3-2002

Bubble World - A Novel Visual Information Retrieval Technique

Christopher L. Van Berendonck

BUBBLE WORLD
A NOVEL VISUAL INFORMATION
RETRIEVAL TECHNIQUE

THESIS

Christopher L. Van Berendonck, Flight Lieutenant, RAAF

AFIT/GCS/ENG/02M-09

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
9 Mar 02	Final	Jun 01 - Mar 02

Title and Subtitle Bubble World - A Novel Visual Information Retrieval Technique	**Contract Number**
	Grant Number
	Program Element Number
Author(s) Flight LT Christopher L. Van Berendonck, RAAF	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Bldg 640 WPAFB OH 45433-7765	**Performing Organization Report Number** AFIT/GCS/ENG/02M-09
Sponsoring/Monitoring Agency Name(s) and Address(es) Capt David Marsh AFRL/Information Directorate Information Technology Division Rome NY	**Sponsor/Monitor's Acronym(s)**
	Sponsor/Monitor's Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes The original document contains color images.	

Abstract
With the tremendous growth of published electronic information sources in the last decade and the unprecedented reliance on this information to succeed in day-to-day operations, comes the expectation of finding the right information at the right time. Sentential interfaces are currently the only viable solution for searching through large infospheres of unstructured information, however, the simplistic nature of their interaction model and lack of cognitive amplification they can provide severely limit the performance of the interface. Visual information retrieval systems are emerging as possible candidate replacements for the more traditional interfaces, but many lack the cognitive framework to support the knowledge crystallization process found to be essential in information retrieval. This work introduces a novel visual information retrieval technique crafted from two distinct design genres: (1) the cognitive strategies of the human mind to solve problems and (2) observed interaction patterns with existing information retrieval systems. Based on the cognitive and interaction framework developed in this research, a functional prototype information retrieval system, called Bubble World, has been created to demonstrate that significant performance gains can be achieved using this technique when compared to more traditional text-based interfaces. Bubble World does this by successfully transforming the internal mental representation of the information retrieval problem to an efficient external view, and then through visual cues, provides cognitive amplification at key stages of the information retrieval process. Additionally, Bubble World provides the interaction model and the mechanisms to incorporate complex search schemas into the retrieval process either manually or automatically through the use of predefined ontological models.

Subject Terms
Visual Information Retrieval, Cognitive Propositions, Design Corollaries, Ontology Models

Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU
Number of Pages 174	

AFIT/GCS/ENG/02M-09

THESIS

Presented to the faculty of the Graduate School of Engineering & Management

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science (Computer Science)

Van Berendonck, C. L.

Flight Lieutenant, RAAF

March 2002

.

AFIT/GCS/ENG/02M-09

Acknowledgments

I would first like to thank my advisor, LtCol Timothy Jacobs, for his guidance and patience throughout the course of this research. His dedication and enthusiasm has been invaluable to the success of this thesis. I would like to extend a special thanks to Maj Karl Mathias for not only providing a firm foundation in information retrieval methods but also providing valuable assistance along the way. Additionally I would like to thank the members of the 02M weekly research group, for their technical input and constant comradeship. Their thoughts and distractions provided a great source of motivation along the rocky road to completion.

Foremost, I would to thank my family for their undying devotion through it all. My wife for her patience and understanding, my oldest son for his enthusiasm in my work and my youngest son in ensuring I did not take this too seriously.

Table of contents

Page

Acknowledgments .... **iv**

List of Figures .... **x**

List of Tables .... **xii**

Abstract .... **xiii**

I. Introduction 1

1.1 Problem Background *2*

1.2 Problem *5*

1.2.1 Problem Statement 5

1.2.2 Problem Methodology 6

1.3 Scope *8*

1.4 Thesis Overview *8*

II. Background 10

2.1 The Basics of Information Retrieval *10*

2.1.1 Historic Overview 10

2.1.2 The Retrieval Process 12

2.1.3 Modeling 14

2.1.3.1 Boolean Model 15

2.1.3.2 Vector Model 17

2.1.3.3 Probabilistic Model 19

2.1.4 Retrieval Evaluation 22

2.1.4.1 Recall and Precision 22

2.1.4.2 Harmonic Mean 24

2.1.4.3 E Measure ... 25

2.2 Interaction Models ... 25

2.3 Visual Interfaces for Information Retrieval and Browsing ... 28

2.3.1 VIBE and Lyberworld ... 30

2.3.2 WebVIBE ... 31

2.3.3 InfoCrystal ... 32

2.3.4 Other Visualization Applications ... 33

2.3.4.1 Scatter/Gather ... 33

2.3.4.2 TileBars ... 34

2.3.4.3 Tkinq ... 35

2.3.4.4 Bead ... 36

2.3.4.5 Narcissus ... 37

2.3.4.6 Cat-a-Cone ... 38

2.3.4.7 Envision ... 39

2.3.4.8 LightHouse ... 40

III. Framework Analysis ... 41

3.1 Cognitive Strategies ... 41

3.1.1 Cognitive Science in Visual Information Retrieval ... 43

3.1.1.1 The use of diagrams ... 45

3.1.1.2 Conceptual Design Issues ... 46

3.1.2 Cognitive Propositions ... 47

3.2 Design through usage analysis ... 48

3.2.1 Design Corollaries ... 49

3.2.1.1 Default Settings ... 49

3.2.1.2 Query Complexity ... 50

3.2.1.3 Query Terms ... 51

3.2.1.4 Term Specificity ... 51

3.2.1.5 Erroneous Queries 52

3.2.1.6 User Sessions 53

3.2.1.7 Query Refinement 55

3.2.1.8 Result Viewing 56

3.3 Summary *57*

IV. Design Methodology 59

4.1 Visual Cues *59*

4.2 Visual Dimensions *60*

4.3 Design Methodology *60*

4.3.1 The Simple Query 61

4.3.1.1 Query Specification 61

4.3.1.2 Visualizing the Simple Query 61

4.3.2 Visualizing Relationships 63

4.3.2.1 Coding 63

4.3.2.2 Spatial Positioning 66

4.3.2.3 C-Factor 68

4.3.3 Physicalization 70

4.3.4 Manipulating the query 71

4.3.5 Visualizing complex nested queries 72

4.3.5.1 The Boolean Filter 73

4.3.5.2 The Proximity Filter 78

4.3.6 Interaction 79

4.3.6.1 Overview 79

4.3.6.1.1 Exploratory View 81

4.3.6.1.2 Icon View 82

4.3.6.1.2.1 Additional Browsing Tools 84

4.3.6.1.3 Document View 86

4.3.7 Ontology Models 87
4.3.7.1 Integrating Ontology Models into Schemas 88
4.3.7.2 Thesaurus 92
V. Experimental Evaluation 93
5.1 Introduction 93
5.2 Related Research 94
5.3 Experimental Design 96
5.3.1 The Hypotheses 96
5.3.2 Test Platform 98
5.3.3 The Queries 98
5.3.4 The experiment 104
5.3.4.1 Introduction 104
5.3.4.2 Method 106
5.4 Experimental Analysis 107
5.4.1 Preliminary Discussion 107
5.4.2 Two-Sample t-test 107
5.4.3 Mean Average Times 108
5.4.3.1 Mean Average Accuracy 109
5.4.4 Additional t-test 109
5.4.5 Analysis and Discussion of Results 110
5.4.5.1 Comparison By Times 110
5.4.5.2 Comparison By Accuracy 116
5.4.5.3 Inclusion of Ontology Models 118
5.5 General Discussion 119
5.6 User Feedback 120
5.7 Conclusion 123
VI. Conclusion and Future Research 124

6.1 Conclusion *124*

6.2 Future Research *128*

6.2.1 Application Enhancements 128

6.2.2 Further Application Research 129

6.2.3 Other Applications 130

6.2.3.1 Web Search Engine 130

6.2.3.2 Email 131

6.3 Summary *131*

Appendix A. Tutorial Extracts 133

A.1 Textual Interface Tutorial *133*

A.2 Graphical Interface Tutorial *137*

Appendix B. Experiment Documentation 146

B.1 Experiment Questions *146*

B.2 Testing Answer Sheet *148*

B.3 User Testing Questionnaire *151*

Appendix C. Experimental Results 153

Bibliography 154

Vita 159

List of Figures

FIGURE 1. THE PROCESS OF RETRIEVING INFORMATION (ADAPTED FROM BAEZA-YATES, RIBEIRO-NETO [7]) 13

FIGURE 2. BOOLEAN COMBINATIONS AS VENN DIAGRAMS (ADAPTED FROM DJOERD HIEMSTRA [ ]) 15

FIGURE 3. THE STANDARD MODEL FOR INFORMATION ACCESS[] 26

FIGURE 4. KNOWLEDGE CRYSTALLIZATION PROCESS (CARD, MACKINLAY, AND SHNEIDERMAN) 28

FIGURE 5. VIBE GRAPHICAL INTERFACE 31

FIGURE 7. INFOCRYSTAL GRAPHICAL INTERFACE 33

FIGURE 8. SCATTER/GATHER INTERFACE 34

FIGURE 9. TILEBARS INTERFACE 35

FIGURE 10. TKINQ INTERFACE 36

FIGURE 11. BEAD INTERFACE 36

FIGURE 12. NARCISSUS INTERFACE 37

FIGURE 13. CAT-A-CONE INTERFACE 38

FIGURE 14. ENVISION INTERFACE 39

FIGURE 15. LIGHTHOUSE INTERFACE 40

FIGURE 16. STRUCTURE OF A SIMPLE QUERY 62

FIGURE 17. EXAMPLES OF SIMPLE QUERIES 63

FIGURE 18. DEMONSTRATION OF SOME CODING TECHNIQUES USED 65

FIGURE 19. THE EFFECT OF NEIGHBORHOOD ZONING OF ICONS 70

FIGURE 20. REPRESENTATION OF A SIMPLE SCHEMA 73

FIGURE 21.BOOLEAN FILTER EXAMPLE – STEP 1 74

FIGURE 22. BOOLEAN FILTER EXAMPLE – STEP 2 75

FIGURE 23. BOOLEAN FILTER EXAMPLE – STEP 3 76

FIGURE 24. BOOLEAN FILTER EXAMPLE – STEP 4 77

Figure 25.Proximity Filter Example 78
Figure 26. The three interaction views 81
Figure 27. The formation of collection sets 83
Figure 28. Transitions through the Icon View abstraction layers 84
Figure 29. Icon View Environment 85
Figure 30. Query term hits within document content 87
Figure 31. Example of an Ontology Model of an Aircraft 88
Figure 32. Aircraft Ontology with Identified Categories 89
Figure 33. Example Transformation of the Ontology Model of an Aircraft 90
Figure 34. Example of Automatic Schema Expansion 91
Figure 35. Adding Thesaurus Words To the Model 92
Figure 36. Comparison of Average Times by Question 111
Figure 37. Comparisons of Average Times by Query Type 111
Figure 38. Result of Issuing Question 6 113
Figure 39. Test Case Example, single schema 115
Figure 40. Query used to answer Questions 10, 11 and 12 116
Figure 41. Comparison of Accuracy by Question Number 117
Figure 42. Comparison of Accuracy by Question Type 117
Figure 43. Text Interface 133
Figure 44. Text interface with Example Query 134
Figure 45. Document view of Text Interface 135
Figure 46. Bubble World Exploratory View 138
Figure 47. Bubble World Icon View 140
Figure 48. Bubble World Document View 141
Figure 49. Example of Selecting an Orb 142
Figure 50. Example of Building Multiple Schemas 144

