Web Search at the 1998 ACM Conference on Advances in Information Retrieval (see Table 3.1) in an experimental study, and found significant differences between the performance of different search tools for some of these questions. Five of these questions (1, 2, 6, 7, 10) can be considered the specific fact-finding kind. Roussinov concluded that two of these ten questions (6 and 8) were extremely difficult since no subject could correctly answer them. Among the remaining eight questions, four of them (1, 3, 9, and 10) were identified as "tough" questions where the others (2, 4, 5, and 7) were identified as "easy" ones. Roussinov also found that the performance of the subjects was not significantly different between the easy and difficult questions.

Using specific fact-finding questions is a good way to test the success of a presentation system. These questions have their answers in (a) particular site(s) within the collection of search results yet finding those answers, especially in a fast manner, requires the ability to quickly overview the document collection as well as easily focusing on a specific part of a collection when needed. We believe that the visual systems especially with the fisheye zooming would facilitate this kind of a fast focusing and refocusing, and will better support the information seekers.

Table 3.1

Roussinov's experimental questions

1. I want to find where Max Beerbohm, the English caricaturist, lived in at the end of his life.
2. What does it cost to ride on the upper deck of the Star Ferry across Hong Kong harbor to Tsimshatsui?
3. Where can I get good pfeffersteak in Hagerstown, MD, USA?
4. If I visit Singapore, what, if any, buildings designed by I. M. Pei's can I see there?
5. Names of hotels in Kyoto (Japan) that are near the train station.
6. What is the cost of overnight train tickets, including sleeper accommodations (double occupancy) from Paris to Munich?
7. How long does it take to get by train from Copenhagen to Oslo?
8. Was the Ring Cycle performed at Bayreuth, Germany, in summer 1998?
9. I'm looking for the names of campgrounds around Lake Louise (Canada) that have showers.
10. I need a map showing the location of the Penfold's winery in Australia.

Extended fact-finding questions require more synthesis than specific fact-finding questions. The indirect nature of the search to find answers to questions of this sort makes it essential to have a general understanding of the information space to connect different pieces of information. The visual system particularly with the fisheye zooming capabilities as described should be a good candidate to render this kind of a general understanding and the ability to quickly browse through different areas on the information map. Shneiderman (1997) lists three example questions of extended fact-finding as seen in Table 3.2. Note that the answers to each one of these questions constitute a list rather than a single entity. To be able to make the answers to these questions less vague and easier to evaluate, we modified them and appended the list with three more questions of this sort to constitute the list in Table 3.3.

Table 3.2

Shneiderman's extended fact-finding questions

1. What other books are by the author of Jurassic Park?
2. What kinds of music is Sony publishing?
3. Which satellites took images of the Persian Gulf War?

Table 3.3

The extended fact-finding questions

1. Find two other books by the author of Jurassic Park.
2. Find five different kinds of music that Sony is publishing.
3. Name three satellites that took images of the Persian Gulf War.
4. How many swimming medals did the country that had won the most gold medals in swimming in the 1972 Olympics win in the 1996 Olympics?
5. What are the two most recent movies by the director of "Shining"?
6. Which one of Luxembourg's neighbors has the highest literacy rate?

Using the questions from tables 3.1 and 3.3 we designed three tasks: Task A, Task B, and Task C with virtually similar difficulty levels. This similarity allows us to control for the "task" variable in the conceptual model of Figure 1.2. To establish this similarity, we picked one "easy" and one "tough" question according to Roussinov's findings from Table 3.1 for each task. A similar reasoning was applied in assigning the questions of Table 3.2 to each task. According to this arrangement, the tasks in Table 3.4 were formed. Our next task was to do the initial testing on this list to finalize the tasks that would be used in the user study.

Table 3.4

The experimental tasks

Task	Questions
A	1. Where can I get good pfeffersteak in Hagerstown, MD, USA? 2. If I visit Singapore, what, if any, buildings designed by I. M. Pei's can I see there? 3. Find two other books by the author of Jurassic Park. 4. Name three satellites that took images of the Persian Gulf War.
B	1. Find the names of hotels in Kyoto (Japan) that are near the train station. 2. I'm looking for the names of campgrounds around Lake Louise (Canada) that have showers. 3. Find five different kinds of music that Sony is publishing. 4. How many swimming medals did the country that had won the most gold medals in swimming in the 1972 Olympics win in the 1996 Olympics?
C	1. How long does it take to get by train from Copenhagen to Oslo? 2. I need a map showing the location of the Penfold's winery in Australia. 3. What are the two most recent movies by the director of "Shining"? 4. Which one of Luxembourg's neighbors has the highest literacy rate?

3.3.3 Control Variables

The amount of training was held constant for each subject. Therefore, the "amount of training" variable is controlled similar to the task variable. Age, sex, native language (English vs. other), cognitive style, and Web search experience are used as control variables.

Figure 3.14 displays our research model with the complete set of operational variables based on the conceptual model of Figure 1.2.

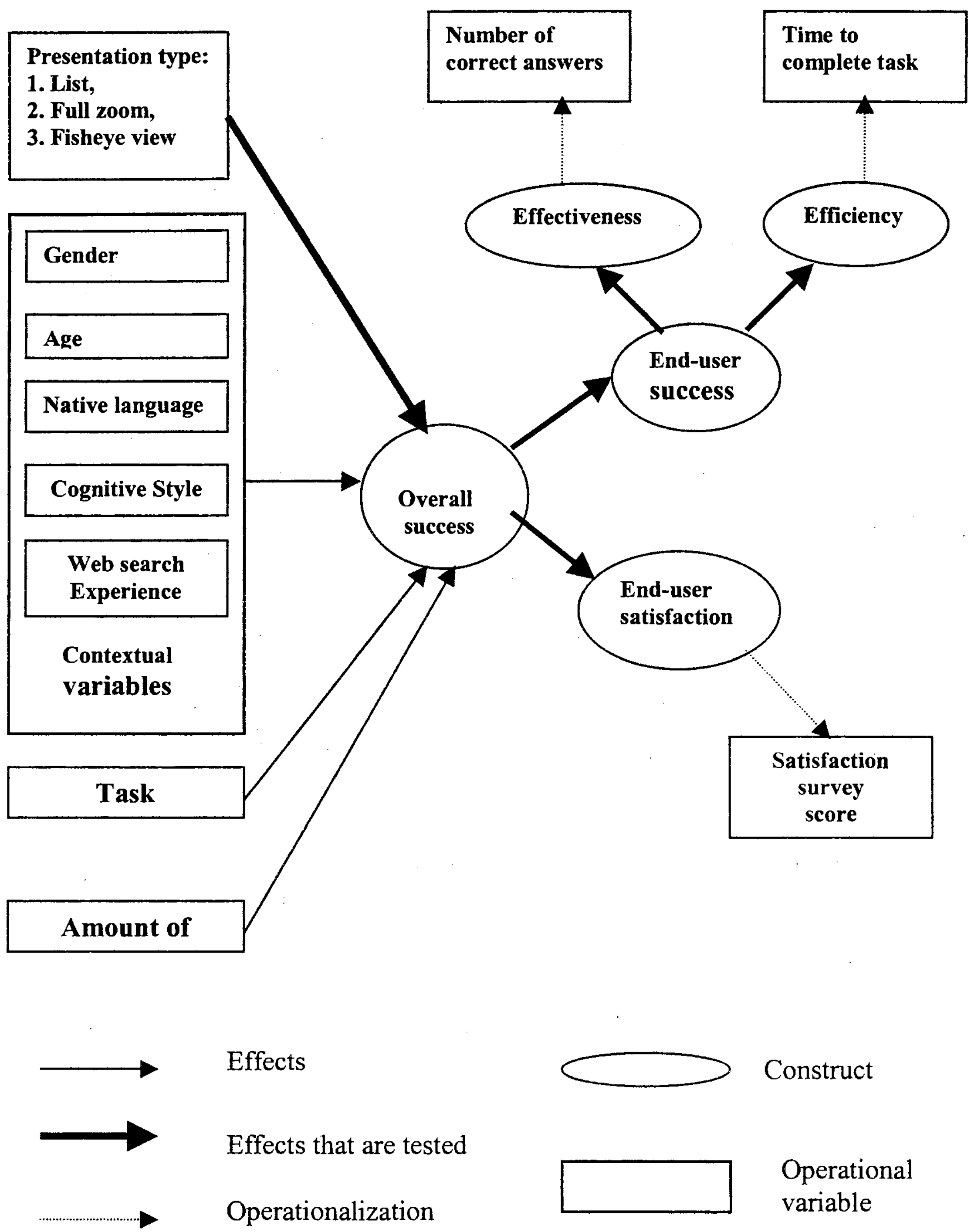
3.3.4 Measurements

The measurement of "gender", "age", "native language", and "number of correct answers" variables are straightforward. The "time to complete task" was measured by including a timestamp to mark the beginning of each phase of the experiment, and the time each subject submitted his/her answer. To measure "Web search experience", we adopted and used a two-item scale (Wang et al. 2000) composed of the frequency, and the duration of Web search engine use. Table 3.5 displays the survey that was conducted to collect demographic data and Web search experience.

We used the Group Embedded Figure Test (GEFT) (Witkin et al. 1971) for determining "cognitive style". This test is known to measure a very salient, i.e. field dependence, dimension of cognitive style, and has been shown to be fairly reliable (Witkin et al. 1971). Subjects of the GEFT are shown simple figures that are "embedded" in more complex ones and are asked to identify the simple figure within

Figure 3.14[9]

The Research Model



[9] This model does not explicitly show interaction between the independent variables, or dependent variables yet the existence of interactions must be tested for in an empirical study.

the larger one. Depending on how many such simple figures they can identify , they are assigned a score on a range between 0 and 18. The subjects that score high on this test

Table 3.5

The Initial Questionnaire

Please answer the following questions. 1. What is your age group? A) 19 or younger B) 20-22 C) 23-25 D) 26 or older 2. What is your sex? A) Male B) Female 3. What is your native language? A) English B) Not English 4. How long (approximately) have you used a search engine (e.g. Lycos, AltaVista, Northern Light, etc.) to search for information on the Web? A) Never B) For two or less than two years C) For more than two, but less than four years D) For four or more years 5. How often (on the average) do you use a search engine (e.g. Lycos, AltaVista, Northern Light, etc.) to search for information on the Web? A) Never B) Once a month or less often C) More often than once a month, but not as often as once a week D) Once a week or more often

are considered to be more field independent. Although some authors have made the assumption that for relatively field dependent individuals the Web is expected to be a much more difficult environment (Wang et al. 2000), we refrain from making that assumption, but rather use field-independence to control for individual differences. To that effect, in our further analyses, we refer to the field-independence with the assumption that the reader has an understanding of its role in our study.

The final scale we discuss is a multi-item one adopted from Stasko et al. (2000) that was used for measuring "satisfaction". The satisfaction survey is based on a 7-point Likert scale and contains the items displayed in Table 3.6.

Table 3.6

The Satisfaction Survey

1. There are definitely times that I would like to use this system.
2. I would like this system available for my use all the time.
3. I found this system useful.
4. I found this system confusing to use.
5. I liked this system.

3.3.5 Procedure

The first step we took before conducting the user studies was to validate the usability of the experimental tasks (questions) that we had identified. For this purpose, we ran the queries one by one to find out whether the answers to the questions can be found within the first one hundred results that are returned by the search engine, and whether finding the answers required some nontrivial effort. We decided this kind of a prescreening would enhance the quality of the data that are collected, and the need to eliminate questions after data collection would be less likely. During this screening process we found out that since some of the questions on our list were part of different experiments before, they were extensively studied and the answers were explicitly published on the Web. This led to the elimination of the first two questions from both tasks A and C, and the second question from task B. We replaced these questions by similar questions that we made. We also discovered that the fourth questions of tasks A and C were too difficult to answer just by consulting the first hundred Web pages, and they were replaced by more reasonable questions. Finally, we decided that the effort to answer the third questions of tasks B and C were too trivial for us to keep them on our list, and we replaced those questions with more nontrivial ones. This modification in the tasks led to the final list of questions to be tested in the experiment as seen in Table 3.7.

We conducted controlled experiments with several groups of business (mostly MIS) students from a large northeastern university to empirically test the hypotheses formulated in the previous section. Each experiment session started with the subjects' filling out a questionnaire on demographic data (the control variables) such as age, gender, and native language (English vs. other). Data on the subjects' Web search experience were also collected by means of the same questionnaire. In addition, the subjects went through a cognitive style test. The details of the instruments used for these measurements were discussed in part 3.3.4.

Table 3.7

The modified list of the experimental tasks

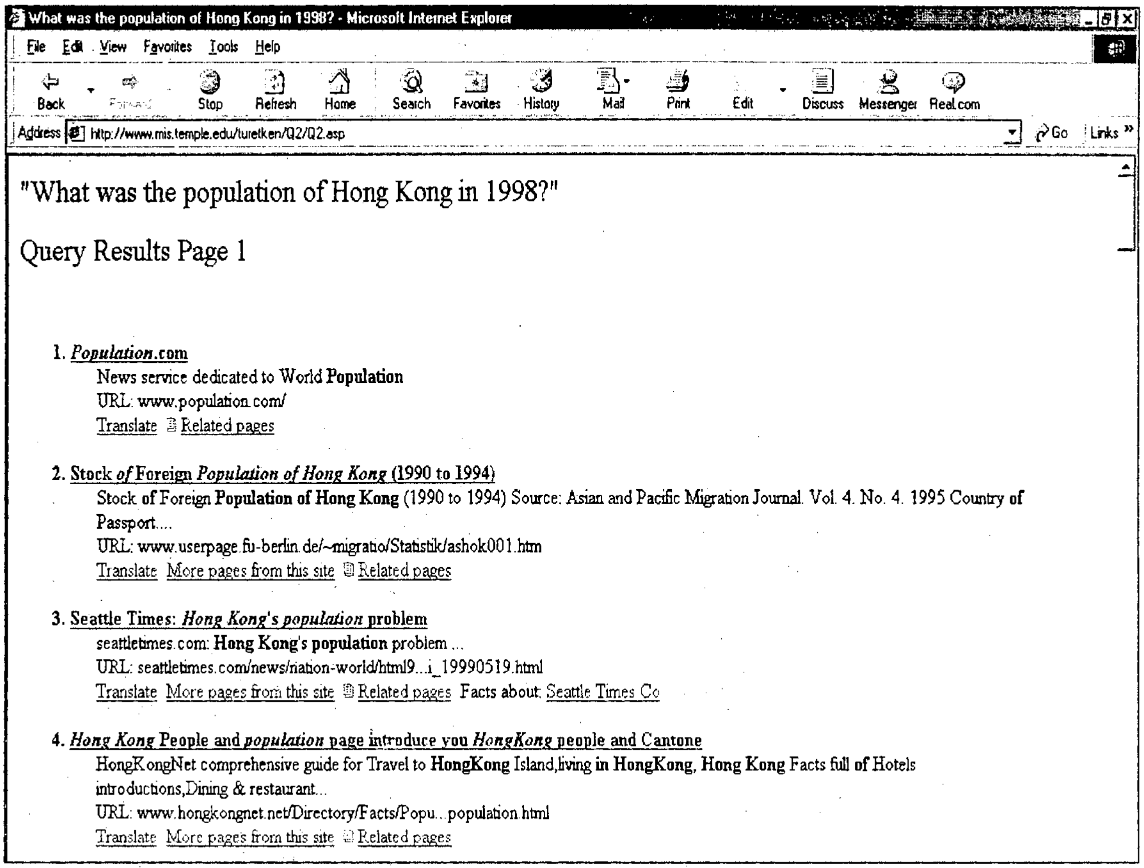
Task	Questions
A	1. Where can I get good filet mignon in Madison, WI? 2. What was the population of Hong Kong in 1998? 3. Find two other books by the author of Jurassic Park. 4. Name three shows that took stage in Broadway in 1989
B	1. Find the names of two hotels in Kyoto (Japan) that are near the train station. 2. How long does it take to get by train from Paris to Munich? 3. I need a map of Kusadasi, Turkey. 4. Where did William Shakespeare die?
C	1. I'm looking for the names of campgrounds around Lake Tenkiller (Oklahoma) that have showers. 2. How many track medals did the country that had won the most gold medals in track in the 1972 Olympics win in the 1996 Olympics? 3. What are the two most recent movies from the director of "Full Metal Jacket"? 4. Which countries are the neighbors of Ukraine?

For the purposes of the experiment, the subjects were randomly assigned to one of three groups. Every subject underwent a training session, which resembled the experimental phases, and familiarized the subjects with the experimental procedure and the Web-based forms that they were asked to fill in. For training, we used a similar task (see Figure 3.11) to the ones that they would actually be working on (Table 3.7) during the experiment.

After the training, each group of subjects underwent three phases of experimentation. In each of the three phases, subjects were given tasks A, B, and C in the same order. The difference was in the presentation of search results. For the first group, Task A had no visual support while Task B was supported by the full zoom visual system and Task C was supported by the fisheye zoom visual system. For the second group, Task C had no visual support while Task A was supported by the full zoom visual system and Task B was supported by the fisheye zoom visual system. Finally for the third group, Task B had no visual support while Task C was supported by the full zoom visual system and Task A was supported by the fisheye zoom visual system. See figures 3.15, 3.16, 3.17, and 3.18 for the alternative ways of presenting the results to the query for answering the second question of Task A. Figure 3.16 is the visual overview, which is common to both methods of visualization. Figures 3.17 and 3.18 display the alternative ways of zooming the "populous world" section of the overview in Figure 3.16.

Figure 3.15

Text-based Presentation of Search Results to Query 2


What was the population of Hong Kong in 1998? - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss Messenger Real.com

Address http://www.mis.temple.edu/turetken/Q2/Q2.asp Go Links »

"What was the population of Hong Kong in 1998?"

Query Results Page 1

1. *Population*.com
   News service dedicated to World **Population**
   URL: www.population.com/
   Translate Related pages

2. Stock *of* Foreign *Population of Hong Kong* (1990 to 1994)
   Stock of Foreign **Population of Hong Kong** (1990 to 1994) Source: Asian and Pacific Migration Journal. Vol. 4. No. 4. 1995 Country **of** Passport....
   URL: www.userpage.fu-berlin.de/~migratio/Statistik/ashok001.htm
   Translate More pages from this site Related pages

3. Seattle Times: *Hong Kong's population* problem
   seattletimes.com: **Hong Kong's population** problem ...
   URL: seattletimes.com/news/nation-world/html9...i_19990519.html
   Translate More pages from this site Related pages Facts about: Seattle Times Co

4. *Hong Kong* People and *population* page introduce you *HongKong* people and Cantone
   HongKongNet comprehensive guide for Travel to **HongKong** Island,living **in HongKong**, **Hong Kong** Facts full **of** Hotels introductions,Dining & restaurant...
   URL: www.hongkongnet.net/Directory/Facts/Popu...population.html
   Translate More pages from this site Related pages

Figure 3.16

Visual Overview of the Search Results to Query 2

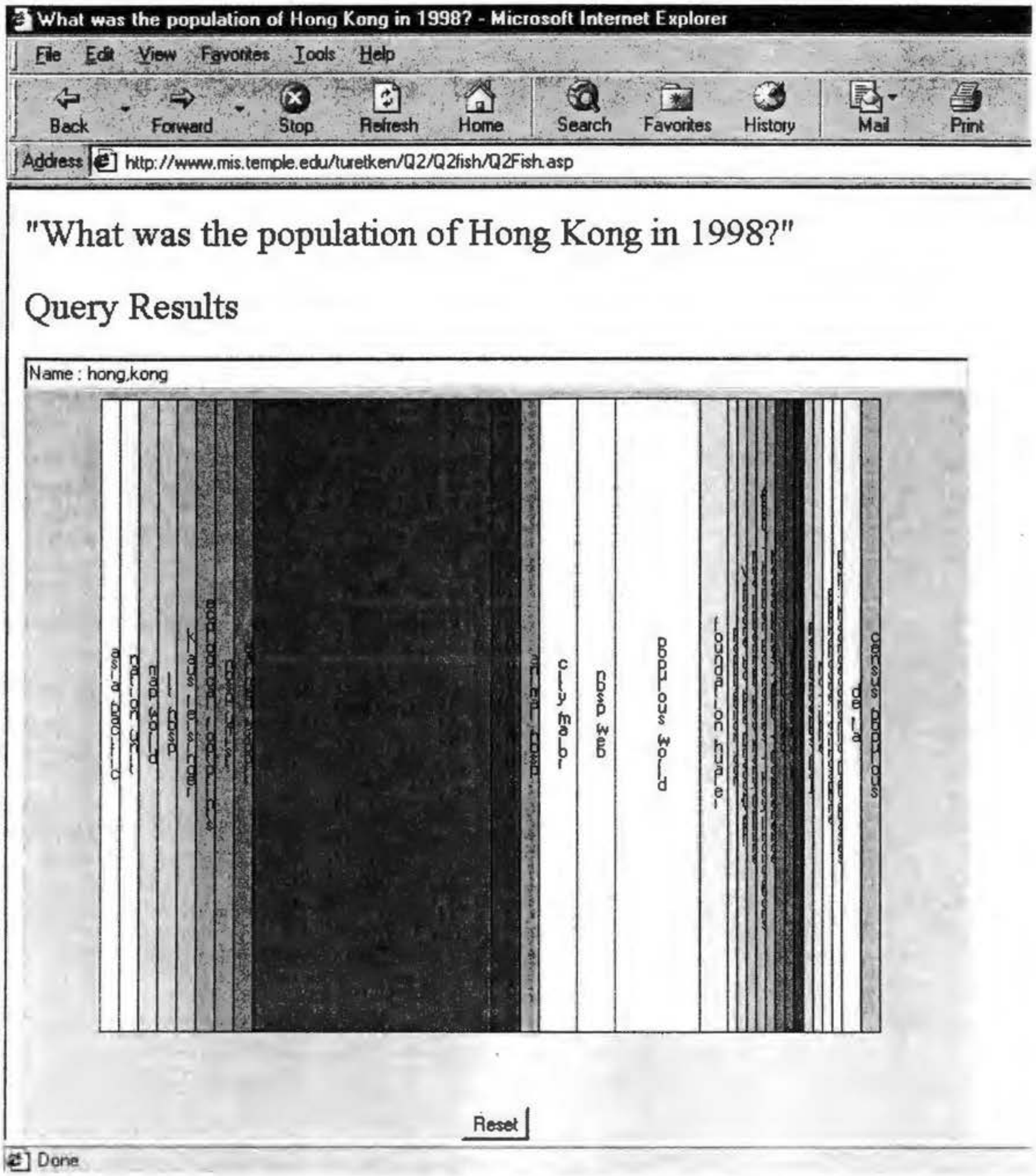


Figure 3.17

"Full" zooming of the "populous world" section of Figure 3.16

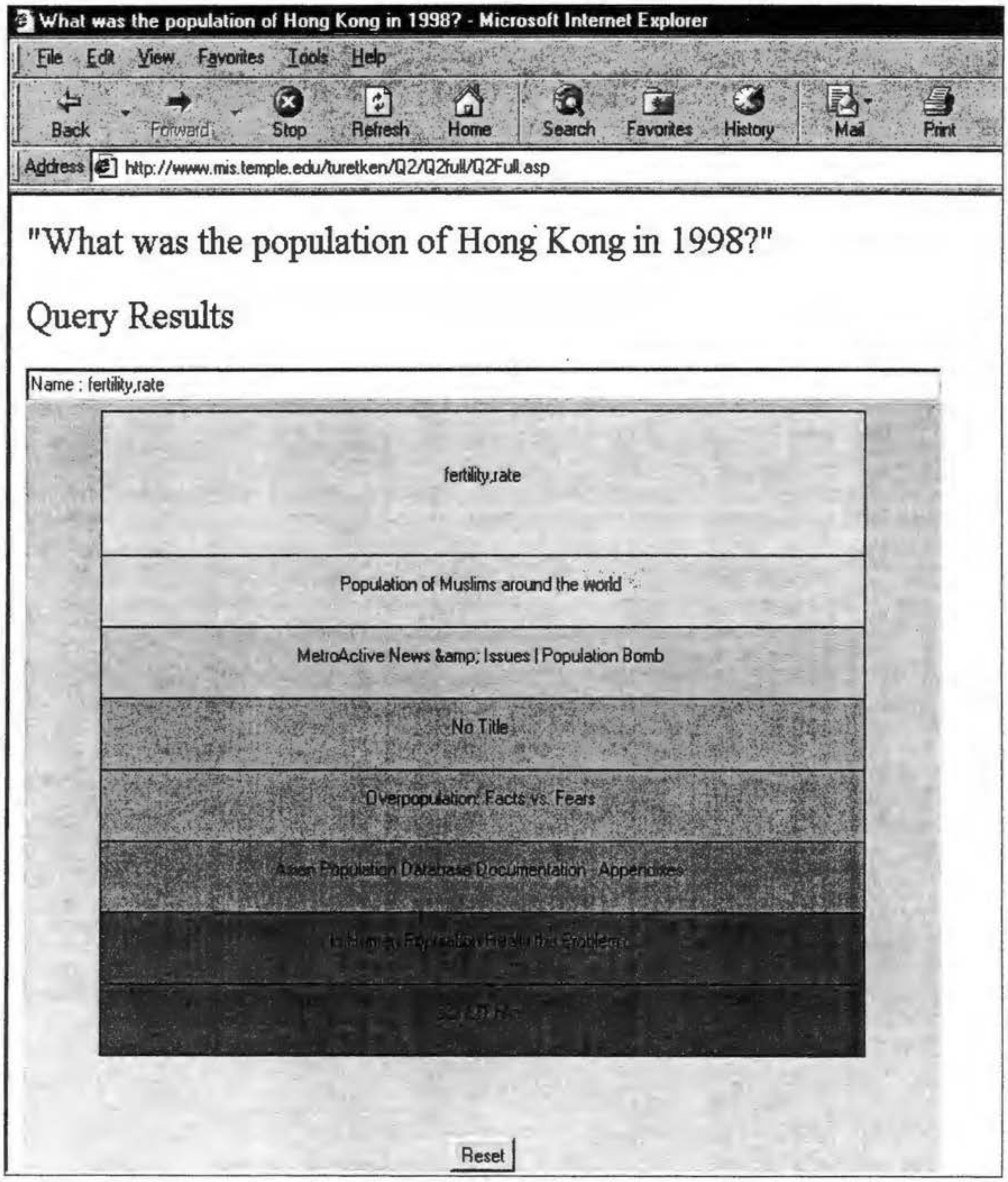
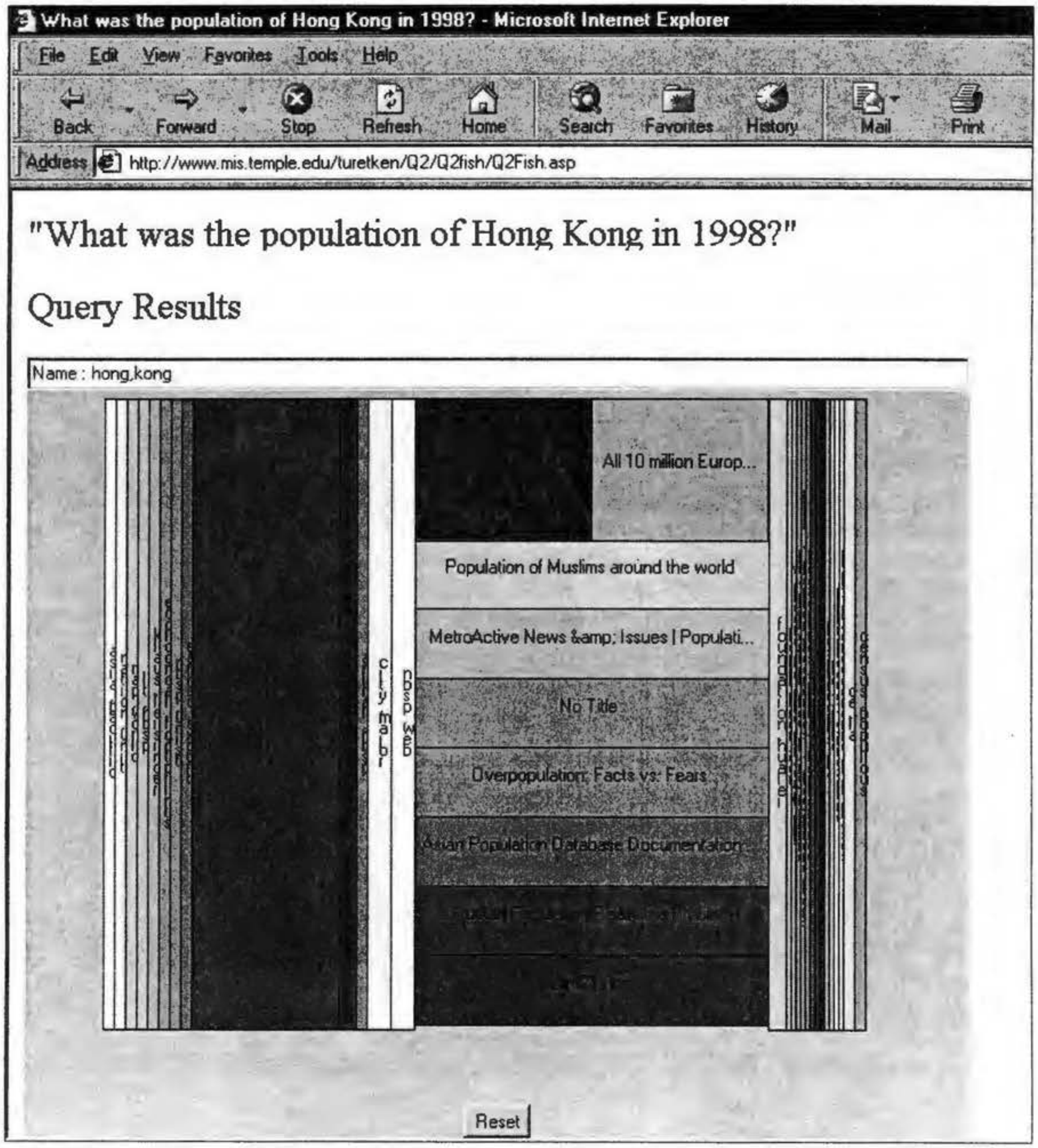