List of Tables

TABLE 1. REQUIREMENTS FRAMEWORK ... 57

TABLE 2. TYPE 1 QUERIES ... 100

TABLE 3. TYPE 2 QUERIES ... 101

TABLE 4. TYPE 3 QUERIES ... 102

TABLE 5. TYPE 4 QUERIES ... 103

TABLE 6. SPREAD SHEET OF EXPERIMENTAL RESULTS ... 153

Abstract

With the tremendous growth of published electronic information sources in the last decade and the unprecedented reliance on this information to succeed in day-to-day operations, comes the expectation of finding the right information at the right time. Sentential interfaces are currently the only viable solution for searching through large infospheres of unstructured information, however, the simplistic nature of their interaction model and lack of cognitive amplification they can provide severely limit the performance of the interface. Visual information retrieval systems are emerging as possible candidate replacements for the more traditional interfaces, but many lack the cognitive framework to support the knowledge crystallization process found to be essential in information retrieval. This work introduces a novel visual information retrieval technique crafted from two distinct design genres: (1) the cognitive strategies of the human mind to solve problems and (2) observed interaction patterns with existing information retrieval systems. Based on the cognitive and interaction framework developed in this research, a functional prototype information retrieval system, called Bubble World, has been created to demonstrate that significant performance gains can be achieved using this technique when compared to more traditional text-based interfaces. Bubble World does this by successfully transforming the internal mental representation of the information retrieval problem to an efficient external view, and then through visual cues, provides cognitive amplification at key stages of the information retrieval process. Additionally, Bubble World provides the interaction model and the mechanisms to incorporate complex search schemas into the retrieval process either manually or automatically through the use of predefined ontological models.

BUBBLE WORLD – A NOVEL VISUAL INFORMATION RETRIEVAL TECHNIQUE

I. Introduction

Until recently, Information Retrieval has been an area of interest for only a select group of people such as librarians and information experts. Today, with the popularity of personal computers and the exponential growth of the World Wide Web (WWW) [1:367], interest by the general public is rapidly increasing and for most, becoming part of every day life. On the rise, but perhaps less ominously, is the growth of digital libraries now popular both in and outside of educational institutions. Information is changing daily, and consumers of this information are becoming increasingly reliant on it for both their professional and private lives. For the military, information is considered so vital for successful day-to-day operations that an unprecedented amount of resources is expended on gathering it and providing access to those who need it. With this reliance comes the expectation of retrieving the right information in a timely fashion. "Good" information is not a measure of the quality of the source alone, but also the usefulness of that source. If it is retrieved beyond the lifespan of the task at hand, its usefulness reduces to hindsight. Therefore, the tools to manage, retrieve, and filter this information are critical to the success of those who rely on it.

Government and intelligence agencies, military services, defense industry, and military research and development centers invest heavily on up-to-the-minute, comprehensive and reliable global defense information. The U.S. Air Force has long recognized the importance of "information" and has produced a vision of the Battlespace Infosphere, which moves the information technology away from the concept of system or network centric approach to an information centric approach. Military commanders require precise command, control, and planning information for any given mission to ensure the front line war fighter does not unnecessarily put his or her life in harms way. In each case, sourcing, gathering and providing access to vital information is only half of the solution. Retrieving the correct information, making inferences about comparisons and understanding it to produce a solution to a problem in a timely fashion are critical to the overall success of all who rely on this information.

With the tremendous growth of electronic information sources comes the need for effective and efficient information retrieval methods. At the core of information retrieval is the ability to retrieve information that most closely matches the need of the user. This is known as the relevance of the information and relies on the information retrieval system being able to interpret the contents of the information set and then, in some fashion, return a ranked list to the user. The key concept here is how this relevance is decided. If done well, the returned information will be an exact set of what the user expected and nothing more. This suggests that relevance is not just choosing the best information, but also excluding information that has no use to the user.

1.1 Problem Background

Traditional information retrieval techniques typically return a list of documents ranked from most relevant to least relevant, based on the determination of a search algorithm. Although the search algorithms (or models) are improving, resulting in better relevance ordering, they are far from perfect. Not only are large lists of irrelevant documents presented to the user, but limited clues are presented for determining how best to adjust the query to improve the result. Even if search models could be improved to return only relevant documents, current studies suggest that interaction with an information collection through foraging alone provides only part of the knowledge gathering process [2].

The standard information access model used by almost all information retrieval engines assumes an interaction cycle consisting of a query specification (user specifying a need), the submittal and reformulation of that query to the system, the foraging for and receipt of the results, the examination and evaluation of the resultant set, and the reformulation of the query to repeat the process until the perfect result set is found. This model is simplistic in that it relies on an underlying assumption that the user's information need is static and progressive refinement leads to only relevant information. It does not take into account that users generally learn during this browsing process and then refine their query requirements accordingly. Consequently, under this interaction model, users are frequently confronted with long disorganized lists of essentially irrelevant information that does not address their need.

The interaction in the standard information retrieval model allows foraging of data through search queries and browsing but provides very little cognitive amplification to the user. If the relevant document set returned is not meaningful to the user, then he must begin again by issuing another query with almost no more prompting than before. An improvement to this technique is the addition of relevance marking, in which the user can select a document to be relevant (and perhaps others as not) or issue further queries based on the precision of a particular document. This helps in the determination of relevant documents, but it adds little to the knowledge collection process found to be essential in the search process of the user [3][4].

Information retrieval systems are divided along system-based and user-based concerns. While the system-based concerns deal mainly with efficient information representation and retrieval, the user-based concerns must deal with the context of the problem and the cognitive state of the user. Users are drawn to the retrieval system when they lack some knowledge to solve a task and therefore must search for information they know little or nothing about. As the user searches, he learn's more about the problem and potential solutions thus refining how he conceptualize's the problem. The need of the user is evolving throughout this search process. As time passes, this evolution can significantly change not only the queries they issue, but also the initial need. Even if the information is well known, vocabulary is a problem. Many users may know what it is they are looking for, but may have difficulty in articulating this in a way that is syntactically consistent with the retrieval system.

Good information retrieval systems must allow for this interactive dialog model. They must provide support for information seeking strategies such as searching and browsing and they must provide effective cues for the location, use, and characteristics of the retrieved information.

They must also provide sufficient feedback for interactive refinement of the information need. Current text list methods do not adequately support many of these characteristics, so researchers have looked elsewhere for answers.

In the last decade or so, visual interfaces have become an important part of the information retrieval process. They have emerged to improve the effectiveness of information retrieval on large information sources by addressing the interaction between the system and the user. The concept is simple: increase the information density by mapping the document similarity space into more dimensions. The reason is intuitive: amplify the cognitive perception of the user to improve the knowledge collection process. The results have been disappointing: no single interface has emerged as an effective replacement to the current single dimension text list approach. This is not to say that the visualization techniques are not effective, only to say that interfaces so far have delivered less than their full potential.

Consider the evaluation of text, numeric and graphical presentations for information retrieval interfaces conducted by the University of Pittsburgh [5]. The most striking finding was that over 60% of users tested preferred the visual methods, i.e., icon list and spring displays (such as VIBE) over the text based approaches even though performance was superior with the latter. This result stands in contrast to previous studies [6] that indicated that users of information retrieval systems were found to be resistant to these new visualization techniques. There are many influences that may have affected this paradigm change, such as improvements in hardware technology, however, the most likely reason is that user experience to this technology is changing and therefore, they are now more likely to expect visual interfaces. This study suggest that visualization techniques are not yet delivering the performance expected by the scientific community, but when they do, they stand a chance of being readily accepted by the greater populace.

It is widely accepted that visual information retrieval systems show promise, but do not yet deliver the performance and/or usability gains that have been expected by researchers. The difficulty lies in providing enough information to encourage forward browsing within the context of a search problem. It is generally thought that providing structure to the querying of unstructured information may benefit the discovery process, but this is usually found only in exact search systems that already have inherent structure and not with free text querying on information that has little or no structure at all. This research will investigate this concept further. The rest of this chapter is dedicated to stating the problem of this research paper, the scope of the research effort, the methodology to solve the problem and an overview of what is to come.

1.2 Problem

1.2.1 Problem Statement

"Design a novel information retrieval technique that can demonstrate performance gains when compared to more traditional text-based interfaces. The technique should successfully transform the internal mental representations of the information retrieval problem to an isomorphic external view, thereby allowing cognitive amplification at key stages of the information retrieval process. Additionally, the interaction model should allow the incorporation of complex search schemas to enhance the knowledge crystallization process. This should be achieved either manually or automatically through the use of predefined ontological models."