Figure 3.18

"Fisheye" zooming of the "populous world" section of Figure 3.16



The experimental design that we described facilitates all three modes of presentation methods to be used for all three tasks. Assuming that the order that the subjects are exposed to the presentation methods has no significant effect (i.e. no phase-task combination and learning effect), the success (number of correct questions and time to finish the task) and satisfaction of the subjects using a specific presentation method can be found by aggregating the success (i.e. performance) and satisfaction measures from the three different groups, and the data collected by this design can be analyzed as if they were collected by means of a repeated measures design. We test the validity of the assumption in Chapter 4. The reason that we chose such a design over a regular repeated measures design where all subjects would go through the same task and presentation method combination in the same order is to control for the effect of the sequence the subjects are exposed to the differing presentation methods. If significant differences between presentation methods were found with a regular repeated measures design, we would have no way of knowing whether this was a learning effect, and the difference was caused because of this learning, i.e. the different sequence that the subjects were exposed to the presentation methods. For that reason, it is obvious that the internal validity of a regular repeated measures design would not be as good as the one we are proposing herein. Our experimental design is displayed in Table 3.8.

During an experimental session, the subjects were given 12 minutes (an average 3 minutes per question) for the completion of each experimental phase, and were reminded of the time left for the specific phase every three minutes. At the end of each phase, the subjects were asked to evaluate the mode of presentation that they had experienced in that particular phase. The details on the instrument to render this evaluation were displayed in Table 3.6.

A total of six sessions of the experiment (each time with a different sample) were held. The first one of these sessions was used for testing the experimental procedure, and the second one did not yield usable data due to a technical problem with the connection of the Web forms to the central database. This resulted in a total sample size of 78 subjects.

Table 3.8

The Experimental Design

Phase	**Task/Support**		
	Group 1	**Group 2**	**Group 3**
Phase 1	Task A No visualization	Task A Full zoom	Task A Fisheye zoom
Phase 2	Task B Fisheye zoom	Task B No visualization	Task B Full zoom
Phase 3	Task C Full zoom	Task C Fisheye zoom	Task C No visualization

4. RESULTS

In this section, the results of the experiment and statistical analyses of the data are discussed.

4.1 Preliminary Analyses

4.1.1 Data Representation

Before the statistical analyses, it is worthwhile to discuss the representation of the data for a better understanding of the analysis process. As will be recalled from the discussion of the scales, two of the independent (control) variables, "sex" and "native language" in our research model were originally measured on single-item categorical (nominal) scales (male vs. female, and English vs. Not English) therefore there is no need to transform these variables for analysis purposes. Similarly, the "age" variable was measured on an ordinal scale where the first category represents the youngest age group, and the last one represents the oldest hence this variable does not require any further transformation, either. Two of the dependent variables, "number of correct answers"[10] or briefly "score", and "time" (to complete each task, in seconds) variables were measured on a continuous (ratio) scale, and are not transformed. The other variables in our model are not as amenable for numerical analyses therefore needed further elaboration. Next, we discuss this further effort in transforming those variables for the facilitation of data analysis.

---

[10] The number of correct answers was determined simply by following the URLs that the subjects provided as the most relevant sources for answering the questions and finding out whether the answers to the questions can be found therein.

The measurement and interpretation of the control variable, "cognitive style" was explained in the previous chapter. Originally, this measurement yields a score that is an integer between 0 and 18, which does not provide categories of cognitive style that we can classify our subjects into. Witkin et al. (1971) classify their subjects into four quartiles based on their score distribution. Similarly, the exploration of the distribution of the scoresof our subjects led us to categorize our test sample into four different cognitive style groups, where the first group constitutes the most field independent individuals, and the fourth group constitutes the most field dependent ones.

The "Web search experience" ("Web Experience") variable was measured through two items, and therefore does not have a simple representation. A quick overview of the answers that were given to the two "Web search experience" questions (i.e. questions 4 and 5) of the survey in Table 3.5 revealed that there were four natural categories of subjects in our sample: the subjects with "high" Web search experience marked "D" for at least one of the questions 4 and 5, i.e. they indicated frequent Web search engine use, or long period of familiarity with Web search engines. On the other hand, the subjects with "very low" Web search experience marked "A" for both questions meaning that they had not used a search engine before, and those with "low" Web search experience marked "B" for both questions indicating that they had a relatively short exposure to web search engines (less that 2 years), and that they very seldom use Web search engines (once a month or less frequently). We classified all the other subjects, i.e. those who marked "C" for both questions, into the "moderate" Web search experience category. This way the "Web search experience" variable is transformed into a categorical one with four levels, and was represented by labels 1 through 4.

As displayed in the last section of Chapter 3, the measurement scale we used for our third dependent variable, "satisfaction" is a 7-point Likert scale with five items. In order to decide whether this scale is reliable (i.e. repeatable) and whether these different items can be combined into a single satisfaction score, we performed two different analyses. First, to measure the reliability of the scale, we calculated coefficient α[11]. This calculation is possible if the scale is a ratio, or at least an interval scale therefore would not be possible with the original categorical coding of the variable. For this reason, the values of responses for items 1, 2, 3, and 5 in the scale were represented on a scale that ranges from 1 through 7 (1 representing minimum possible satisfaction, and 7 representing the maximum). A very similar transformation was applied for item 4, yet this time doing the opposite representation for the different answers, namely representing the answer "A" by 7, and "G" by 1 since this item captures a negative opinion, and strong agreement with the statement here implies low satisfaction.

After the mentioned transformations, coefficient α is calculated, and as shown in Figure 4.1, the value of α is fairly high for this scale hence it is safe to assume that the scale is reliable.

[11] For this and the following statistical calculations and analyses, we used SPSS, version 10

Figure 4.1

Reliability Calculations (SPSS Output)

Method 2 (covariance matrix) will be used for this analysis

R E L I A B I L I T Y   A N A L Y S I S   -   S C A L E   (A L P H A)

1. P1S1
2. P1S2
3. P1S3
4. P1S4
5. P1S5

Correlation Matrix

	P1S1	P1S2	P1S3	P1S4	P1S5
P1S1	1.0000				
P1S2	.7768	1.0000			
P1S3	.7534	.8690	1.0000		
P1S4	.3610	.4008	.3638	1.0000	
P1S5	.8081	.8343	.8945	.3561	1.0000

N of Cases = 77.0

Reliability Coefficients 5 items
Alpha = .8966 Standardized item alpha = .8996

Correlation Matrix

	P2S1	P2S2	P2S3	P2S4	P2S5
P2S1	1.0000				
P2S2	.9113	1.0000			
P2S3	.8723	.8861	1.0000		
P2S4	.4240	.4537	.3872	1.0000	
P2S5	.7981	.7810	.7737	.4947	1.0000

N of Cases = 76.0
Reliability Coefficients 5 items

Alpha = .9100 Standardized item alpha = .9133

Figure 4.1 (ctd.)

Correlation Matrix

	P3S1	P3S2	P3S3	P3S4	P3S5
P3S1	1.0000				
P3S2	.8017	1.0000			
P3S3	.8082	.8155	1.0000		
P3S4	.3297	.3232	.2670	1.0000	
P3S5	.8072	.8063	.7925	.3331	1.0000

N of Cases = 75.0

Reliability Coefficients 5 items

Alpha = .8832 Standardized item alpha = .8860

Next, to understand whether the scale measures a single concept (i.e. the assumed satisfaction construct) or multiple concepts, we performed a factor analysis. The results of this analysis are shown in Figure 4.2. As can be seen in the figure, a single factor explains most (73.5%) of the variance for these five variables and the scree plot supports this observation. This leads us to conclude that the scores for the items in this scale can be combined into a single "satisfaction score" and hence providing us with a surrogate for the "satisfaction" variable in our research model. What is left to be decided then is the weight distribution of these variables, i.e. how much of the satisfaction each variable actually measures. The component score matrix of Figure 4.2 suggests that the weights of the variables in explaining satisfaction are close enough to each other. For practical purposes, we consider these weights equal. With this observation, we proceed with our transformation by simply averaging the score of each individual on the five items and calculating a single "satisfaction score" as the third dependent variable of the model.

Figure 4.2

Factor Analysis Results

KMO and Bartlett's Test

Kaiser-Meyer-Olkin Measure of Sampling Adequacy.		.843
Bartlett's Test of Sphericity	Approx. Chi-Square	326.017
	df	10
	Sig.	.000

Communalities

	Initial	Extraction
P1S1	1.000	.787
P1S2	1.000	.867
P1S3	1.000	.876
P1S4	1.000	.262
P1S5	1.000	.883

Extraction Method: Principal Component Analysis.

Total Variance Explained

	Initial Eigenvalues			Extraction Sums of Squared Loadings		
Component	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	3.675	73.502	73.502	3.675	73.502	73.502
2	.797	15.943	89.445			
3	.272	5.440	94.886			
4	.166	3.327	98.212			
5	.938E-02	1.788	100.000			

Extraction Method: Principal Component Analysis.

Scree Plot

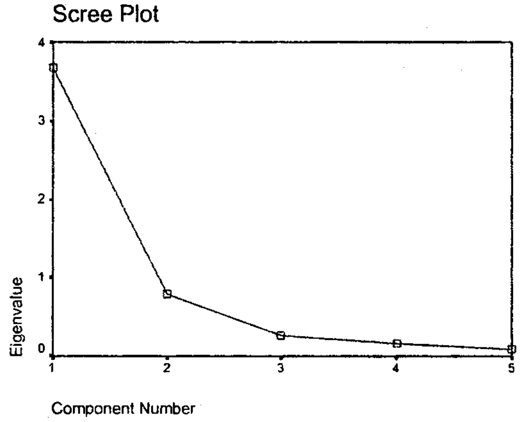


Figure 4.2 (ctd.)

Component Matrix[a]

	Component
	1
P1S1	.887
P1S2	.931
P1S3	.936
P1S4	.512
P1S5	.940

Extraction Method: Principal Component Analysis.

a. 1 components extracted.

Rotated Component Matrix[a]

a. Only one component was extracted. The solution cannot be rotated.

Component Score Coefficient Matrix

	Component
	1
P1S1	.241
P1S2	.253
P1S3	.255
P1S4	.139
P1S5	.256

Extraction Method: Principal Component Analysis.
Rotation Method: Varimax with Kaiser Normalization.

4.1.2 The Experimental Design Revisited

As explained before in Section 3.3.4, the experimental design that was used to collect data for this study is very similar to a repeated measures design, and the collected data can be analyzed as if they were collected through a repeated measures design if we can make the assumption that the order that the subjects were exposed to different presentation methods, (presentation method is the main effect that we are trying to test),has no significant effect on their performance and satisfaction. Accordingly, the next step in the data analysis process is to test the validity of this assumption. We start this step by revisiting our experimental design as done in Table 4.1, and by naming each of the nine cells in the diagram from 1 to 9 as illustrated.

Three factors differentiate these nine cells from each other: the presentation method, task, and phase. We are interested in the effect of the presentation method. Hence, to isolate the effect of the task and phase combination from the effect of the presentation method, we performed a test to compare the levels of the three dependent variables of our conceptual model between cells 1, 5, and 9, between cells 2, 6, and 7, and between cells 3, 4, and 8 separately. This way, we only compare the effect of the task and phase combination since the "method" in each of these comparisons is constant and hence controlled for. This comparison requires three different tests where the independent variable is the phase-task combination and the control variables are the "age", "sex", "native language", "cognitive style", and "Web experience" of the subjects. Accordingly, we performed a separate Multivariate Analysis of Variance (MANOVA) for each different presentation method. The two most common statistics used for significance testing in MANOVA are *Wilks' lambda* and *Roy's greatest characteristic* *root (gcr)*. According to Hair et al. (1995), "the measure to use is the one most immune to violations of the assumptions underlying MANOVA and yet maintains the greatest power. There is agreement that either Wilks' lambda or Pillai's criterion best meets these needs, although evidence suggests that Pillai's criterion is more robust if sample size decreases, unequal cell sizes appear, or homogeneity of covariances is violated." Accordingly, we test the significance of these two statistics in the rest of our analyses in this chapter.

Table 4.1

The Repeated Measures Design

Phase	**Task/Support**		
	Group 1	**Group 2**	**Group 3**
Phase 1	**1** Task A No visualization	**2** Task A Full zoom	**3** Task A Fisheye zoom
Phase 2	**4** Task B Fisheye zoom	**5** Task B No visualization	**6** Task B Full zoom
Phase 3	**7** Task C Full zoom	**8** Task C Fisheye zoom	**9** Task C No visualization

Figure 4.3.a

Preliminary Analyses for the Comparison of the Groups with "No visualization"

Descriptive Statistics				
	GROUP	**Mean**	**Std. Deviation**	**N**
Score	**1**	1.5769	.9868	26
	2	1.0417	.6903	24
	3	1.7143	1.0556	21
	Total	1.4366	.9522	71
Time	**1**	509.7692	157.3374	26
	2	559.7500	167.5824	24
	3	566.6190	169.2186	21
	Total	543.4789	164.0839	71
Satisfaction	**1**	3.6538	1.4561	26
	2	3.8667	1.5205	24
	3	3.8857	1.6378	21
	Total	3.7944	1.5149	71

Box's Test of Equality of Covariance [a]

Box's	9.50
F	.74
df1	12
df2	20810.11
Sig.	.71

Tests the null hypothesis that the observed matrices of the dependent variables are equal

Figures 4.3, 4.4, and 4.5 display the results of MANOVA for the non-visual, full zoom, and fisheye zoom techniques respectively.

Figure 4.3.(ctd.)

Comparison of the Groups with "No visualization"

Multivariate Tests(d)							
Effect		**Value**	**F**	**Hypothesis df**	**Error df**	**Sig.**	**Observed Power(a)**
Intercept	**Pillai's Trace**	.323	9.705(b)	3.000	61.00	.000	.996
	Wilks' Lambda	.677	9.705(b)	3.000	61.00	.000	.996
GROUP	**Pillai's Trace**	.133	1.478	6.000	124.00	***.191***	.557
	Wilks' Lambda	.870	1.465(b)	6.000	122.00	***.196***	.552
Cognitive Style	**Pillai's Trace**	.048	1.027(b)	3.000	61.00	.387	.266
	Wilks' Lambda	.952	1.027(b)	3.000	61.00	.387	.266
AGE	**Pillai's Trace**	.011	.226(b)	3.000	61.00	.878	.090
	Wilks' Lambda	.989	.226(b)	3.000	61.00	.878	.090
SEX	**Pillai's Trace**	.047	.994(b)	3.000	61.00	.402	.258
	Wilks' Lambda	.953	.994(b)	3.000	61.00	.402	.258
LANGUAGE	**Pillai's Trace**	.031	.659(b)	3.000	61.00	.581	.181
	Wilks' Lambda	.969	.659(b)	3.000	61.00	.581	.181
Web Experience	**Pillai's Trace**	.074	1.623(b)	3.000	61.00	.193	.406
	Wilks' Lambda	.926	1.623(b)	3.000	61.00	.193	.406
a Computed using alpha = .05							
d Design: Intercept+GROUP+Cogntive Style+AGE+SEX+LANGUAGE+Web Experience							

Figure 4.3.a shows descriptive statistics, and the results of Box's M Test, which point to the appropriateness of MANOVA for the purpose of comparing different groups and hence the different phase-task combinations.

Figure 4.4

Comparison of the Groups with "Full Zoom"

Descriptive Statistics

	GROUP	Mean	Std. Deviation	N
Score	**1**	1.3913	.9881	23
	2	1.3750	.9696	24
	3	1.2174	.9023	23
	Total	1.3286	.9437	70
Time	**1**	487.5652	162.8384	23
	2	510.2917	151.7221	24
	3	458.2609	116.4467	23
	Total	485.7286	144.6153	70
Satisfaction	**1**	3.7913	1.4123	23
	2	2.4750	1.1807	24
	3	3.0000	1.7633	23
	Total	3.0800	1.5460	70

Box's Test of Equality of Covariance [a]

Box's	9.883
F	.770
df1	12
df2	21672.25
Sig.	.682

Tests the null hypothesis that the observed matrices of the dependent variables are equal across

Figure 4.4 (ctd.)

Multivariate Tests(d)							
Effect		**Value**	**F**	**Hypothesis df**	**Error df**	**Sig.**	**Observed Power(a)**
Intercept	**Pillai's Trace**	.241	6.349(b)	3.000	60.00	.001	.957
	Wilks' Lambda	.759	6.349(b)	3.000	60.00	.001	.957
GROUP	**Pillai's Trace**	.137	1.497	6.000	122.00	***.185***	.563
	Wilks' Lambda	.866	1.497(b)	6.000	120.00	***.185***	.562
Cognitive Style	**Pillai's Trace**	.139	3.238(b)	3.000	60.00	.028	.717
	Wilks' Lambda	.861	3.238(b)	3.000	60.00	.028	.717
AGE	**Pillai's Trace**	.032	.658(b)	3.000	60.00	.581	.180
	Wilks' Lambda	.968	.658(b)	3.000	60.00	.581	.180
SEX	**Pillai's Trace**	.013	.262(b)	3.000	60.00	.853	.097
	Wilks' Lambda	.987	.262(b)	3.000	60.00	.853	.097
LANGUAGE	**Pillai's Trace**	.131	3.006(b)	3.000	60.00	.037	.681
	Wilks' Lambda	.869	3.006(b)	3.000	60.00	.037	.681
Web Experience	**Pillai's Trace**	.095	2.106(b)	3.000	60.00	.109	.512
	Wilks' Lambda	.905	2.106(b)	3.000	60.00	.109	.512
a Computed using alpha = .05							
d Design: Intercept+GROUP+Cognitive Style+AGE+SEX+LANGUAGE+Web Experience							

Figure 4.5

Comparison of the Groups with "Fisheye Zoom"

Descriptive Statistics				
	GROUP	Mean	Std. Deviation	N
Score	1	1.1538	.8339	26
	2	1.4800	.9183	25
	3	1.0400	.9345	25
	Total	1.2237	.9033	76
Time	1	451.0000	162.0351	26
	2	460.8800	139.2220	25
	3	443.6400	138.1490	25
	Total	451.8289	145.2977	76
Satisfaction	1	3.2846	1.3684	26
	2	3.0080	1.3260	25
	3	3.4880	1.6259	25
	Total	3.2605	1.4393	76

Box's Test of Equality of Covariance [a]

Box's	8.125
F	.637
df1	12
df2	25742.79
Sig.	.813

Tests the null hypothesis that the observed
matrices of the dependent variables are equal across

Figure 4.5 (ctd.)

Multivariate Tests(d)							
Effect		**Value**	**F**	**Hypothesis df**	**Error df**	**Sig.**	**Observed Power(a)**
Intercept	**Pillai's Trace**	.134	3.395(b)	3.000	66.00	.023	.742
	Wilks' Lambda	.866	3.395(b)	3.000	66.00	.023	.742
Cognitive Style	**Pillai's Trace**	.090	2.180(b)	3.000	66.00	.099	.530
	Wilks' Lambda	.910	2.180(b)	3.000	66.00	.099	.530
AGE	**Pillai's Trace**	.026	.581(b)	3.000	66.00	.629	.164
	Wilks' Lambda	.974	.581(b)	3.000	66.00	.629	.164
SEX	**Pillai's Trace**	.049	1.124(b)	3.000	66.00	.346	.290
	Wilks' Lambda	.951	1.124(b)	3.000	66.00	.346	.290
LANGUAGE	**Pillai's Trace**	.165	4.342(b)	3.000	66.00	.007	.849
	Wilks' Lambda	.835	4.342(b)	3.000	66.00	.007	.849
Web Experience	**Pillai's Trace**	.081	1.949(b)	3.000	66.00	.130	.481
	Wilks' Lambda	.919	1.949(b)	3.000	66.00	.130	.481
GROUP	**Pillai's Trace**	.032	.367	6.000	134.00	***.899***	.152
	Wilks' Lambda	.968	.363(b)	6.000	132.00	***.901***	.151
a Computed using alpha = .05							
d Design: Intercept+GROUP+Cognitive Style+AGE+SEX+LANGUAGE+Web Experience							

Figure 4.3.b shows the results of the test that is of main interest at this point. As seen in the multivariate tests of differences between groups (and hence the different phase-task combinations), there is no significant difference at the 0.05 level of Type I error. For this reason, the univariate tests are skipped with the conclusion that the phase-task combination has no effect when the subjects are given the text based (i.e. the non-visual) system.

Figures 4.4 and 4.5 are identical to Figure 4.3 in terms of the purpose and meaning of the performed tests, the only difference being that they are performed for the other two methods of presentation, i.e. full zoom visualization, and fisheye view visualization. The interpretation of the results in these figures is very similar to the previous one, thus it is not repeated here. The conclusion that is drawn from the three figures altogether is that the phase-task combination has no significant effect in the performance (i.e. success) and satisfaction of the experimental subjects.

This conclusion may be indicative of either one of two facts. There is either no phase effect and no task effect therefore we can discard the order that the subjects received the methods, and treat this design as a repeated measures design; or there is a phase effect opposed by a task effect hence none of them is observable.

At this point, we can return to the assumption that we made about the tasks after their selection, and based on that assumption, conclude that there is no phase (and hence learning) effect since the tasks are virtually identical, and there is no task effect. Yet even without this assumption, it can be observed that there is no need to find out the individual effects of task and phase since they are not the objects of our main analysis, and as long as their combined effect is zero, we can proceed with testing the effect of the presentation method. Furthermore, an effort to separate the effect of one of these variables from that of the other would require that the same group of subjects be presented the same task using a different order of the presentation methods, and in this case, a learning effect would be inevitable, and this would hamper our analyses.

The conclusion that the order (phase-task combination) does not affect the dependent variables of the model simplifies our hypothesis testing effort to a considerable extent. As a result of this conclusion, the dependent variable measures from each subject can be treated as the repeated measures of the same variables with a "within subjects", i.e. repeating, factor of presentation method. This way, the group membership becomes irrelevant, and the whole sample can be considered as one group receiving different treatments at three different times and being measured on the same variables (the dependent variables). Accordingly, the test of hypotheses can be performed simultaneously by means of a Repeated Measures Multivariate Analysis of Variance (MANOVA) and multiple (paired) comparisons.

4.2 Analysis of the Revised Data

4.2.1 MANOVA Assumptions

To find out whether MANOVA should be preferred to three separate Repeated Measures Multivariate Analyses of Variance (ANOVA), we first test to see whether at least two of the three dependent variables ("number of correct answers", "time to complete the question set", and "satisfaction") variables are correlated. For this purpose, the correlations between the dependent variables are analyzed for each different treatment (presentation method) as displayed in Figure 4.6. Figure 4.6 suggests that there are a number of significant correlations between the dependent variable measures therefore a MANOVA can be justified.

Another assumption of MANOVA is that the dependent variable measures collected from the subjects in the three groups are normally distributed (multivariate normal). There is no test of multivariate normality hence this assumption is tested through separate univariate normality tests. Figures 4.7 through 4.9 display Normal P-P plots where the distribution of the dependent variables for each different presentation method in comparison to an ideal normal distribution (the straight line in each graph) can be seen. These plots suggest that the distributions of the dependent variables are sufficiently close to normal hence we can assume the (univariate) normality condition is satisfied.

In the light of these analyses, it is appropriate to analyze the data and test the hypotheses with MANOVA.

Figure 4.6

Correlation between the Dependent Variables

Correlations with "No Visualization"				
		Score	**Time**	**Satisfaction**
Score	**Pearson Correlation**	1.000	.107	-.039
	Sig. (2-tailed)	.	.369	.739
	N	77	72	77
Time	**Pearson Correlation**	.107	1.000	.162
	Sig. (2-tailed)	.369	.	.173
	N	72	72	72
Satisfaction	**Pearson Correlation**	-.039	.162	1.000
	Sig. (2-tailed)	.739	.173	.
	N	77	72	77

Figure 4.6. (ctd.)

Correlations with "Full Zoom Visualization"				
		Score	**Time**	**Satisfaction**
Score	**Pearson Correlation**	1.000	.311(**)	.223
	Sig. (2-tailed)	.	.007	.056
	N	77	73	74
Time	**Pearson Correlation**	.311(**)	1.000	.244(*)
	Sig. (2-tailed)	.007	.	.040
	N	73	73	71
Satisfaction	**Pearson Correlation**	.223	.244(*)	1.000
	Sig. (2-tailed)	.056	.040	.
	N	74	71	74
** Correlation is significant at the 0.01 level (2-tailed).				
* Correlation is significant at the 0.05 level (2-tailed).				

Correlations with "Fisheye Zoom Visualization"				
		Score	**Time**	**Satisfaction**
Score	**Pearson Correlation**	1.000	.290(*)	-.043
	Sig. (2-tailed)	.	.010	.707
	N	78	78	77
Time	**Pearson Correlation**	.290(*)	1.000	.041
	Sig. (2-tailed)	.010	.	.726
	N	78	78	77
Satisfaction	**Pearson Correlation**	-.043	.041	1.000
	Sig. (2-tailed)	.707	.726	.
	N	77	77	77
* Correlation is significant at the 0.05 level (2-tailed).				

Figure 4.7

Plots for "No Visualization"