This problem has many facets that must be addressed. First, the design of a new technique requires an extensive background study on existing visual information retrieval techniques to be conducted to understand where the current level of research is. Second, allowing cognitive amplification at key stages of the information retrieval process requires understanding of a number of concepts: (1) how humans solve problems in information retrieval, (2) how they interact with an information retrieval system, (3) what it is that requires amplification and (4) what the key stages of the information retrieval process are. Third, an understanding of what the knowledge crystallization process is and how the schemas of the problem relate to the information retrieval process must be understood. Understanding this is vital in deciding how ontological models can help to automate this process. Fourth, demonstrating performance gains against a traditional text based interface requires an experiment to gain some empirical measurements to compare performance. The challenge for this research is to identify (1) the scenario in which the experiment is to be conducted, (2) the metrics that would be used to measure comparative performance based on this scenario and (3) a methodology to conduct the experiment successfully with limited resources and time.

1.2.2 Problem Methodology

The thesis begins with an investigation into information retrieval techniques, interaction models, and visualization techniques, in an attempt to understand some of the inherent shortfalls that have led to the limited results observed in visual information retrieval systems. This helps to define an informal knowledge base of "lessons learned" from previous research work, which helps circumvent the possibility of similar problems hampering the new system. This section does not critique or compare any existing work, instead it is used to understand the domain of the problem and to gain insight into what works and what does not.

Next, a design framework is developed based on two distinct design genres: (1) the cognitive strategies of the human mind to solve problems and (2) observed user interaction patterns with existing information retrieval systems. Understanding how humans think about information retrieval problems and how they interface with the information retrieval system to solve those problems, is a key challenge that must be addressed before any new techniques can be established. Therefore, considerable emphasis is placed on this component of the research, the end result of which is a generic framework of cognitive propositions and design corollaries that provide a useful guideline to the development of any information retrieval system.

With a framework and previous research to proceed, the next challenge is to culminate the best of these ideas into a new information retrieval technique capable of delivering performance gains to the user. To do this, the technique must transform the internal mental representation of the information retrieval problem to an isomorphic external view, and then through visual cues, provide cognitive amplification at key stages of the information retrieval process. Understanding the key stages of information retrieval comes from discovering how the knowledge crystallization process integrates into the interaction model of the user. Once this is known, providing the mechanisms to incorporate complex search schemas into the retrieval process is possible. At this stage reproduction of the schemas is done manually by the user, however, a natural extension to this is to incorporate automatic schema expansion through the use of predefined ontological models. Using these predefined models, a domain expert can guide users to search relevant areas of the information collection that may not be immediately visible to the user.

The next challenge for this research is to develop a functional prototype information retrieval system. The prototype system has three purposes in this research: (1) it provides a pre-production system to be used as a proof of concept research tool, (2) it provides the mechanism to perform experimental evaluation, thereby demonstrating the techniques developed in this thesis, and (3) it provides a platform to base additional research on once this thesis is complete.

The last challenge of this research is the validation and verification phase. In order to demonstrate the effectiveness of the techniques developed, one or both of the following approaches are generally adopted. The first is a 'recognition experiment', which tests the techniques in isolation of the prototype system. This can be useful in gaining insight into how well users understand the visual representations presented to them but provides no information about the effectiveness of the technique to perform information retrieval. The second method, and the one that is adopted in this research, is an experiment that directly compares the prototype system to a representative text-based system. This experiment has some obvious benefits since now direct comparisons of the two interfaces can be achieved in a 'real world' scenario, but it is also much more difficult to construct. This research approaches this problem in the following manner. First, background research is performed on experiments previously conducted with similar aims, in an attempt to pollinate those ideas into this research. Second, the query construct itself is examined to identify what makes a 'real world' scenario in information retrieval. This leads to a representative set of 'real world' queries that become the foundation to the experiment. Third, the experiment is designed around these queries and conducted on two select groups: the control group, which use the text-based interface to perform retrieval tasks and the experimental group, which use the new visual interface to perform the same tasks.

1.3 Scope

Improved information retrieval can benefit a wide range of domain applications, from the World Wide Web to digital libraries, and any large information source in between. In fact, any information item that can in some way be evaluated for relevance stands to gain by improving the interaction between the user and the system and amplifying the cognitive feedback through carefully selected visual techniques. However, each application has specific needs even if the underlying methodology is the same. An interface for web browsing would differ from a bibliographic search on a digital library even though the concept of information is similar.

For this research, it is enough to limit the concepts of this research to a prototype system. The information sources are limited to predefined and extensively tested collections that are readily available and provide a range of useful testing metrics. The prototype system is developed around these information collections to provide an application-oriented system suitable for testing. User testing focuses on comparing a representative text list method against the proposed system. No attempt is made to evaluate this system against other mainstream text list applications or visualization techniques.

1.4 Thesis Overview

This chapter addresses some of the motivational reasons behind developing multidimensional visualization interfaces and defines the particular problem for this research. Chapter 2 discusses the salient literature in this field to build a foundation and starting point for this research. Chapter 3 develops the requirements framework for any new visualization technique by investigating the psychology of how humans solve problems and the observations on how they interact with existing information retrieval systems. Chapter 4 then looks at the design methodology of the new visualization techniques by building on the framework developed in Chapter 3. Chapter 5 outlines the testing paradigm used for empirical analysis of the developed techniques against a system representative of current methods. The results of the study are presented and discussed at the end of this chapter. Chapter 6 concludes the research and discusses some possible areas for future work.

II. Background

2.1 The Basics of Information Retrieval

Information retrieval deals with the representation, storage, organization of, and access to information items [1]. This should be differentiated from data retrieval, which consist mainly of determining the information sources that match the components of the user query. Users of an information retrieval system are generally more concerned with retrieving information about a subject than retrieving data that satisfies a given query. A data retrieval language must precisely match information objects through the use of regular expressions or in a relational algebra expression. An information retrieval system attempts to interpret the contents of the information items (usually a document collection) and then rank them according to the perceived relevance to the user. This distinction is important since now the information retrieval system must cope with semantically ambiguous information that can lead to errors in the retrieved collection, sometimes referred to as 'imprecise searching'. However, organizing the information items based on the characterization of the user's information need is not a simple task. The difficulty comes from not only knowing how to extract information from the user at the appropriate time, but how to decide relevance based on this information. The following section is devoted to providing a brief overview of the instruments of automatic information retrieval.

2.1.1 Historic Overview

The information in this section has been compiled from the American Society of Indexers website [7]. Papyrus and leather scrolls as used by the ancient Greeks and Romans were one of the first methods used to store information in written form so that it could be retrieve later. Yet, as scholars began to write larger works and more information and data was stored, it became useful for them to devise other techniques to organize this information, so that locating certain passages would be easier for the reader. Pliny the Elder (died 79 A.D.) wrote a massive work called *The*

Natural History in 37 Books. In order to make the works friendlier to read, the entire first book was devoted to what was essentially a large table of contents. At the end of this preface, he indicated that this practice was first employed in Latin literature by Valerius Soranus, who lived during the last part of the second century B.C. and the first part of the first century B.C.

Indexing is one of the oldest among the figurative or applied senses of the word, and originates back to the times of ancient Rome. The term *index* was used for the little slip attached to papyrus scrolls on which the title of the work (and sometimes also the name of the author) was written so that each scroll on the shelves could be easily identified without having to pull them out for inspection. However, since no two scrolls were ever exact and numerical indicates such as page or chapter numbers were not in use, very few were constructed. The first modern indexes began to be compiled only after the invention of printing around 1450. Most of these early indexes were arranged only by the first letter of the first word, the rest being left in no particular order. Gradually, alphabetization advanced to an arrangement by the first syllable, that is, the first two or three letters, the rest of an entry still being left unordered. Only very few indexes compiled in the 16th and early 17th centuries had fully alphabetized entries, but by the 18th century full alphabetization became the rule. For centuries, indexes were created manually as categorization hierarchies and in fact most libraries today still use some form of categorical hierarchy to classify their collection. However, with the advent of the computer came the proliferation of information sources requiring indexing and categorizing. The tide was turning for a new way of conducting business.

Information retrieval systems were initially developed in government laboratories to support research towards science and technology. They were essentially based around bibliographic databases containing largely textual information, and were used by highly trained search intermediaries. Libraries were among the earliest institutions to make use of these information retrieval systems, which took the form of commercial vendor products searching remote electronic databases to provide reference services to patrons.

The second generation of information retrieval systems provided increased search capability which allowed searching by subject headings, by keywords, and by other more complex query facilities. However, as the modern computer continued to advance along with the meteoric rise of the World Wide Web, some fundamental changes were occurring. Access to various information sources became cheaper and subsequently reached a greater audience. Advances in digital communication networks provided remote access to information that was previously not possible. The freedom to publish information and to subscribe to information was unprecedented. Suddenly, for the first time in history, the mass populous had access to a highly interactive large publishing medium. This overwhelming interest has lead to the current deployment of information retrieval systems, a third generation focusing on improving graphical interfaces, electronic forms, hypertext features and open system architectures.

2.1.2 The Retrieval Process

Assuming that a document database exists and is indexed in some way, then the retrieval process begins with a 'user need'. The user need is traditionally expressed in a free text manner, so to provide a logical view of this need to the system a series of text operations must be performed. These text operations are the same that are used for indexing purposes of the documents themselves, and include techniques such as structure recognition, removal of accents, spacing and stopwords, identification of noun groups, and finally stemming. Once the logical view of the text is extracted, then query operations might be applied before the query itself is generated. The query is now a system representation of the user need and is used to obtain the retrieved documents. This process is augmented with indexing to reduce search times.

At this point, the documents must be ranked according to some relevance measure before being presented to the user. What happens next tends to be application dependent, however, the user may select a subset of the retrieved documents to initiate a user feedback cycle. In this way, the system is able to change the query formulation based on this additional user information. The expectation here is that the new representation will yield better results iteratively as the cycle continues. The architecture of this process is shown in Figure 1.