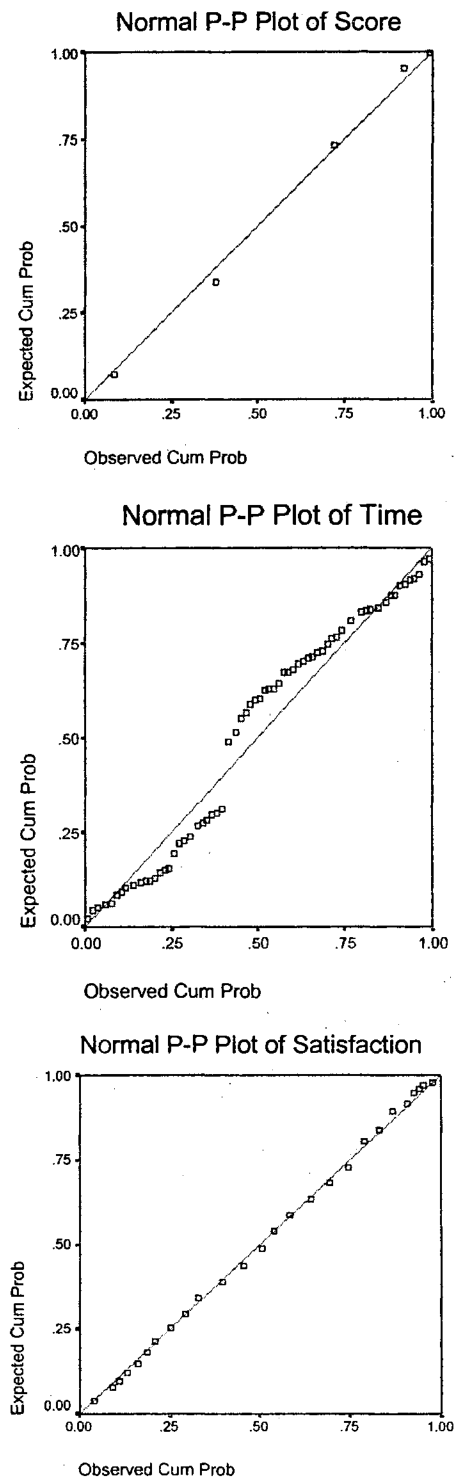


Figure 4.8

Plots for the "Full Zoom" Visualization

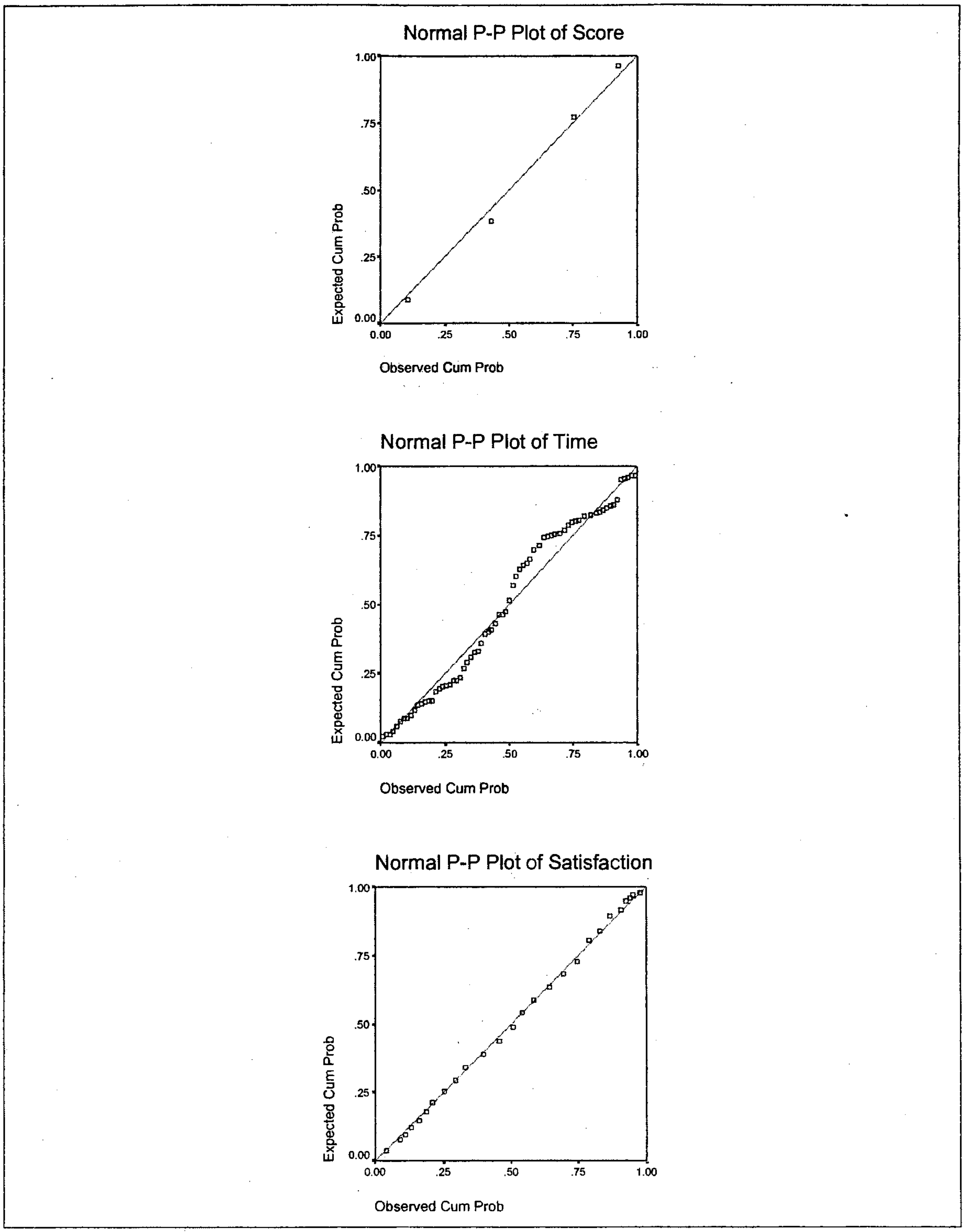
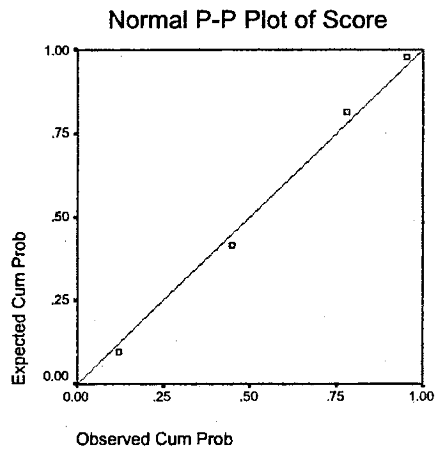
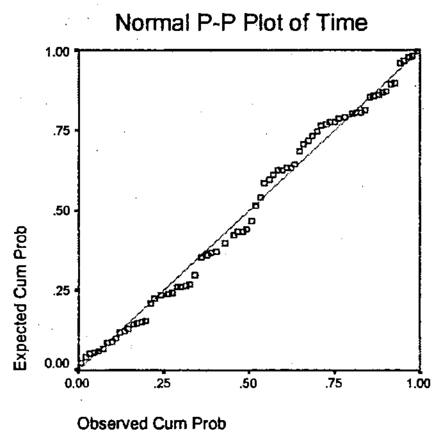
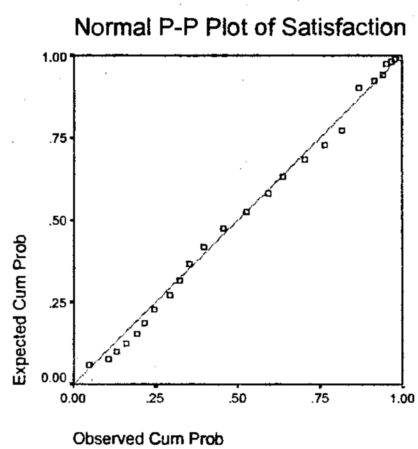


Figure 4.9

Plots for the "Fisheye Zoom" Visualization







4.2.2 Hypotheses Testing

For convenience, the hypotheses of our study are restated in Figure 4.10.

Figure 4.10

Hypotheses of the Study

H1a: Existence of visualization increases <u>the number of correct answers.</u>
H1b: Existence of visualization reduces <u>the time to complete the task of answering the questions.</u>
H1c: The visual systems result in <u>higher user satisfaction</u> than the text-based system does.
H2a: Use of the fisheye zoom instead of the full zoom increases <u>the number of correct answers.</u>
H2b: Use of the fisheye zoom instead of the full zoom reduces <u>the time to complete the task of answering the questions.</u>
H2c: The fisheye zoom system results in <u>higher user satisfaction</u> than the full zoom system does.

To test these hypotheses, we conducted a MANOVA with a within subjects factor of presentation "method", and between subjects factors of "sex", "native language", "age", "Web experience", and "cognitive style" (field independence). We included multiple (paired) contrasts in the analysis in order to compare the "visual" systems to the "non-visual", and the "fisheye zoom" system to the "full zoom system" without the need for separate tests.

Figure 4.11

The factors of the model

Within-Subjects Factors

Measure	METHOD	Dependent Variable
SCORE	1	SCOREFUL
	2	SCOREFIS
	3	SCORENO
TIME	1	TIMEFULL
	2	TIMEFISH
	3	TIMENO
SAT	1	SATFULL
	2	SATFISH
	3	SATNO

Between-Subjects

		N
AGE	1	1
	2	44
	3	18
	4	15
SEX	1	37
	2	41
LANGUAGE	1	47
	2	31
Web Experience	1	2
	2	3
	3	12
	4	61
Cognitive Style	1.00	21
	2.00	12
	3.00	19
	4.00	26

The original data set contained missing variable values for a total of 12 cases. We chose to treat these cases by a method that is very commonly used in data analysis. Given that there were only one or a few missing values for each of these cases, we chose to replace these missing values by the average of the variable (the values of the variables from the other cases) instead of eliminating these cases all together. This way the original sample size is conserved (this was a convenience sample, and was limited in size). We continue our analyses with this larger sample, (of size 78) hence obtaining higher power and improved generalizability (i.e. external validity).

The factors of the model and the descriptive statistics based on the "method" factor are displayed in Figures 4.11, 4.12, 4.13, and 4.14 respectively.

Figure 4.12

Descriptive Statistics for "Score"

	Mean	**Std. Deviation**	**N**
Full Zoom	***1.2885***	.9518	78
Fisheye Zoom	***1.1923***	.9125	78
No visualization	***1.4026***	.9570	78

As seen in Figure 4.12, on the average, the subjects had the highest scores, i.e. found the largest number correct answers, without visualization. Following that was the score obtained by the full zoom visualization method, and then that with the fisheye view method. This observation of the score variable is in direct contrast with our hypotheses. While the observed result may be due to the effect of the presentation method, it may also be caused by other factors, or may be pure selection error.

Figure 4.13

Descriptive Statistics for "Time"

	Mean	**Std. Deviation**	**N**
Full Zoom	***481.2051***	141.0319	78
Fisheye Zoom	***453.6667***	143.9846	78
No Visualization	***543.8974***	156.8787	78

Figure 4.14

Descriptive Statistics for "Satisfaction"

	Mean	**Std. Deviation**	N
Full Zoom	***3.1821***	1.5854	78
Fisheye Zoom	***3.3013***	1.4642	78
No Visualization	***3.8462***	1.5675	78

Figure 4.15

Multivariate Tests

Within Subjects Effect		Value	F	Hypothesis df	Error df	Sig.
METHOD	Pillai's Trace	.107	2.459	6.000	262.000	***.025***
	Wilks' Lambda	.894	2.505(b)	6.000	260.000	***.022***
METHOD * AGE	Pillai's Trace	.105	.799	18.000	396.000	.703
	Wilks' Lambda	.897	.797	18.000	368.181	.704
METHOD * SEX	Pillai's Trace	.033	.731	6.000	262.000	.625
	Wilks' Lambda	.967	.727(b)	6.000	260.000	.628
METHOD * LANGUAGE	Pillai's Trace	.029	.648	6.000	262.000	.692
	Wilks' Lambda	.971	.644(b)	6.000	260.000	.695
METHOD * Web Experience	Pillai's Trace	.182	1.418	18.000	396.000	.119
	Wilks' Lambda	.826	1.430	18.000	368.181	.114
METHOD * Cognitive Style	Pillai's Trace	.178	1.389	18.000	396.000	.132
	Wilks' Lambda	.829	1.398	18.000	368.181	.129

Figures 4.13 and 4.14 show that the relationship between the presentation method and the time to complete the tasks is in the hypothesized direction ($\boldsymbol{time}_{\textit{fisheye}} < \boldsymbol{time}_{\textit{full zoom}} < \boldsymbol{time}_{\textit{no visualization}}$), and that between the presentation method and satisfaction is partially in the hypothesized direction ($\boldsymbol{satisfaction}_{\textit{fisheye}} > \boldsymbol{satisfaction}_{\textit{full zoom}}$, and $\boldsymbol{satisfaction}_{\textit{no visualization}} > \boldsymbol{satisfaction}_{\textit{fisheye}}$). Yet as mentioned before, we do not yet have evidence as to the cause of these observed

outcomes. Subsequently, we explore the statistical significance of these results to find out the actual cause of the observed levels of the score, time, and satisfaction variables.

Figure 4.15 displays the significance testing of the main effect of our model, as well as that of the interaction of the main effect with the control variables at the multivariate level. It has been repeated a number of times before that the main effect of interest in this study is the presentation method, and that the role of the other variables is mainly in the form control variables. Accordingly, our hypotheses are formulated to test the effect of "presentation method", and for this reason we do not discuss the significance of the effect of the other independent variables in detail. Nevertheless, it may be interesting to note that the only control variable with a significant effect on the dependent variable set was "native language" ($p=0.001$).

Figure 4.15 shows that at the 0.05 level of α, the effect of "presentation method" is significant ($p=0.025$), and that of all interaction terms is not. This result simply implies that there is an overall difference between the success of the three different presentation methods, and that this difference is independent of the level of the control variables. However, we still cannot decide whether this means that there is a significant difference between the three different presentation methods in how they affect the individual outcomes, i.e. "score", "time", and "satisfaction". Being able to test our hypotheses depends on that decision for which we follow a stepwise approach, and look at the univariate tests for the "method" effect next, as displayed in Figure 4.16.

Figure 4.16

Univariate Tests

Source	Measure		Type III Sum of Squares	df	Mean Square	F	Sig.
METHOD	SCORE	Sphericity Assumed	.929	2	.464	.570	.567
		Greenhouse-Geisser	.929	1.796	.517	.570	.549
		Huynh-Feldt	.929	2.000	.464	.570	.567
		Lower-bound	.929	1.000	.929	.570	.453
	TIME	Sphericity Assumed	97222.259	2	48611.129	6.655	***.002***
		Greenhouse-Geisser	97222.259	1.977	49169.681	6.655	***.002***
		Huynh-Feldt	97222.259	2.000	48611.129	6.655	***.002***
		Lower-bound	97222.259	1.000	97222.259	6.655	***.012***
	SATISFACTION	Sphericity Assumed	1.500	2	.750	.639	.530
		Greenhouse-Geisser	1.500	1.708	.878	.639	.506
		Huynh-Feldt	1.500	2.000	.750	.639	.530
		Lower-bound	1.500	1.000	1.500	.639	.427

Figure 4.16 displays that the effect of the presentation method is significant for the "time" variable only. We have two hypotheses, i.e. H1b, and H2b about the effect of the presentation method on time. Following on the stepwise analysis approach, these two hypotheses are tested by means of the a priori defined paired comparisons, results of which are displayed in Figure 4.17. According to the comparison in Figure 4.17, the only significant effect of the method is on the time it took the subjects to complete the tasks using the visual systems as opposed to the text-based system (H1b), and the direction of

the effect is in the hypothesized direction as previously mentioned (see Figure 4.13). We continue the discussion on the results of the analyses in the next section.

Figure 4.17

Paired Comparisons

Source	Measure	METHOD	Type III Sum of Squares	df	Mean Square	F	Sig.
METHOD	SCORE	Level 2 vs. Level 1	.805	1	.805	.580	.449
		Level 3 vs. Previous	.790	1	.790	.562	.456
	TIME	Level 2 vs. Level 1	21950.01	1	21950.01	1.618	.208
		Level 3 vs. Previous	129370.87	1	129370.87	11.022	***.001***
	SATISFACTION	Level 2 vs. Level 1	1.409	1	1.409	.984	.325
		Level 3 vs. Previous	1.193	1	1.193	.487	.488

4.3 Post Hoc Analyses

The results of the previous section partially support our hypotheses since they point to the fact that visualization significantly increases the speed of the information seekers browsing the collection of Web search results. However, we take caution in interpreting this result since it may be indicative of the fact that while using the visual methods, the subjects quit searching for the answers and finished the tasks faster hence resulting in lower average scores. Although not significant, the results on the effect of "score" support this sort of an explanation as well. In order to rule out this alternative explanation, we performed our analyses again, this time with a subset of the overall sample that was formed by eliminating those subjects who did poorly (only one correct answer or none in all three phases) in finding the right answers to the experimental questions. The analyses using this restricted sample were performed following the same procedure as with the original sample.