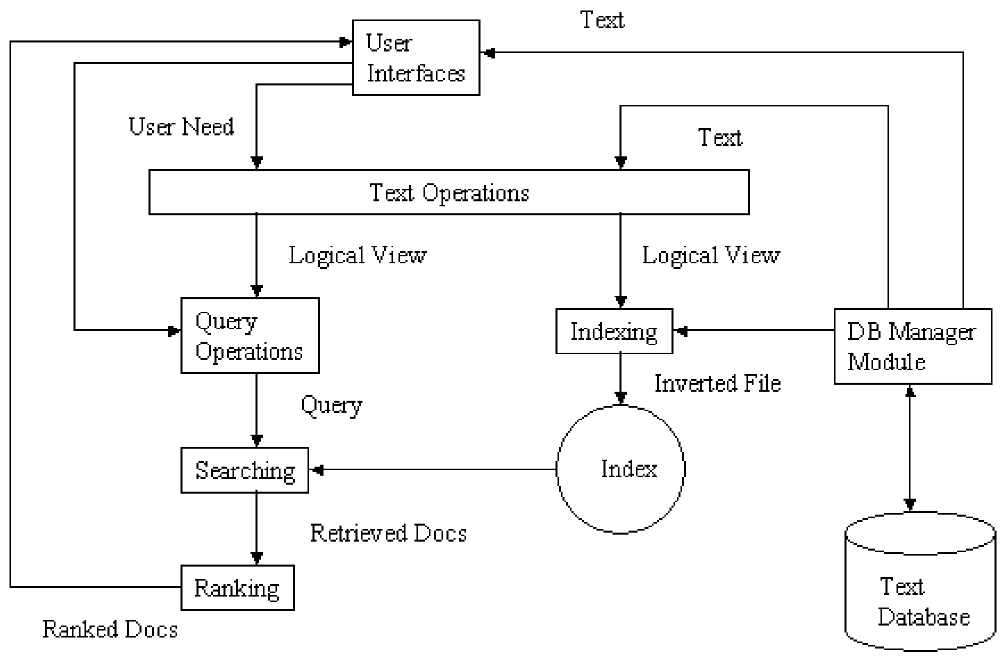


Figure 1. The Process of retrieving information (adapted from Baeza-Yates, Ribeiro-Neto [1])

The retrieval process requires some fundamental assumptions to be valid. First, the user must be able to declare their need using the interface provided. This is almost never the case, since the user is required to provide a direct representation for the query that the system will execute. Second, the user need is static over the search. However, if the first assumption is not valid, then the user will more than likely formulate a query that is poorly defined or inherently broad. This will force the user to browse the documents in the collection and not search them, subsequently changing the need of the user. Third, the context of the user need is not changed by the text operations. This requires the text language to be linguistic and amenable to pure formal manipulation. This is quite an appealing assumption since it reduces the representations to a manageable level for the system, however in doing so, it hides the complexity inherent in the human language from the matching procedure. The very same mechanisms that make the matching process difficult, homeosemy, makes the human language a rich communication medium. For this reason, it is unlikely that the third assumption is valid. Fourth, the documents chosen as relevant by the user are indeed more relevant than those chosen by the system. This intuitively seems reasonable, but relies strongly on the previous assumptions also being relevant. As discussed above, this may not be the case.

2.1.3 Modeling

At the core of information retrieval is the ability to retrieve information that most closely matches the need of the user. This is known as the relevance of the information and relies on the information retrieval system being able to interpret the contents of the information set and then, in some fashion, return a ranked list to the user. The key concept here is how this relevance is decided. If done well, the returned information set will be an exact list of what the user expected and nothing more. This suggests that relevance is not just choosing the best documents, but also excluding documents that have no use to the user.

The traditional technique of determining relevance revolves around matching key index terms, expressed in some way by the user, to the text of the retrieval documents. By key index term, one generally means a word that contains meaning and that adds value to a search. How relevance is determined through key index terms depends on the ranking algorithm that has been implemented. Ranking algorithms operate on basic premises regarding the notion of document relevance. These sets of premises yield distinct information retrieval models, which in turn attempt to predict what is and is not relevant based on the specified user need.

Retrieval models are broadly categorized into two retrieval taxonomies, exact-match and best-match. Exact-match retrieval models require the query to specify precise retrieval criteria and return a set of documents that precisely match the query, and nothing more. Since the queries are well structured, they tend to be efficient and predictable but often are difficult for the user to formulate and suffer from poor results due to their inherent precision. Best-match retrieval models allow a query to specify good or 'best matching' criteria and returns a ranked ordered list of documents based on a measure of quality of relevance. The queries tend to be full text and are therefore easier to formulate and, generally speaking, return superior results to those of exact-matching. The problem with best-match models is the underlying assumption that keywords are orthogonal, that is, collections of words have no semantic meaning. The model can be extended to include phrase or proximity matching to help with this orthogonal issue, however, since full text is not a natural language in itself, some cognitive expressions such as "control" can never be defined.

Three classical information retrieval models are the boolean, the vector and the Probabilistic models. Termed classical not only in the sense of longevity but also because they represent classical problems in the information retrieval arena. The next section discusses these information retrieval models in brief detail.

2.1.3.1 Boolean Model

Perhaps the simplest of the retrieval models both intuitively and mathematically, is the boolean model. The boolean model is an exact-match retrieval method, which requires the query to be specified as boolean expressions with precise semantics. Three logical operators are used for this purpose: logical product 'AND', logical sum 'OR' and the logical difference 'NOT'. These can be visualized in the Venn diagrams of Figure 2.

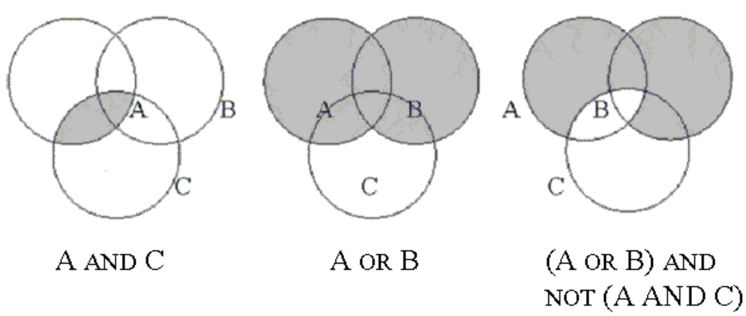


Figure 2. Boolean combinations as Venn Diagrams (adapted from Djoerd Hiemstra [8])

A more formal definition of the boolean model is given by Baeza-Yates and Ribeiro-Neto [1] as:

> **Definition:** *For the boolean model, the index term weight variables are all binary i.e.* $w_{i,j} \in \{0,1\}$. *A query q is a conventional boolean expression. Let* $\vec{q}_{dnf}$ *be the disjunctive normal form for the query q. Further, let* $\vec{q}_{cc}$ *be any of the conjunctive components of* $\vec{q}_{dnf}$. *The similarity of a document* d_j *to the query q is defined as*
>
> $$sim(d_j, q) = \begin{cases} 1 & \text{if } \exists \vec{q}_{cc} \mid (\vec{q}_{cc} \in \vec{q}_{dnf}) \wedge (\forall k_i, g_i(\vec{d}_j) = g_i(\vec{q}_{cc})) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$
>
> *where* g_i *is a function that returns the weight associated with the index term* k_i *in any t-dimensional vector.*
>
> *if* $sim(d_j,q) = 1$ *then the boolean model predicts that the document* d_j *is relevant to the query q (it might not be). Otherwise, the prediction is that the document is not relevant.*

The boolean model was the first to be implemented on a computer. It is simple, efficient and still widely used today. However, it is also the model that has generated the most criticism over time for the very reasons it has become popular. First, the information retrieval is based on an exact matching of boolean formalisms, which does not perform nearly as well as the uncertainty models. Second, its simplicity is in the implementation and not in its use. Users require training to learn how to formulate queries and even then, it is not a trivial task to translate an information need into a formal boolean expression. Third, there is little control of the output size. Documents are not rank ordered but returned as a dichotomous collection of relevant or non-relevant documents. Fourth, the terms are assumed to be independent of each other resulting in semantically incorrect expressions. Fifth, no mechanisms are available to specify weight factors to specific index terms.

2.1.3.2 Vector Model

The vector model was conceived in an attempt to address the fact that binary weight matching was too limiting for an information retrieval framework. The model, suggested by Salton and McGill [9] and based on Luhn's similarity criterion [10], considered the index representations and the query as vectors in a high dimensional Euclidean Space, where each term is assigned a dimension. The similarity measure between two vectors $\vec{d}$ and $\vec{q}$ is then the cosine of the angle that separates the two vectors. Again, more formally by Baeza-Yates and Ribeiro-Neto [1]:

> **Definition:** *For the vector model, the weight $w_{i,j}$ associated with a pair (k_i, d_j) is positive and non-binary. Further, the index terms in the query are also weighted. Let $w_{i,q}$ be the weight associated with the pair [k_i,q] where $w_{i,q} \geq 0$. Then, the query vector* $\vec{q}$ is defined as $\vec{q}=(w_{1,q}, w_{2,q}, \ldots, w_{t,q})$ where t is the total number of index terms in the system. The vector for a document d_j is represented by
>
> $\vec{d}_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$.

$$sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} * w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} * \sqrt{\sum_{j=1}^{t} w_{i,q}^2}} \tag{2}$$

The vector model uses a framework of partial matching of key index terms to determine the relevance of a particular document. It does this by using non-binary term weights to determine the degree of similarity between the document terms and the query. In this way, partial matching and frequency of key index terms contributes to the score of the document. One of the key concepts of the term weight metric is not only how well each index term matches a particular document, but also how unique that index term is to that document. For instance, common terms seen in all documents add little value to a particular search since no differentiation between the documents is obtained. Terms unique to only some documents, however, have the greatest distinguishing factor and therefore are weighted accordingly. This is known as the inverse document frequency or the idf factor. The definition provided by Baeza-Yates and Ribeiro-Neto [1] is:

> **Definition:** *Let N be the total number of documents in the system and n_i be the number of documents in which the index term k_i appears. Let $freq_{i,j}$ be the raw frequency of term k_i in the document d_j (i.e., the number of times the term k_i is mentioned in the text of the document d_j). Then, the normalized frequency $f_{i,j}$ of term k_i in document d_j is given by:*
>
> $$f_{i,j} = \frac{freq_{i,j}}{\max_l \ freq_{l,j}} \qquad (3)$$
>
> *where the maximum is computed over all terms which are mentioned in the text of the document d_j. If the term k_i does not appear in the document d_j then $f_{i,j} = 0$. Further, let idf_i, inverse document frequency for k_i, be given by:*
>
> $$idf_i = \log \frac{N}{N_i} \qquad (4)$$
>
> *the best known term-weighting schemes use weights which are given by:*
>
> $$w_{i,j} = f_{i,j} * idf_i \qquad (5)$$
>
> *where $w_{i,j}$ is the calculated weight of term i in document j*
>
> *further, Salton and Buckley [11]suggest:*
>
> $$w_{i,q} = \left(0.5 + \frac{0.5 \, freq_{i,q}}{\max_l \ freq_{l,q}} \right) idf_i \qquad (6)$$
>
> *where $freq_{i,q}$ is the raw frequency of the term k_i in the text of the information request q and $w_{i,q}$ is the associated weight.*

The vector model has some principle advantages over the boolean model. First, the partial matching strategy now allows documents to be returned that are only approximately what the query requested. An uncertainty model that matches the query and the document in a best-match fashion will generally outperform those that require perfect matching. Second, term-weighting is now possible which improves the retrieval performance. Third, the query specification process is reduced to declaring full text, making the formulation of queries for the user much easier. Fourth, the document set returned is a ranked list sorted by most relevant to least relevant, allowing the output collection size to be specified. Fifth, the vector model provides a basis for a wide range of retrieval operations such as indexing, relevance feedback, document classification and clustering.

There are some limitations with the vector model but these are not insurmountable. The discriminating power of the model generally requires more index terms in the query than that required for a boolean query and those terms are essentially considered orthogonal by design. In practice neither of these issues poses a threat to the validity of the model. Free text querying allows easy manipulation of the query to gain increased discriminatory power, and research has yet to suggest that considering term dependencies improves the relevance of the retrieved documents [1:30].

2.1.3.3 Probabilistic Model

Maron and Kuhns [12] argued against Luhn's ideas on using similarity, and instead suggest that the documents in a collection should be ranked according to their probability of relevance. Robertson [13] called this criterion the 'probability ranking principle' and formulated the following principle:

> *If a reference retrieval system's response to each request is a ranking of the documents in the collections in order of decreasing probability of usefulness to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data has been made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data.*

The probabilistic model uses a statistical framework to gain insight into the document collection. It assumes that there is a set of documents that will provide an ideal answer set which in turn defines properties characterized by the semantics of the index terms found in them. However, the ideal answer set is not known initially, if it were, the problem would be trivial. So, the probabilistic model relies on a relevance feedback cycle from the user to identify those documents that might describe the answer set, and in doing so repeatedly, evolves the description closer and closer to that describing the ideal answer set.

The probabilistic principle assumes that the probability of relevance depends on the query and document representations only. Further, it assumes that a subset of preferred documents exists as the answer set for the query. The question remains as to how to compute the probabilities of relevance given these two assumptions.

Given a query q, the probabilistic model defines the degree of similarity between q and any document in the collection d_j, as the ratio of the probability that the document d_j is relevant, to the probability that the document is not. More formally, from Baeza-Yates and Ribeiro-Neto [1], the definition of the probabilistic model can be given by:

> **Definition:** *For the probabilistic model, the index term weight variables are all binary i.e.,* $w_{i,j} \in \{0,1\}$, $w_{i,q} \in \{0,1\}$. *A query q is a subset of index terms. Let R be the set of documents known (or initially guessed) to be relevant. Let* $\overline{R}$ *be the complement of R (i.e., the set of non-relevant documents). Let* $P(R \mid \vec{d}_j)$ *be the probability that the document* d_j *is relevant to the query q and* $P(\overline{R} \mid \vec{d}_j)$ *be the probability that* d_j *is non-relevant to q. The similarity* $sim(d_j,q)$ *of the document* d_j *to the query q is defined :*
>
> $$sim(d_j,q) = \frac{P(R \mid \vec{d}_j)}{P(\overline{R} \mid \vec{d}_j)} \qquad (7)$$
>
> $P(k_i \mid R)$ *stands for the probability that the index term* k_i *is present in a document randomly selected from the set R.* $P(\overline{k}_i \mid R)$ *stands for the probability that the index term* k_i *is not present in the document randomly selected from the set R. Using Bayes' rule, and assuming independence of index terms, equation 7 can be written as:*
>
> $$sim(d_j,q) \sim \sum_{i=1}^{t} w_{i,q} \times w_{i,j} \times (\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{1 - P(k_i \mid \bar{R})}{P(k_i \mid \bar{R})}) \qquad (8)$$

Since R is not known a priori, it is necessary to make assumptions about their initial probabilities. This can be done by assuming that $P(k_i|R)$ is constant for all index terms k_i (typically 0.5) and that the distribution of index terms amongst the non-relevant documents can be approximated by the distribution of index terms among all the documents in the collection. These assumptions yield:

$$P(k_i \mid R) = 0.5 \qquad (9)$$

$$P(k_i \mid \overline{R}) = \frac{n_i}{N} \qquad (10)$$

where n_i is the number of documents containing the keyword k_i and N is the collection size.

After the initial ranking, a subset of the documents V is chosen as relevant. The probabilities are then recalculated based on V and V_i, the number of documents in V that contain the term *i*. The probabilities are then updated with the equations:

$$P(k_i \mid R) = \frac{V_i + 0.5}{V + 1} \qquad P(k_i \mid \bar{R}) = \frac{n_i - V_i + 0.5}{N - V + 1} \qquad (11)$$

The adjustment factors are used to eliminate the possibility of $P(k_i|\sim R)$ being zero, in the case that $V_i = n_i$.

The probabilistic model is attractive in that it is one of the few retrieval models that do not need additional term weighting algorithms to be implemented. For this reason, it has become one of the most influential retrieval models to date. The main disadvantages with the model, offered by Baeza-Yates and Ribeiro-Neto[1] include: (1) the need to guess the initial separation of documents into relevant and non-relevant sets, (2) the fact that the method does not take into account the frequency with which an index term occurs inside a document (i.e., all weights are binary), and (3) the adoption of the independence assumption for index terms.

2.1.4 Retrieval Evaluation

With the exponential growth of information sources comes the expectation of better performing information retrieval search engines. This expectation has focused the computer science community in improving existing information retrieval systems or creating new ones. The increased proliferation of these search engines has created the need to be able to compare one system to another using standard evaluation metrics. Unfortunately, there is no single method to do so, but instead a variety of differing techniques. In addition, different researchers use different terminology, which adds to the complexity of planning experiments and comparing the results. When all is said and done, and results are obtained, there are differing opinions as to what the results actually mean. For instance, one school of thought uses a retrieval performance evaluation technique, that evaluates a search engine against a test collection on predetermined queries and relevant documents. The assumption is that the set of relevant documents for a query is independent of the user. Another school of thought argues that user characteristics are important when evaluating what is a 'successful' search, and that the relationship between the search engine characteristics and user satisfaction must be better understood. The next section of this thesis looks at some current retrieval evaluation techniques, highlights some of the problems in using them to evaluate information retrieval systems, and discusses the current trends towards better evaluation methodologies.

2.1.4.1 Recall and Precision

Recall and precision is a relevance based evaluation measure that is generally considered the base line technique for evaluation of information retrieval systems. Recall is the fraction of all documents (the set R) considered relevant that are retrieved by the search engine. Precision is the fraction of the retrieved documents (the set A) that is relevant. This is defined by Baeza-Yates and Ribeiro-Neto[1] as:

$$recall = \frac{| Ra |}{| R |} \qquad (12)$$

$$precison = \frac{| Ra |}{| A |} \qquad (13)$$

where R is the set of all relevant documents, A is the document set returned by the retrieval strategy and R_a is the intersection of the sets R and A.

Recall and Precision assume that the documents in the answer set A have been examined by the user all at once, however this is usually not the case since they are presented in rank order of relevance. This means that the Recall and Precision metrics vary as the user proceeds with the evaluation. For this reason, proper evaluation requires plotting a precision versus recall plot for each distinct query. To evaluate the retrieval performance of all the test queries, average precision figures at each recall level can be taken to generate a single plot. A common practice is to use this plot to compare distinct retrieval algorithms.

A single plot to measure the overall quality of the answer set and the breadth of the retrieval algorithm is simple, intuitive and qualitatively useful. There are times, however, in which a single figure of merit for a single query may be more advantages than an averaging plot over all queries. For instance, it may be beneficial to know which queries perform best on given algorithms. For this reason, several techniques can be used to calculate single value summary figures. These techniques are listed and briefly described below:

- <u>Average Precision at Seen Relevant Documents</u>: Precision figures are averaged over all values obtained after each new relevant document is observed. This measure favors systems that return relevant documents quickly.
- <u>R-Precision</u>: Precision is calculated by the number of relevant documents retrieved at the R-th position in the ranking, where R is the total number of relevant documents for the current query.

- Precision Histograms: Measure of the difference between R-Precision values between two retrieval algorithms plotted over all queries under test.

- Summary Table Statistics: A table of statistical summaries such as the number of queries used in the task, total number of documents retrieved by all queries, total number of relevant documents effectively retrieved with all queries, and the like.

Although Precision and Recall merits are widely popular and have been used extensively for many evaluation measures, there are a number of drawbacks that have been identified. First, documents relevant to a particular query must be identified independently for Recall values to be calculated. This is not always feasible for large document collections and is usually restricted to smaller evaluation collections. Second, the recall and precision measures are inherently related by the union of the relevant document set and the answer set. Therefore, having two figures of merit here seems somewhat redundant and a single figure merit may be more appropriate. Third, an assumption is made that the relevance of the documents is independent of the user. This has two implications, that no two users will have differing views on the relevance of the material found in each document, and that interactivity of the user at each stage (searching through browsing) is ignored. Neither implication is particularly attractive. Fourth, recall and precision may not be appropriate for systems requiring a weak ordering.

2.1.4.2 Harmonic Mean

The harmonic mean F, [14] was developed in order to combine both the Recall and Precision values into a single metric. Let r(j) be the recall for the j-th document in the ranking, P(j) be the Precision for the j-th document in the ranking, then the harmonic mean of r(j) and P(j) is calculated as:

$$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}} \tag{14}$$

The function F has values in the interval [0,1] where 0 indicates no relevant documents have been retrieved and 1 that all the ranked documents are relevant.