Figure 4.18

Descriptive Statistics for the Modified Sample

Dependent Variable		**Mean**	**Std. Deviation**	**N**
Score	**Full Zoom**	1.4825	1.0175	57
	Fisheye Zoom	1.4386	.9067	57
	No Visualization	1.6737	.9477	57
Time	**Full Zoom**	507.2105	125.5025	57
	Fisheye Zoom	479.4035	141.4043	57
	No Visualization	580.1053	149.4160	57
Satisfaction	**Full Zoom**	3.3193	1.5431	57
	Fisheye Zoom	3.3421	1.4490	57
	No Visualization	3.8772	1.4652	57

Figure 4.19

Multivariate Tests for the Reduced Sample

Within Subjects Effect		Value	F	Hypothesis df	Error df	Sig.
METHOD	**Pillai's Trace**	.203	3.426	6.000	182.000	***.003***
	Wilks' Lambda	.799	3.563(b)	6.000	180.000	***.002***
METHOD * Cognitive Style	**Pillai's Trace**	.239	1.327	18.000	276.000	.170
	Wilks' Lambda	.770	1.371	18.000	255.044	.146
METHOD * AGE	**Pillai's Trace**	.165	1.337	12.000	276.000	.197
	Wilks' Lambda	.841	1.342	12.000	238.409	.196
METHOD * SEX	**Pillai's Trace**	.065	1.017	6.000	182.000	.416
	Wilks' Lambda	.936	1.009(b)	6.000	180.000	.421
METHOD * LANGUAGE	**Pillai's Trace**	.030	.459	6.000	182.000	.838
	Wilks' Lambda	.970	.455(b)	6.000	180.000	.841
METHOD * Web Experience	**Pillai's Trace**	.221	1.221	18.000	276.000	.243
	Wilks' Lambda	.791	1.223	18.000	255.044	.243

The descriptive statistics of Figure 4.18 agree with the descriptive statistics of the original sample, and the multivariate tests of Figure 4.19 show that there is a main "method" effect, and no significant interaction effects. As was done previously, the next step is to check the significance of the effect of method on the three dependent variables individually. For this purpose, the univariate tests of Figure 4.20 are performed.

Figure 4.20

Univariate Tests for the Reduced Sample

Source	Measure		Type III Sum of Squares	df	Mean Square	F	Sig.
METHOD	SCORE	Sphericity Assumed	2.785	2	1.392	1.387	.255
		Greenhouse-Geisser	2.785	1.777	1.567	1.387	.255
		Huynh-Feldt	2.785	2.000	1.392	1.387	.255
		Lower-bound	2.785	1.000	2.785	1.387	.245
	TIME	Sphericity Assumed	151604.494	2	75802.247	8.883	***.000***
		Greenhouse-Geisser	151604.494	1.971	76900.844	8.883	***.000***
		Huynh-Feldt	151604.494	2.000	75802.247	8.883	***.000***
		Lower-bound	151604.494	1.000	151604.494	8.883	***.005***
	SAT	Sphericity Assumed	.472	2	.236	.214	.808
		Greenhouse-Geisser	.472	1.614	.293	.214	.760
		Huynh-Feldt	.472	2.000	.236	.214	.808
		Lower-bound	.472	1.000	.472	.214	.646

As was the case with the original sample, the only outcome that is significantly affected by "method" at the 0.05 level is "time". Subsequently, as the final step in validating our previous conclusions as to the supporting of the research hypotheses, we look at the paired comparisons for the reduced sample as in Figure 4.21.

Figure 4.21

Paired Comparisons for the Reduced Sample

Source	Measure	METHOD	Type III Sum of Squares	df	Mean Square	F	Sig.
METHOD	SCORE	Level 2 vs. Level 1	4.459	1	4.459	2.455	.124
		Level 3 vs. Previous	.832	1	.832	.505	.481
	TIME	Level 2 vs. Level 1	71515.541	1	71515.541	4.470	***.040***
		Level 3 vs. Previous	173770.086	1	173770.086	12.776	***.001***
	SAT	Level 2 vs. Level 1	.155	1	.155	.123	.727
		Level 3 vs. Previous	.592	1	.592	.250	.620

The results displayed in Figure 4.21 are interesting in the sense that they do not only sustain our previous convictions as to the level of support that the statistical analyses provide for our hypotheses, but they actually support a hypothesis (H2b) that the analysis with the original data set had not supported. Since this modified sample is more carefully selected, we will have more faith in the results thereof.

In the light of these observations, Table 4.2 displays a summary of the results of our data analyses. As can be seen in this summary, our empirical findings give partial support to our research hypotheses suggesting that the use of visualization in the presentation of Web search results significantly increases the search efficiency, i.e. the speed of finding specific pieces of information among the results (H1b is supported by the analyses with both samples), and a fisheye zoom visual system results in a more efficient search than a full zoom system (H2b is supported by the reduced sample).

Table 4.2

Summary of the Data Analyses Results

Hypothesis	Original Test Sample	Reduced Test Sample
H1a	No support	No support
H1b	**Support in hypothesized direction**	**Support in hypothesized direction**
H1c	No support	No support
H2a	No support	No support
H2b	No support	**Support in hypothesized direction**
H2c	No support	No support

In their 1992 paper, Todd and Benbasat discuss the validity of the general assumption, which proclaims that if decision makers are provided with expanded processing capabilities, they will use them to analyze problems in more depth and, as a result, make better decisions. According to the authors, a possible explanation for why this proposition has not been supported empirically can be found in behavioral decision making theories. "The literature on behavioral decision making indicates that the conservation of effort may be more important than increased decision quality in some cases. If this is so, then the use of a decision aid may result in effort savings, but not improved decision performance" (Todd and Benbasat 1992). According to the empirical evaluation in the mentioned study, the subjects behaved as if effort minimization was an important consideration, and did not produce higher quality when they used better decision aids.

Although the experimental tasks used in our experiments are not traditional decision-making tasks, we believe a very similar explanation applies to why the use of our visual systems increased efficiency (reduced effort), but did not significantly affect effectiveness. Based on such an explanation, it can be said that our hypotheses were based on a too optimistic assumption. In that case, a possibly better, i.e. more realistic way of formulating our research questions into research propositions would be as follows:

<u>Proposition 1a</u>: *In case the subjects spend the same amount of effort, i.e. time for working on the search-related tasks, the visual systems will result in higher effectiveness than the text-based system does.*

<u>Proposition 1b:</u> The subjects using the visual systems will be as effective as the ones using the text-based system by spending less time on the search-related tasks.

<u>Proposition 2a</u>: In case the subjects spend the same amount of effort, i.e. time for working on the search-related tasks, the fisheye zoom system will result in higher effectiveness than the full zoom system does.

<u>Proposition 1b:</u> The subjects using the fisheye zoom visual system will be as effective as the ones using the full zoom system by spending less time on the search-related tasks.

It is our conviction that the above propositions will find a substantial amount of empirical support if used in future studies. The next chapter continues the discussion on our empirical results along with the other studies of the dissertation in a broader context.

5. CONCLUSIONS AND DISCUSSION

5.1 Brief Summary

In the previous chapters, we reported on our study that addresses an important Web-related information overload problem: users of commercial Web search engines often get overloaded by the long textual list of results. The causes of the problem are numerous, and are being studied by various groups of researchers, and yet there is not an overall comprehensive solution because of the complexity of the problem. Our approach to the alleviation of the problem was to create an extra layer between the information seeker and the search engine. This function of this layer is to hierarchically cluster the search results and present a visual representation of these clusters while allowing the users of the system to further explore the details of the collection by alternative zooming techniques. We developed a model of this process as depicted in Figure 2.20, and implemented two prototype systems based on this model. The only difference between these two systems is the use of different zooming methods for users to explore the details of the document collection. We proposed that the combined use of clustering and visualization would lead to higher success than the traditional textual presentation of search results, and that the visual system that uses a "fisheye zoom" method would lead to higher success than the system that uses the "full zoom" method. We empirically tested the hypotheses that are the operational form of these propositions and found partial support such that the use of visualization in the presentation of Web search results significantly increased the speed of finding specific pieces of information among the results. Furthermore, a fisheye zoom visual system resulted in a more efficient search than a full zoom system since the users of the former system found answers to the experimental questions faster than the users of the latter.

The rest of this chapter starts with a discussion on the contributions of the study. Afterwards, the results of the prototype development effort and the empirical studies are elaborated identifying the weaknesses and possible directions for future research.

5.2 Contributions of the Study

As Roussinov (1999) quotes from Gey et al. (1999):

> There is a growing opinion in the Information Retrieval community that a key to improving information access systems is to focus attention on the human-computer interface.

From this point of view, research that aims to enhance the quality of IR system interface is valuable, especially in today's world, where information is used by people with various backgrounds and varying degrees of computer experience for a myriad of purposes ranging from online entertainment to online shopping. The following comment from one of our experimental subjects provides anecdotal evidence on this:

> Excellent job! It is time saving and easy. (*Refers to the fisheye zoom system*)

In that sense, we believe that the results of our research effort have the promise of significant improvements in the interface design of practical IR systems, especially considering that there is an important tendency towards user-friendly desktop applications, and that visual aids such as windows, frames, icons, and images are inevitable elements of user-friendly interface design.

Our review of the previous work on visual information retrieval systems showed that there were a few conceptual models of the information visualization process. In that sense, Shneiderman's "visual information seeking mantra" (1996) is very insightful since it is a fairly complete model. However, this model is very broad, and does not concern itself with any specific information-seeking paradigm. To that effect, we believe that our overall analysis of clustering-based visual information search (Figure 1.1) makes a good contribution by providing a logical sequence of tasks that need to be performed by a visual system such as the one that we developed and reported in this study. The simple model of Figure 1.1 can be useful for future visual information search system designers, at least as a good starting point.

The approach that we used for the visualization component of the prototype was adopted from Johnson and Shneiderman (1991). To our knowledge, the visualization of Web search results is a new domain for the use of their treemap algorithm hence the modification of this algorithm for our problem domain is a specific contribution of this study. The original algorithm does not include a zooming feature since it requires all details of an information space to be presented in a single level. The heuristic rules that we used for creating an overview of the overall information space, and the zooming facilities, especially the fisheye technique as described in Chapter 3 are original. We consider this set of extra features as something that the community of interface designers may use for enhanced designs. These design features do not have to be limited to information search engine interfaces or Web-based information retrieval systems by any means, meaning that their applicability may have much further reaches than what we have elaborated within this research.

Testing of the usability of systems by means of empirical studies is essential for the IS field. Much research has been done on the success or appropriateness of different information presentation systems, yet with few consistent and insightful results due to the lack of well-founded theory and conceptual models (Vessey 1991). In this research, we tried to base our empirical studies on a sound conceptual model. However, to our knowledge a complete model on the causal relationships of interest to us did not exist. Subsequently, we combined the theoretical models that were developed for different parts of the causal system that we have been studying. This led to the conceptual model of Figure 1.2. We believe that this model has the capability of explaining system success, and can be used as a base model for empirical research in the field.

As any conceptual model, the model of Figure 1.2 tries to show relationships between constructs, i.e. latent variables. The success of empirical studies based on such a model depends on how well the latent variables are operationalized. Our specific choices for the operationalization of the constructs in our conceptual model led to the research model of Figure 3.14. We followed the same approach as the one that we used for constructing our conceptual model for developing the operational model in Figure 3.14. This approach involved reviewing the literature and finding a different study for appropriate operational variable(s) for each different construct in our model. It is our belief that the operational variables developed as described are good surrogates for the constructs, and are easy to measure and therefore can be adopted by future empirical studies.

For testing our hypotheses, we conducted a controlled experiment, which is known as a method with high internal validity. Yet, a common concern involved in controlled experiments is the trade-off between higher control and a large sample size. The experimental design of Table 3.8 provided us with the control that would lead to such high internal validity, while suppressing the disadvantage caused by the small sample size due to the repeated nature of the design. This resulted in the extra advantage of high power and enhanced the generalizability of our results. We believe that our design has substantial promise for future experiments especially in the lack of a large sample, and hence an important contribution of this dissertation.

The empirical part of our study did not produce results to fully support our hypotheses. However, as explained in the last section of Chapter 4, our hypotheses may be based on an overly optimistic assumption. Accordingly, the lack of support for hypotheses H1a and H2a (improved system effectiveness) may be understable especially considering that our hypotheses on improved system efficiency (H2a and H2b) were empirically supported. In any case, this research made an important contribution by providing empirical results for answering important research questions

5.3 Discussion and Directions for Future Research

5.3.1 The Prototype System

The building of the prototype system involved many decisions on the selection or implementation of the system components. These decisions were determinants of the quality of the prototype system, and hence are worth discussing for identifying the weaknesses of the system and possible improvement efforts.

The AltaVista search engine was used as the back end of our systems to facilitate the actual search process. AltaVista is one of the hundreds of commercial search engines available today. Our choice of AltaVista instead of the other possible alternatives was due to the fact that it is one of the more powerful engines in terms of the indexed database, and flexibility in query formulation. Furthermore, the format of the "AltaVista results" page is relatively easy to parse for further processing.

Despite these reasons, our selection of the search engine was still not the result of a comprehensive comparison, but rather, was somewhat arbitrary. While this may not be considered a particular weakness of our system, it may still be worthwhile to consider and experiment with other search engine alternatives. Also, after the development project was initiated, it came to our realization that AltaVista changes the way it formats its "Search Results" page every so often. This entails a periodical modification of the code that parses this page to extract the title and URL information. This is another viable reason to consider alternative Web search engines that has comparable power to AltaVista while lacking the aforementioned disadvantageous features.

Although not in depth, our brief evaluation of other plausible Web search engine alternatives suggests that there is promise for the Northern Light search engine, which is especially interesting in that it provides an alternative clustering-based presentation of search results. Considering that our approach is based on the clustering of the search results, the clusters that are readily available from Northern Light can be used for visualization purposes. Furthermore, since the phase before clustering, namely, the retrieval and local saving of the documents is the most time consuming step of the process our prototype follows, this feature may be even more promising for our system, because it will add the much needed speed that the system needs to find more practical applicability.

The clustering feature in Northern Light is relatively recent, but still shows potential as discussed. On the other hand, the idea of presenting clusters of search results can easily be adopted by other commercial search engines. Northern Light or any other search engine that chooses to include the clustering feature as part of their system can adopt our visual design ideas, and implement a system with better integration of the individual components. Such a holistic approach would yield higher system efficiency. That being an implication for the future, three is still a need to do further research to address the issue of finding the right search engine that matches with our specific design guidelines for the rest of the prototype system.