2.1.4.3 E Measure

Similar to the harmonic mean, the E evaluation measure [15] attempts to combine the recall and precision values into a single metric. This time however, the user can weight the metric towards recall or precision by adjusting the value of a constant b. Values of b>1 indicates a preference to precision and values of b<1 indicate a preference to recall. Therefore, E(j), the E evaluation measure relative to r(j) and P(j) is calculated as:

$$F(j) = \frac{1+b^2}{\frac{b^2}{r(j)} + \frac{1}{P(j)}} \tag{15}$$

2.2 Interaction Models

The standard information access process assumes an interaction cycle consisting of a query specifying a need, the submittal and reformulation of that query to the system, the foraging for and receipt of the results, the examination and evaluation of the resultant set and the reformulation of the query to repeat the process until the perfect result set is found.

This interaction model is the only model currently used by search engines today. However, the model is simplistic in that it relies on an underlying assumption that the users information need is static and progressive refinement leads to only relevant information. It does not take into account that users in general dislike being confronted with a long disorganized list of retrieval results that do not address their information need and that users generally learn during the search process.

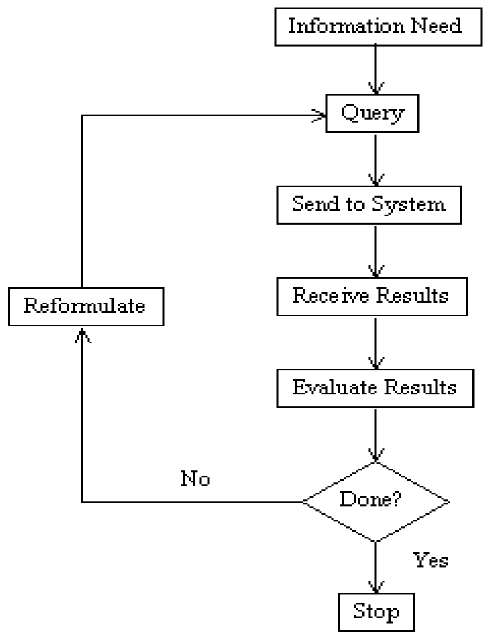


Figure 3. The Standard Model for Information Access[1:263]

The Standard Model is useful for describing the basics of information access systems, however, it's simplicity is being challenged as to whether it provides an accurate model at all. Observation studies [16][17][4] suggest that the information seeking process consists of a series of interconnected but diverse searches on one problem-based theme. While results for a goal tend to trigger new goals, and hence searches in new directions, the context of the problem and the previous searches are carried from one stage of search to the next. Additionally, it was found that the main value of the search resided in the accumulated learning and acquisition of information that occurred during the search process, rather than in the final results set.

The Sensemaking work of Russell et al [18] suggests that information work is a process in which information retrieval plays only a small part. The observation is that most of the effort made in Sensemaking is in the synthesis of a good representation (ways of thinking about the problem) and not the actual retrieval itself. O'Day and Jeffreis' study [4] suggested that a significant part of the information access revolved around finding trends, making comparisons, aggregating information, identifying a critical subset, assessing, and interpreting.

Card et al [19] introduces the concept of knowledge crystallization as a task in which one person gathers information for some purpose, makes sense of it by constructing a representational framework (or schema) and then packages it into some form for communication or action. This knowledge crystallization process is shown in Figure 4 below. As with the Standard Model, the process begins by defining a task and then gaining some insight into the data available to solve that task. Unlike the Standard Model which repeats this process iteratively, the knowledge crystallization process progresses to searching for and defining representational schemes about that data. If relevant data is found that does not fit the schema, then the search continues for a better schema. If the data is not relevant, it is omitted as part of the abstraction process. This continues until the instantiated schema defines the problem adequately or the user voluntarily terminates the search with the partial solution.

It is clear that the current Standard Model used for information access does not address these issues satisfactorily, yet all standard search engines today employ it as their underlying methodology. As the needs of the user become more complex and dynamic, and as the information sources become more diverse in media types, this assistance will not be adequate without some additional cognitive assistance to the user. The next section takes a look at some clever visualization techniques that are emerging as possible candidate solutions to the problems faced by today's techniques.

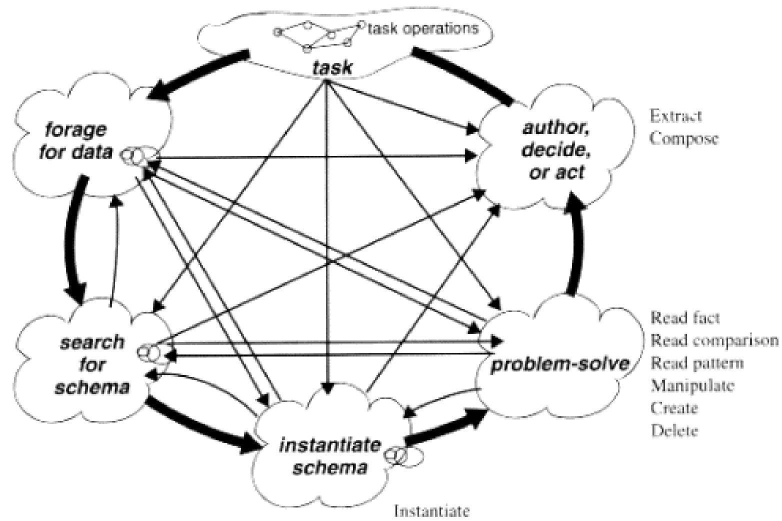


Figure 4. Knowledge Crystallization Process (Card, Mackinlay, and Shneiderman)

2.3 Visual Interfaces for Information Retrieval and Browsing

Typical information search interfaces provide a query interface, sometimes with the choice of specifying basic or complex queries, and an answer interface. The basic query interface allows the user to type in free text to specify a need whilst the advanced query interface allows the addition of boolean operators and other features, such as phrase search, proximity search, and wild cards. The answer interface usually consists of the top ten or so ranked documents that the search engine determines are the most relevant to the issued query.

This technique allows foraging of data through search queries and browsing, however, very little cognitive amplification occurs for the user. If the relevant document set returned is not meaningful to the user, then they must begin again by issuing another query with almost no more prompting than before. An improvement to this technique is the addition of relevance marking, in which the user can select a document to be relevant (and perhaps others as not) or issue further queries based on the precision of a particular document. This helps in data foraging but it does not complete the knowledge crystallization process found to be essential in the knowledge search process of the user.

A relatively recent approach to improving the effectiveness of information retrieval from large information systems has been to address the interaction between the system and the user. Graphical user interfaces are becoming an important part of document retrieval systems. Rather than just using an index matching algorithm to select (as precisely as possible) those documents of interest to the user and listing these documents in sequential order, the role of the graphical interface is to organize and present this information in a way that helps the user select from them. Card et al [19] state that the purpose of information visualization is to use perception to amplify cognition. Several approaches to visualization in information retrieval are techniques that map a multidimensional document similarity space into fewer dimensions. The key here is to allow users to explore a document similarity space with few enough dimensions that it can be visualized directly. Such interfaces are intended to aid users who are unfamiliar with the scope or semantics of the document collection, or who are unable to express their information requirements appropriately to retrieve relevant documents. Furthermore, these interfaces may address one or both of the following weaknesses:

- Although lists of documents can be ranked according to their similarity to a query, a document's position in the list provides few clues to the positions of related and unrelated documents.

- Computed similarities are summary measures, and therefore reveal almost none of the underlying document representation. Users can not tell why a document was ranked high or low without retrieving the document representation.

Visualization techniques are emerging as possible candidate solutions to the problems faced by today's information retrieval techniques. TileBars[20] and SeeSoft[21] use a "query term hits within document content" approach [1]. This essentially highlights the occurrences of the terms or descriptors that match those of the user's query within the document's content. VisDB[22] sorts the documents based on their relevance with respect to the query and then maps the relevance factors to colors. Techniques such as VIBE[23], Lyberworld[24] and InfoCrystals[25] use a "query term hits between documents" approach [1]. This approach shows an overview or summary of the retrieved documents according to which subset of query terms they contain. The following section provides an overview and discussion of the techniques that contributed to this research. For completeness, other techniques are discussed at the end of the section in less detail.

2.3.1 VIBE and Lyberworld

VIBE and Lyberworld are different packages, however, Lyberworld presents a 3D version of the same concepts as VIBE. Therefore, the following paragraphs discuss VIBE with the caveat that many of the issues discussed also relate to Lyberworld.

The VIBE (Visual Information Browsing Environment) System was developed by Kai Olsen of Molde College from a technique researched by the University of Pittsburgh. It is a tool used for visualizing multivariate data such as records in a relational database. In VIBE, records are defined in terms of their degree of association to various Points of Interest (POI). A POI can be any attribute or characteristic to which a numeric association strength can be assigned. For example, the records of a traditional relational database could be displayed in VIBE by defining the numeric fields as POIs, and the association strengths of the records as the values of those fields. A POI can also be a key word or index term for a set of documents, with the association strength for a particular document being the weight of the term in the document (as defined by a term weighting formula or simple term count). A POI can even be a more complex object, such as an entire document, a query, or a user profile. In these cases, the degree of association of a record to a POI must be calculated by some method such as a coefficient of similarity. The VIBE display, shown in Figure 5, is dynamic. The POI icons (displayed as circles) can be moved freely by the user of the system. After each move, the system plots each record icon (the rectangles) so that its position on the display with respect to the POIs is proportional to the relative strengths of the association of the record to the POIs. The size of the rectangles represents an average association over all POIs, since the icon positions depend only on relative, not absolute, association strengths. VIBE itself has little or nothing to say about how documents must be represented, how the degrees of association are calculated, how optimal POIs are identified, and how the display is to be interpreted by a user.

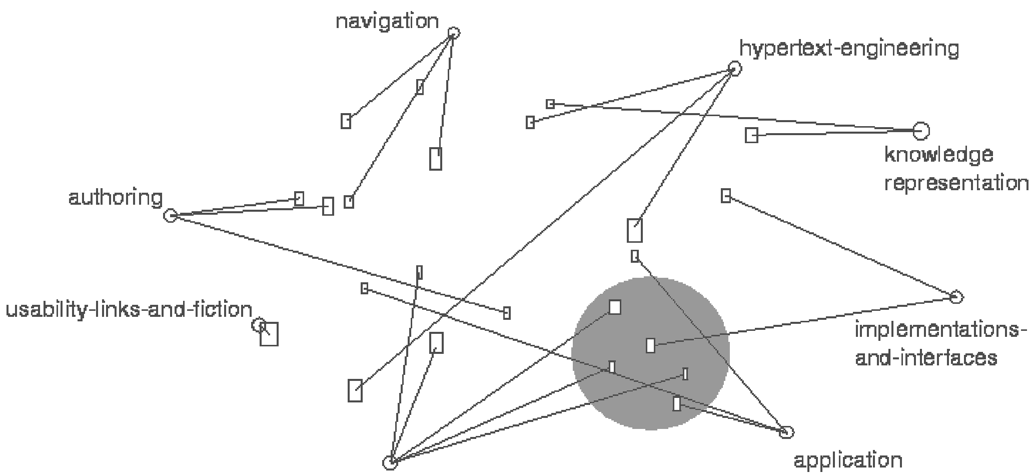


Figure 5. VIBE Graphical Interface

2.3.2 WebVIBE

One of the criticisms with VIBE is the relative complexity of the interface. WebVIBE[26], a derivative of VIBE, was developed by scaling down the functionality of VIBE to provide a user friendly environment. It was designed for the walk-up user, hence no assumptions on the knowledge of the user are assumed besides how to use the World Wide Web with some sort of computer and mouse. The interface, shown in Figure 6, is designed to require little to no instructions to understand it. With this in mind, the use of metaphors with real world “gadgets” is an important aspect of the design.

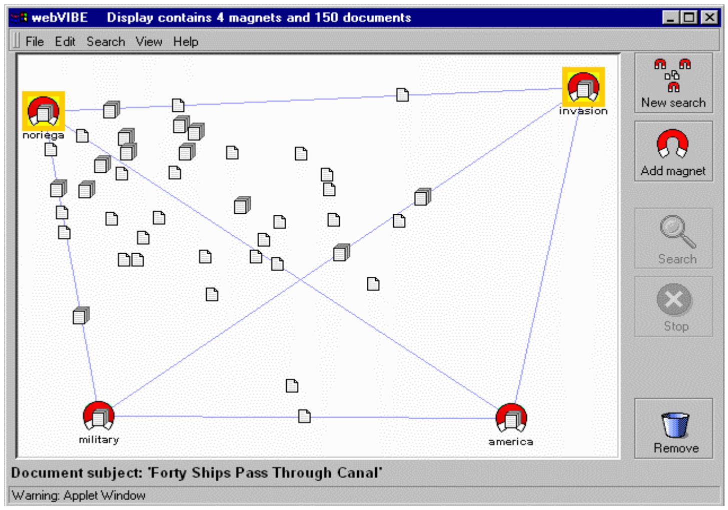


Figure 6. webVIBE Graphical Interface

Overall, webVIBE is designed to be perceived as a natural part of the user's environment with which the user would have to interact. The user is given access to spatial display information, which gives them greater meaning to the relationships of key words.

2.3.3 **InfoCrystal**

The InfoCrystal[25] interface, designed by Anselm Spoerri at MIT and shown in Figure 7, allows users to specify boolean queries graphically, to visualize boolean combinations of query terms, and to see the sizes of the document subsets which satisfy those term combinations. The graphical components of the system are iconic displays known as query spreadsheets, which function similarly to Venn diagrams: icons on the spreadsheet represent different boolean combinations of **n** query terms. The position and shape of the icons indicate which query terms that are represented and how they are combined. Numbers printed inside the icons specify the size of the sets which satisfy the boolean expression, and these document sets can be retrieved by selecting the appropriate icon. InfoCrystal also includes an outlining tool, used to nest query spreadsheets in complex, hierarchical combinations.

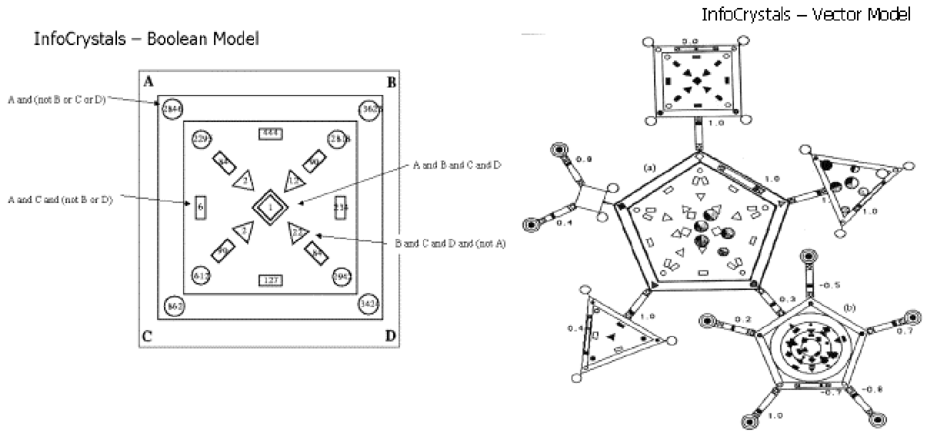


Figure 7. InfoCrystal Graphical Interface

Some additional coding includes:

- Rank coding. Icons with the same shape are grouped in "invisible" concentric circles where the rank of an icon is equal to the number of criteria satisfied and it increases as it move towards the center of an InfoCrystal.

- Orientation: The icons are positioned so that their sides face the criteria they satisfy.

- Size or Brightness and Saturation Coding: Used to visualize quantitative information such as the number of document hits.

2.3.4 Other Visualization Applications

2.3.4.1 Scatter/Gather

Scatter/Gather[27], shown in Figure 8, is a cluster-based document browsing method, which divides the retrieved documents into five clusters, and lists the common keywords for the articles at the top. By examining these keywords, and the number of documents within each cluster, the user can determine which cluster represents his need more precisely.

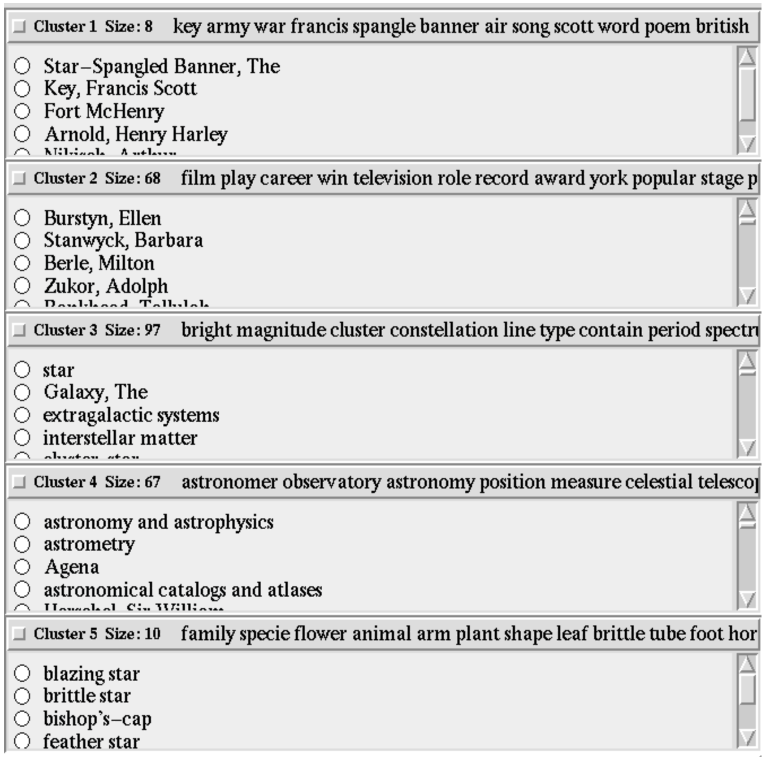


Figure 8. Scatter/Gather Interface

2.3.4.2 TileBars

TileBars [28], shown in Figure 9, requires queries to be entered into separate entries by terms representing one topic. The query set is treated as a conjunction of topics that are listed in order of importance. A statistical segmentation algorithm called TextTiling is used to subdivide documents into multi-paragraph sub-topical units, which creates meaningful segments over average segment lengths. Using this visualization technique, the user can now quickly see if subsets of terms overlap in the same segment of the document indicating well represented termsets and what position in the document this occurs. By displaying intermediate relationships between the query and the retrieved documents, rather than the document descriptions, the technique emphasizes participation of the user in the cycle of query formulation.

Figure 9. TileBars Interface

2.3.4.3 Tkinq

Tking [29], shown in Figure 10, is a system for querying, navigating and visualizing an on-line library catalog. The system displays the presence or absence of significant keywords across the document set using a parallel coordinate technique. Each keyword is assigned a row of bars, with the height of the bar representing the document's relevance rank for that keyword across the document set. The last row presents the total rank for each document ordered from left to right in decreasing order of relevance.

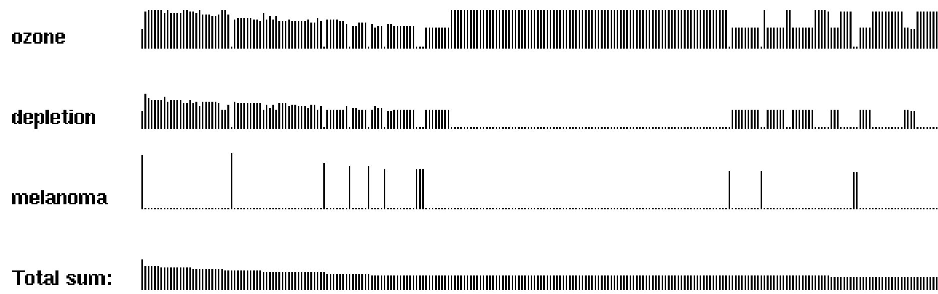


Figure 10. Tkinq Interface

2.3.4.4 Bead

Bead [30], shown in Figure 11, represents articles in a bibliography as particles in 3-space. The 'particles' are spatially organized into position depending on the inter-relationships between them. Similar particles are attracted to each other, and dissimilar particles are repelled. This results in a three dimensional scene, which can be used to visualize patterns amongst the documents in higher dimensionality.