Another point that is related to the speed of the system is the approach that was followed to represent each search result. Currently, the system follows the hyperlink from each result to retrieve the whole Web page, and saves the text portion of those pages locally as text documents. Since it is prone to network traffic and the performance of the communicating servers, the retrieval process is the most time-consuming component of the system. Our reason for following this approach for acquiring the search results was the conviction that the quality of the resulting clusters would be higher if we used the whole pages in clustering instead of using only the title or a summary for each result. We were aware of the speed degradation such an approach would lead to, yet our aim was to test the success of the visualization component rather than the overall system, and the practical use of the system has not become our major priority yet. Accordingly, our experiments were designed to control for the speed of the system by running the queries in advance instead of letting the users work with the whole system. While the speed of the system did not hurt the success of our experiments, we realize that system speed is critical for practical use, and that one way to improve the speed of our system is to find an alternative approach to using the whole pages for representing the search results.

Zamir and Etzioni (1999) found that using snippets instead of full documents for clustering did not lead to significantly lower cluster quality while improving the clustering speed considerably. Based on that finding, a speed improvement in our system can be achieved by just using the short snippet that the search engine provides in the "results" page instead of following the links and retrieving the whole documents.

In this study, we used a tool based on one of numerous possible techniques for clustering the documents. Some of the human subjects in the user studies complained about the quality of the clustering making comments such as "Some of the groups are too small." Our choice of the clustering tool was mainly based on convenience, i.e. availability and accessibility. Intelligent Miner is very fast in creating a cluster structure and needs no further processing in the way it presents its output. Nevertheless, we believe that some of the weaknesses of the system caused by the specific clustering scheme can be remedied by experimenting with different clustering techniques. This is an open issue that requires further investigation, and we will pay specific attention to this particular point in our future research.

Our research reported in this document involves experimentation with a specific (and original) implementation of the fisheye view technique. Consequently, the insights gained from the development and testing effort has only limited generalizability as to the success of the fisheye zooming, or in general terms, details embedded in context idea. It is our conviction that the general principles of fisheye views are fairly applicable in the specific domain of application we addressed, and further research should explore the application of different techniques for fisheye zooming of visual overviews for Web search results. A natural extension of this idea would be the exploration of different methods to create visual overviews of the document hierarchy, and compare them to the map representation that was implemented in this study. An example method of this kind is the hyperbolic tree that we had discussed in Chapter 2. Hyperbolic trees are particularly good fisheye view implementations although their applicability for large collections is debatable since the visual presentations that they produce are not space efficient. However, it would be very interesting to compare this method to the one we developed in this study in the future.

5.3.2 The Empirical Study

As discussed before, the conceptual basis of the empirical portion of our study is the model in Figure 1.2. The formation of this model was a result of our effort to integrate simpler conceptual models that were developed for different purposes. Due to the lack of much theoretical background, we chose to be simplistic in forming this model, for example, by avoiding to include explicit interactions between the independent variables. To that effect, there is need for more theory building research to possible enhance our conceptual model.

The research model of Figure 3.14 reflects a choice of one specific set of operational variables in the model of Figure 1.2. Although this operationalization was intuitively appealing, it is by no means a standard. A natural result of this observation is that it would be interesting to experiment with different operational variable alternatives, for example the quality instead of quantity of correct answers, and to compare the results obtained thereof to the ones reported in this study. Obviously, the freedom in the choice of these operational variables depends on the selection of the tasks. One idea that is worth exploring is to test the systems for complex tasks such as "finding relevant information for writing a report on a well-known philosopher", and evaluating the quality of the resulting information, and using this as a surrogate of interface effectiveness instead of the quantity-based score that we used in this dissertation.

In this study, we tested the combined effect of our system features, i.e. clustering, visualization, etc. Accordingly, we have no way of identifying the individual effects of the different techniques employed on the system success. To that effect, it would be interesting to examine the individual effects of at least two of the more important system features, namely clustering and visualization. As we mentioned before, we clearly see the need to experiment with different clustering and visualization algorithms in the future. Based on the observation made in this paragraph, we find it worthwhile to not only test the effect of one technique at a time, but to test different clustering-visualization pairs to find out the most effective combination in case there exists an interaction between the clustering technique and the visualization technique.

Although our empirical study is based on a theoretical conceptual model, and the groups were formed randomly, there were still some inevitable effects that hurt the internal as well as external validity of the study. One of the complaints that we heard about our experimental design was about the formulation of the search queries. For example, on subject wrote:

> The systems (*referring to the visual systems)* were good. The search could have been formulated better.

It was a straightforward observation that such complaints came from more experienced Web searchers, which suggests that there was an interaction between task and Web search experience that we had not previously envisioned and had no statistical way of detecting, since we had only one type of task. We did not give the subjects the freedom of writing their own queries, hence controlling for that aspect of the task.

Similarly, we did not consider the fact that our visual presentation systems are unfamiliar to the subjects, and that this may cause some variation in the dependent variables, especially in the satisfaction scores, that our model cannot explain. The following comment form an experimental subject is a good example of some of the behavioral variables we failed to address in our model:

> I thought it (*referring to the fisheye zoom system*) looked cool, but I wouldn't have the patience to use it

This was an obvious threat to the internal validity of our experimental design. Related to this, we also observed that the subjects who were not involved with the experiment had harder time learning and using the visual systems. This observation points to the existence of an interaction between the method and an omitted contextual variable "subject involvement". In later studies, either this variable should be integrated into the model, or its effect should be minimized, for example by providing more training on the use of the visual systems.

The following comment from another experimental subject further supports our views on the lack of stronger support for our research hypotheses

> It is sometimes hard to get people out of a paradigm. Anyway, when I went home and thought about it. I realized that I was faster in finding info using your project (*referring to the visual systems*).

In spite of the mentioned weaknesses in the empirical portion, we believe that our study was a promising first step in the application of a new paradigm in the design of user interface for information retrieval systems. We are convinced that we will find better support for our hypotheses when the above-mentioned weaknesses are addressed in future studies.

REFERENCES

1. Baldonado, M. Q. W. (1998), "Interfaces for Information Exploration: Seeing the Forest," Position paper for the CHI98 workshop on Innovation and Evaluation in Information Exploration Interfaces.

2. Bautigam, M. (1996), "Applying Information Navigation Techniques to Web Navgation," Ph.D. Thesis Proposal, University of California, Santa Cruz.

3. Bederson, B. B., Hollan, J. D., Perlin, K., Meyer, J., Bacon, D., & Furnas, G. W. (1996). "Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics," Journal of Visual Languages and Computing, Vol. 7, pp. 3-31.

4. Bederson, B. B., Hollan, J. D., Stewart, J., Rogers, D., Druin, A., Vick, D., Ring, L., Grose, E., & Forsythe, C. (1998). "A Zooming Web Browser," C. Forsythe, J. Ratner, & E. Grose (eds.), Human Factors and Web Development (Chap. 19, pp. 255-266). New Jersey: Lawrence Erlbaum.

5. Bloom, B. S. (1956*) Taxonomy of educational objectives: the classification of educational goals*, New York, Longmans.

6. Bright, J. (1996), "The Information Dilemma: Finding the Nuggets," Australian Accountant, Vol. 66, No. 3, pp. 30-31.

7. Chakrabarti, S., Dom, B., Raghavan, P., Rajagopalan, S., Gibson, D., and Kleinberg, J. (1996) "Automatic resource compilation of analyzing hyperlink

structure and associated text,"

http://decweb.ethz.ch/WWW7/1898/com1898.htm (28 Oct. 1998).

8. Chen H., Chung, Y. M., Ramsey, M., and Yang, C. C. (1998), "An intelligent personal spider (agent) for dynamic Internet/Intranet searching," Decision Support Systems, Vol. 23, pp. 41-58.

9. Chen H., Houston, A. L., Sewell, R. R., Shatz, B. R. (1997), "Internet Browsing and Searching: User Evaluations of Category Map and Concept Space Techniques," Journal of the American Society of Information Science, Vol. 49, No. 7, pp. 582-603.

10. Chen, M., Hearst, M., Hong, J., Lin, J. (1999), "Cha-Cha: A System for Organizing Intranet Search Results," Proceedings of the 2nd USENIX Symposium on Internet Technologies and SYSTEMS (USITS), October 11-14.

11. Chewning, E. C., Jr., and A. M. Harrell (1990), "The Effect of Information Load on Decision Makers' Cue Utilization Levels and Decision Quality in a Financial Distress Decision Task," *Accounting Organizations & Society,* Vol.15, No. 6, pp. 527-542.

12. Collaud, G., Dill, J., Jones C. V., Tan, P. (1995), "A Distorted-View Approach to Assisting Web Navigation," New Paradigms in Information Visualization and Manipulaton, in conjunction with CIKM '95, December 1, Baltimore, MD.

13. Cugini, J., Sebrecths, M. (1999), "Design of 3-D Visualization of Search Results: Evolution and Evaluation," contribution of the National Institute of Standards and Technology.

14. Cutting, D. R., Karger, D. R., Pedersen, J. O., Tukey, J. W. (1992), "Scatter/gather: A cluster-based approach to browsing large document collections," Proceedings of the Sixteenth Annual International ACM Conference on Research and Development in Information Retrieval (pp. 318-329).

15. Cutting, D. R., Karger, D. R., Pedersen, J. O., (1993), "Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collections," Proceedings of the Fifteenth Annual International ACM Conference on Research and Development in Information Retrieval (pp. 126-134).

16. De Bra P., Houben, G. J., Dignum, F. (1997), "Task-Based Information Filtering: Providing Information that is Right for the Job," http://wwwis.win.tue.nl/houben/pub/infwet97.html (1 Nov. 1999).

17. DeLone, W. H., McLean, E. R. (1992), "The Quest for The Dependent Variable," Information Systems Research, Vol. 3, No. 1, pp. 60-95.

18. Denning, P. (1982), "Electronic Junk," *Communications of the ACM*, Vol. 25, No. 2, pp. 163-165.

19. Fox, K.L., Frieder, O., Knepper, M.M., and Snowberg, E. J. (1999), "SENTINEL: A Multiple Engine Information Retrieval and Visualization

System," Journal of the American Society for Information Science, Vol. 50, No. 7, pp. 616-625.

20. French, J. C., Brown, D. E., Kim, Nam-Ho (1997), "A Classification Approach to Boolean Query Reformulation," Journal of the American Society for Information Science, Vol. 48, No. 8, pp. 694-706.

21. Furnas, G.W. (1986), "Generalized Fisheye views," Proceedings of the ACM SIG-CHI 86 Conference on Human Factors in Computing Systems, pp.16-23.

22. Gey, F., Hearst, N, & Tong, R. (1999). Call for paper: 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval. Communications of the ACM, 42(5).

23. Gordon, M., Pathak, P. (1999), "Finding information on the World Wide Web: the retrieval effectiveness of search engines," Information Processing and Management, Vol. 35, pp.141-180.

24. Haag, S., Cummings, M., Dawkins J. (2000), *Management Information Systems for the Information Age*, Irwin McGraw-Hill, St. Louis.

25. Hearst, M. (1995), "TileBars: visualization of Term Distribution Information in Full Text Information Access," Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (SIGCHI '95).

26. Hearst, M., Pedersen, P. (1996), "Reexamining the cluster hypothesis: scatter/gather on retrieval results," Proceedings of the 19th International ACM

SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96), pp. 76-84.

27. Hearst, M., Karadi, C. (1997), "Cat-a-Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy," Proceedings of the 20th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97), pp. 246-255.

28. Heir, J. F. Jr., Anderson, R.E., Tatham, R. L., Black, W., C. (1995), *Multivariate Data Analysis*, *Fourth Edition*, Prentice Hall, Englewood Cliffs, NJ.

29. Hetzler, B., Miller, N. (1998), "Four Critical Elements for Designing Information Exploration Systems," Position paper for the CHI98 workshop on Innovation and Evaluation in Information Exploration Interfaces, Berkeley, CA.

30. Hunt, R. E. and R.G. Newman (1997), "Medical Knowledge Overload: A Disturbing Trend for Physicians," *Health Care Management Review*, Vol. 22, No. 1, pp. 70-75.

31. Jarvenpaa, S. L., Dickson, G. W., DeSanctis, G. (1985), "Methodological Issues in Experimental IS Research: Experiences and Recommendations," MIS Quarterly, Vol. 9. No. 2, pp. 141-156.

32. Johnson, B., Shneiderman, B. (1991), "Treemaps: a space filling approach to the visualization of hierarchical information structures," Proceedings of the Second International IEEE Visualization Conference, pp. 284-291.

33. Johnson, M. (1997), "Battling Information Overload," *Bank Marketing*, Vol. 29, No. 11, pp. 26-27.

34. Johnsson, J. (1991), "Information Overload: CEOs Seek New Tools for Effective Decision Making," *Hospitals*, Vol. 65, No. 20, pp. 24-27.

35. Keller, K. L. and R. Staelin (1987), "Effects of Quality and Quantity of Information on Decision Effectiveness," *Journal of Consumer Research*, Vol. 14, No. 2, pp. 200-213.

36. Lagus, K., Kaski, S., Honkela, T., and Kohonen, T. (1996), "Browsing Digital Libraries with the Aid of Self-Organizing Maps," Proceedings of the Fifth International World Wide Web Conference, WWW5, pp. 71-79.

37. Lamping, J. and Rao, R. (1996), "Visualizing Large Trees Using The Hyperbolic Browser," Proceedings of ACM Conference on Human Computer Interaction.

38. Leung, K. Y. and Apperley, M. D. (1994), "A Review and Taxonomy of Distortion-Oriented Presentation Techniques," ACM Transactions on Computer-Human Interaction, Vol. 1, No. 2, pp. 126-160.

39. Lin, X. (1997), "Map Displays of Information Retrieval," Journal of the American Society of Information Science, Vol. 48, No. 1, pp. 40-54.

40. Marchionini, G. (1995), *Information Seeking in Electronic Environments,* Cambridge, UK: Cambridge University Press.

41. Malhotra, N. K. (1982), "Information Load and Consumer Decision Making," *Journal of Consumer Research*, Vol. 8, No. 4, pp. 419-430.

42. Mattlin, E. (1992), "Hub and Spoke Gets Rolling; Information Overload Can Mean Missed Opportunities," *Institutional Investor*, Vol. 26, No. 10, pp. 155-156.

43. Miller, N., Hetzler, B., Nakamura, G., Whitney, P. (1997), "The Need For Metrics In Visual Information Analysis," Proceedings of the Workshop on New Paradigms in Information Visualization and Manipulation in conjunction with the Sixth ACM International Conference on Information and Knowledge Management (CIKM '97) November 13-14, Las Vegas, Nevada, ACM Press

44. Mukherjea, S., Foley, J. D. (1995), "Showing the Context of Nodes in the World-Wide Web," Proceedings of ACM conference on Human Computer Interaction (CHI95).

45. Mukherjea, S., Foley, J. D., Hudson, S. (1995), "Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views," Proceedings of ACM conference on Human Computer Interaction (CHI95).

46. Nation, D. A., Plaisant, C., Marchionni, G., and Komlodi, A. (1997), "Visualizing web sites using a hierarchical table of contents browser: WebTOC," Proceedings of the third conference on Human Factors and the Web (www.uswest.com/web-conference).

47. Nigay, L., Vernier, F. (1998), "Navigational Interaction Techniques in the Search Results Space," Position paper for the CHI98 workshop on Innovation and Evaluation in Information Exploration Interfaces, Berkeley, CA.

48. O'Leary De (1998), "Expertise Knowledge Management," Computer, Vol. 31, No. 3, p.54.

49. Pirolli, P., Schank, P., Hearst, M. A., Diehl, C. (1996a), "Scatter/Gather Browsing Communicates the Topics Structure of a Very Large Text Collection", Proceedings ACM CHI96 Conference, April 13-18.

50. Pirolli, P., Pitkow, J., Rao, R. (1996b), "Silk from a Sow's Ear: Extracting Usable Structures from the Web," Proceedings ACM CHI96 Conference, April 13-18.

51. Roussinov, D., "Information Foraging Through Automatic Clustering and Summarization: A Self-Organizing Approach," Doctoral Dissertation, University of Arizona, August 1999.

52. Rudy, I. A. (1996), "A Critical Review of Research on Electronic Mail," *European Journal of Information Systems*, Vol. 4, No. 4, pp. 198-213.

53. Santhanam, R., Sein, M. K. (1994), "Improving End-user Proficiency: Effects of Conceptual Training and Nature of Interaction," Information Systems Research, Vol. 5, No. 4, pp. 378-399.

54. Savoy, J. (1997), "Ranking Schemes in Hybrid Boolean Systems: A New Approach," Journal of the American Society for Information Science, Vol. 48, No. 3, pp. 235-253.

55. Schafer, D., Zuo, Z., Greenberg, S., Bartram L., Dill J., Dubs, S., and Roseman, M. (1998), "Navigating Hierarchically Clustered Networks Through Fisheye and Full-Zoom Methods", ACM Transactions on Computer-Human Interaction, 13 (2), pp. 162-188.

56. Sebrechts, M. M., Cugini, J. V., Laskowski, S. J., Vasilakis, J., Miller, M. S. (1999), "Visualization of search results: a comparative evaluation of text, 2D, and 3D interfaces," Proceedings ACM SIGIR99 Conference, August 15-19, pp. 3- 10.

57. Setton, D. (1997), "Information Overload," *Forbes*, Vol. 160, No. 6, pp. 18-20.

58. Shneiderman, B. (1994), "Sorting Out Searching: A User-Interface Framework for Text Searches," IEEE Software, Vol.11, No. 6, pp. 70-77.

59. Shneiderman, B. (1996), "The eyes have it: A task by data type taxonomy of information visualizations", Proceedings of the IEEE Symposium on Visual Languages, pp. 336-343.

60. Shneiderman, B. (1997) *Designing The User Interface, Strategies for Effective Human-Computer Interaction, Third Edition*, Addison Wesley, Reading, MA.

61. Shneiderman, B., Byrd, D., Croft, W. B. (1998), "Dynamic Queries for visual information seeking," Communications of the ACM, Vol.41, No. 4, pp. 95-98.

62. Stasko, J., Catrambone, R., Guzdial, M., McDonald, K. (2000), "An Evaluation of Space-Filling information Visualizations for Depicting hierarchical Structures," Technical Report GIT-GVU-00-03, Georgia Institute of Technology, Graphics, Visualization and Usability Center.

63. Suh, K. S., Jenkins, A. M. (1992), "A Comparison of Linear Keyword and Restricted Natural Language Data Base Interfaces for Novice Users," Information Systems Research, Vol. 3, No. 3, pp. 252-272.

64. Swan, R.C., Allan, J. (1998), "Aspect Windows, 3-D Visualizations, and Indirect Comparisons of Information Retrieval Systems," Proceedings ACM SIGIR98 Conference, August 24-28, pp. 173-181.

65. Swan, R.C., Allan, J., Byrd, D. (1998), "Evaluating a Visual Retrieval Interface: AspInqury at TREC-6," CHI 98 Workshop on Innovation and Evaluation in Information Exploration Interfaces, Los Angeles.

66. Tan, J. K. H., Benbasat, I. (1990), "Processing of Graphical Information: A Decomposition Taxonomy to Match Data Extraction Tasks and Graphical Representations," Information Systems Research, Vol. 1, No. 4, pp. 416-439.

67. Tkach, D. (1998), "Technology Text Mining - Turning Information into Knowledge - A White Paper from IBM," http://allen.comm.virginia.edu/jtl5t/whiteweb.html, 01/22/01.

68. Todd, P., Benbasat, I. (1992), "The Use of Information in Decision Making: An Experimental Investigation of the Impact of Computer Based Decision Aids," Management Information Systems Quarterly, Vol. 16, No. 3, pp. 373-393.

69. Treeven, L., Hill, W. (1999), "Constructing, Organizing, and Visualizing Collections of Topically related Web Resources," ACM Transactions on Computer-Human Interaction, Vol. 6, No. 1, pp. 67-94.

70. Van Rijbersgen, C. J. (1979), *Information Retrieval.* Butterworths, London, 2nd edition.

71. Vessey, I. (1991), "Cognitive Fit: A Theory-Based Analysis of the Graphs Versus Tables Literature," Decision Sciences, Vol. 6, No. 1, pp. 219-240.

72. Wang, P., Hawk, W. B., Tenopir, C. (2000), "Users' interaction with World Wide Web resources: an exploratory study using a holistic approach," Information Processing and Management, Vol. 36, pp. 229-251.

73. Wise, J. A., Thomas, J. J., Pennock, K. Lantrip, D., Pottier, M., Schur, A., and Crow, V. (1995), "Visualizing the Non-visual: Spatial Analysis and Interaction with Information from Text Documents," Proceedings of the IEEE Information Visualization Symposium 51-59.

74. Witkin et al. (1971), *A Manual For The Embedded Figures Test*, Palo Alto, CA, Consulting Psychologists Press.

75. Worcester, B. A. (1997), "Information Overload," *Hotel & Motel Management*, Vol. 212, No. 4, pp. 52-53.

76. Wu, J. (1997), Excite Corporation, personal communication with Ophir Frieder and David Grossman, November 21.

77. Zamir, O. (1998), "Visualization of Search Results in Document Retrieval Systems," General Examination Report, Department of Computer Science and Engineering, University of Washington.

78. Zamir. O., Etzioni O. (1998), "Web document clustering: a feasibility demonstration," Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98), pp. 46-54.

79. Zamir. O., Etzioni O. (1999), "Grouper: A Dynamic Clustering Interface to Web Search Results," Proceedings of the 8th International World Wide Web Conference (WWW8).

APPENDIX A

A Sample from the Java Code for Visualizing the Hierarchy of Documents

```
public void paint(Graphics g)
        {
        int len,n_len;String dots="...",Nam;
        for(int i=0;i<number+1;i++)
           {
           if(drawdata[i].visible)
              {
              center=new Point(
((drawdata[i].left_top).x+(drawdata[i].right_bottom).x)/2,((drawdata[i].left_top)
.y+(drawdata[i].right_bottom).y)/2);
              node=drawdata[i];
              g.setColor(node.RectColor);

g.fillRect((node.left_top).x,(node.left_top).y,(int)node.width,(int)node.height);
              g.setColor(Color.black);

g.drawRect((node.left_top).x,(node.left_top).y,(int)node.width,(int)node.height);

              FontMetrics fm = getFontMetrics(getFont());
              len=(fm.stringWidth(node.Name));
              g.setColor(Color.black);

                //if rectangle is vertically oriented or width below 100,name is
printed in vertically rotated manner.

if((drawdata[i].width<100)&&(drawdata[i].width<drawdata[i].height))
                   {
                   int S_len=(node.Name).length();

                   len=(g.getFontMetrics().getHeight()-6)*S_len;//remember we
have substracted '6' while drawing vert.string.
                   if(len>drawdata[i].height)
                      {
                      float ratio=(float)drawdata[i].height/(float)len;
                      Nam=(node.Name).substring(0,(int)((S_len*ratio)-
4)).concat(dots);
//                      n_len=(fm.stringWidth(Nam));
                      n_len=(g.getFontMetrics().getHeight()-6)* Nam.length();
   //the integers are the result of trial and error method for better placement
of the names in the rectangles.
                    DrawVert(Nam,g,(center.x-3),(center.y - n_len/2 +6));

/*                    Graphics2D g2d = (Graphics2D) g;
                      g2d.rotate(-Math.PI/2,center.x,center.y);
                      g2d.drawString(Nam,(center.x-n_len/2),(center.y));
                      g2d.drawString(Nam,(center.x),(center.y));
                      g2d.rotate(Math.PI/2,center.x,center.y);*/
                      }
                   else
                      {//the integers are the result of trial and error method
for better placement of the names in the rectangles.

                         DrawVert(drawdata[i].Name,g,(center.x),(center.y-
len/2+6));
```

```
//                          Graphics2D g2d = (Graphics2D) g;
//                          g2d.rotate(-Math.PI/2,center.x,center.y);
//                          g2d.drawString(drawdata[i].Name,(center.x-
len/2),(center.y));
//                          g2d.rotate(Math.PI/2,center.x,center.y);*/
                         }
                      }
                  else
                      {
                       if(len>drawdata[i].width)
                           {
                            float ratio=(float)drawdata[i].width/(float)len;
                            int S_len=(node.Name).length();
                            Nam=(node.Name).substring(0,(int)((S_len*ratio)-
4)).concat(dots);
                            n_len=(fm.stringWidth(Nam));
                            g.drawString(Nam,(center.x-n_len/2),center.y);
                            }
                       else
                            g.drawString(drawdata[i].Name,(center.x-len/2),center.y);
                      }
                  }
              }
        }//method paint.

public void mouseMoved(MouseEvent me)
                       {
                       Point point = me.getPoint();
                      GetNode(point);
                       ap.NameShow(curr_name);
                       }
```

APPENDIX B

A Sample from the Java Code for Zooming a Part of the Visualization

```
public void PointClicked(Point location,AppletContext appContext)
       {
       int i;boolean gotit=false;
       int x=location.x;
       int y=location.y;

       for(i=0;i<number+1;i++)
           {
           if(drawdata[i].visible)
              {
             if(  (drawdata[i].left_top.x < x)&&(drawdata[i].right_bottom.x > x)
             &&(drawdata[i].left_top.y < y)&&(drawdata[i].right_bottom.y > y) )
                   {
                   gotit=true;
                   lucky_node=i;
                   }
              }
              if(gotit)
                 break;
           }
       if  (drawdata[lucky_node].weight==1)
          {
          openurl(drawdata[lucky_node].URL, appContext);
          leaf=true;//we don't process if the clicked node is a leaf.
          }

       else //if the clicked node is cluster.

       leaf=false;
       if((lucky_node!=0)&&(drawdata[lucky_node].DOI==1))//if the node being
clicked first time reset the others's DOI's
          for(i=0;i<number+1;i++)
              drawdata[i].DOI=1;

//code for adjusting scaling factor "SCALE".
       if(gotit)
       {
           float min_weight,next_min_weight, par_wt,total_wt;
           Vector temp_ch=new Vector();
           if(drawdata[lucky_node].NumberOfChildren>0)
              {
                total_wt=drawdata[0].weight;//total weight is weight of node 0;
                par_wt=drawdata[lucky_node].weight;
                temp_ch=drawdata[lucky_node].children_IDs;

min_weight=drawdata[(((Integer)temp_ch.elementAt(0)).intValue())].weight;
                next_min_weight=min_weight;

                min_weight=find_min(drawdata[lucky_node]);
                if(  (min_weight/par_wt)<=THRESHOLD)
                   {
                   parent_invisible=true;//flag if parent is invisible.
                   invisible_parent_node=lucky_node;//remember the parent which
is to be invisible.
```

```
                    no_of_visible_nodes=0;
                    for(i=0;i<temp_ch.size();i++)//remember the children numbers
who are visible, irrespective of their weights.


        addMouseMotionListener(new MouseMotionAdapter()
               {
                public void mouseMoved(MouseEvent me)
                      {
                      Point point = me.getPoint();
                     GetNode(point);
                      ap.NameShow(curr_name);
                      }
                })


  public void reset()
        {
        for(int i=0;i<number+1;i++)//adjust the DOI.
           drawdata[i].DOI=1;
        reset=true;
        parent_invisible=false;//so that the manipulated cluster resets.
        UpdateDrWeights();
        SetVisibility();
        CalcDimensions();
        repaint();
        }
```

APPENDIX C

Oklahoma State University
Institutional Review Board

Protocol Expires: 8/20/01

Date : Monday, August 21, 2000 IRB Application No BU014

Proposal Title: VISUALIZATION SUPPORT FOR MANAGING INFORMATION OVERLOAD IN THE WEB ENVIRONMENT

Principal
Investigator(s) :

Ozgur Turetken	Ramesh Sharda
320A CBA	321 CBA
Stillwater, OK 74078	Stillwater, OK 74078

Reviewed and
Processed as: Exempt

Approval Status Recommended by Reviewer(s) : Approved

Signature : Carol Olson

Carol Olson, Director of University Research Compliance

Monday, August 21, 2000
Date

Approvals are valid for one calendar year, after which time a request for continuation must be submitted. Any modifications to the research project approved by the IRB must be submitted for approval with the advisor's signature. The IRB office MUST be notified in writing when a project is complete. Approved projects are subject to monitoring by the IRB. Expedited ɤ

VITA

Ozgur Turetken

Candidate for the Degree of

Doctor of Philosophy

Thesis: VISUALIZATION SUPPORT FOR MANAGING INFORMATION OVERLOAD IN THE WEB ENVIRONMENT

Major Field: Business Administration

Biographical:

Education: Received Bachelor of Science degree in Electrical Engineering, and Master of Business Administration degree from Middle East Technical University, Ankara, Turkey in July 1994, and September 1999 respectively. Completed the Requirements for the Doctor of Philosophy degree with a major in Business Administration at Oklahoma State University in May 2001.

Experience: Employed by Middle East Technical University as a graduate assistant between 1994 and 1996, and by Oklahoma State University as a graduate teaching associate between 1996 and 2000. Have been working for Temple University as an assistant professor of Management Information Systems since September 2000.

Professional Memberships: Association of Information Systems, Decision Sciences Institute, Institute of Operations Research and Management Science