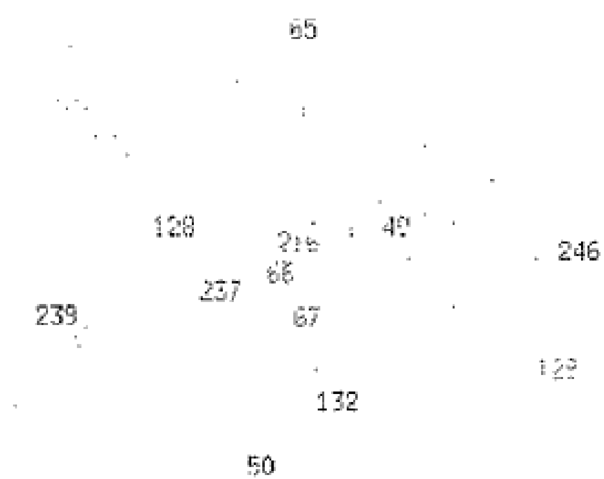


Figure 11. Bead Interface

2.3.4.5 Narcissus

Narcissus [31], shown in Figure 12, is a self organizing system that generates a three dimensional visualization environment through a process which communicates with the application (such as web browsers and programming environments). Each object in the information space is given spatial positioning based on the attraction and repulsion of other objects in the space. Objects therefore migrate through space so that they are spatially close to other objects (which are semantically similar) and connected through active relationships. Users can navigate through the virtual environment and manipulate the objects to make the visualization clearer.

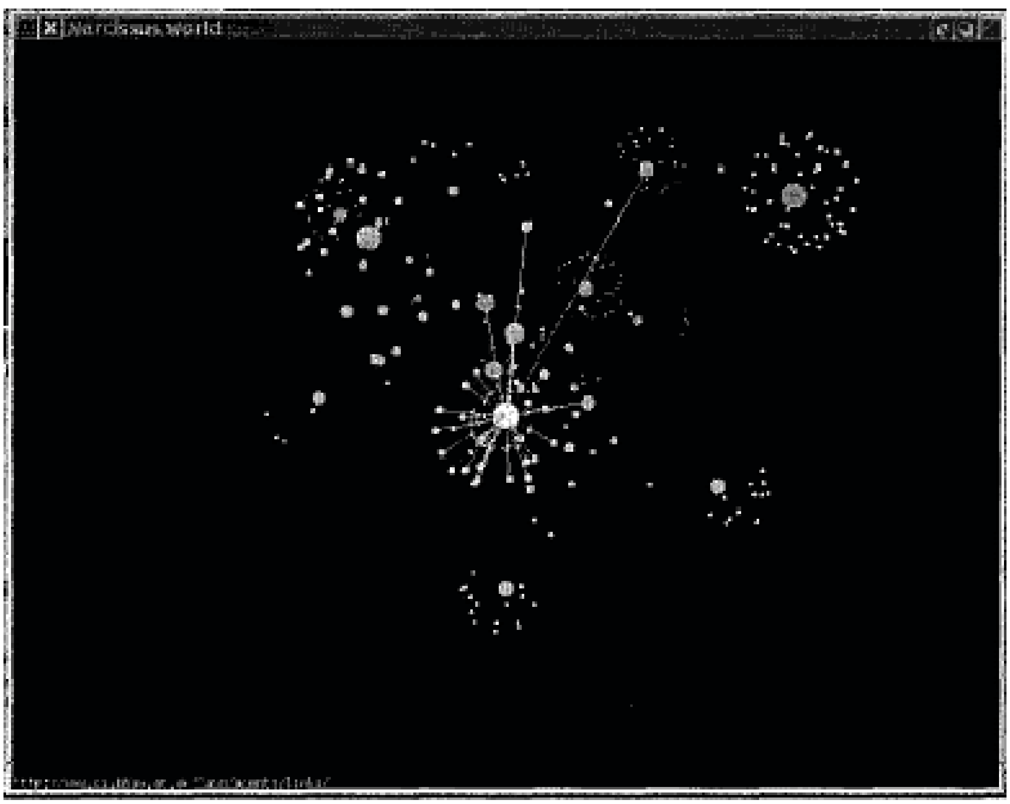


Figure 12. Narcissus Interface

2.3.4.6 Cat-a-Cone

The Cat-a-Cone system [32], shown in Figure 13, integrates search and browsing of very large category hierarchies with their associated text collection. The system works on the principle of separating the graphical representation of the category hierarchies from the graphical representation of the documents. This provides an interaction between browsing and searching by associating documents with corresponding categories and hierarchical context. The retrieved documents are stored in a book representation along with a page of links to the category hierarchy and document contents.

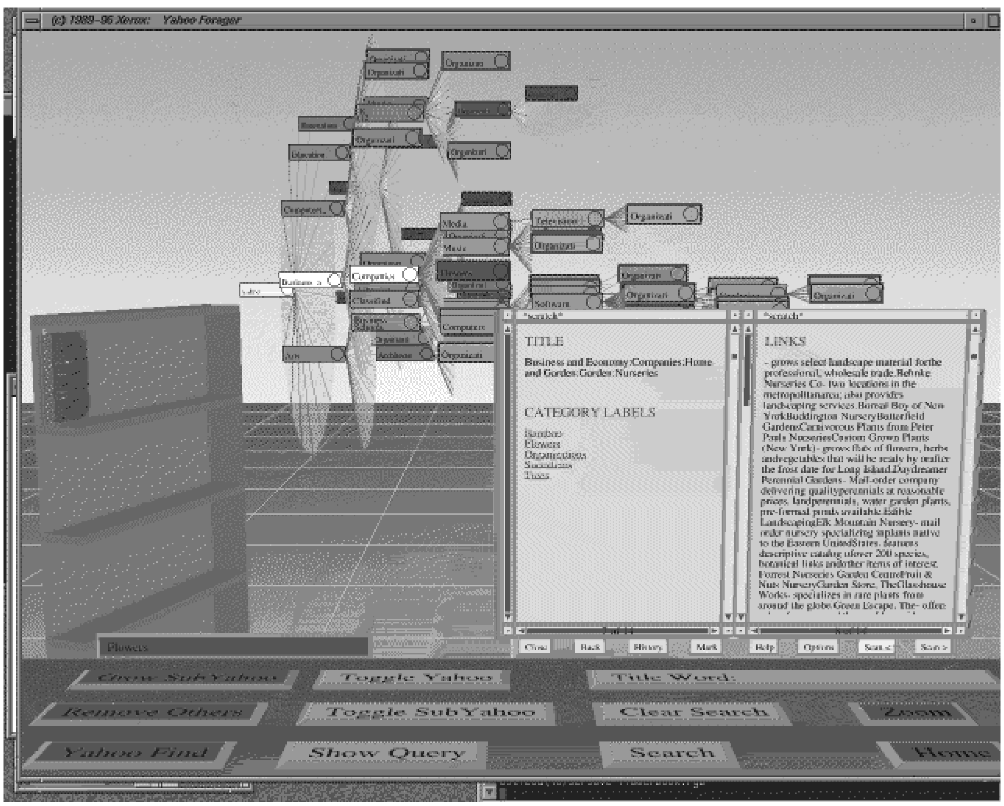


Figure 13. Cat-a-Cone Interface

2.3.4.7 Envision

Envision [33], shown in Figure 14, represents the document space, retrieved from a search, as a set of icons organized in a matrix representation along with an item summary showing a textual listing of bibliographic information for selected icons. This graphical representation supports users in deciding which works to view depending on user controllable attributes and physical positioning in the matrix. Icons may represent single documents or groups of documents representing similar attributes.

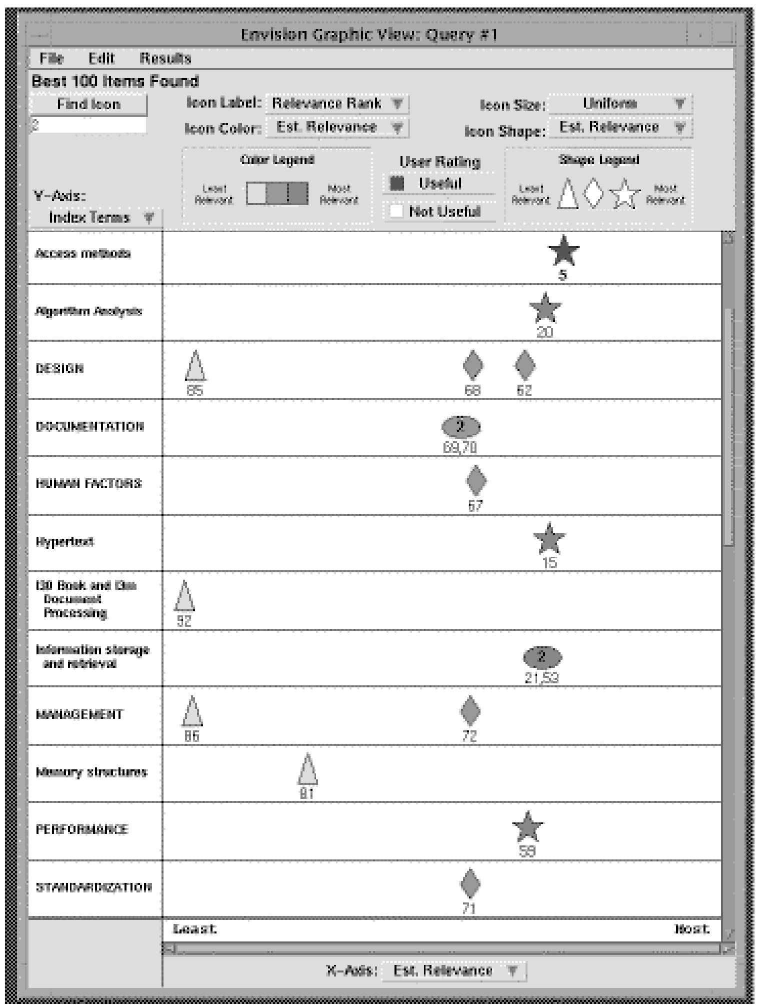


Figure 14. Envision Interface

2.3.4.8 LightHouse

LightHouse [34], shown in Figure 15, combines traditional ranked lists with clustered visualization. Documents are visualized as spheres and positioned in two dimensional space proportionally to the inter-document similarity. Therefore, semantically similar spheres are located close together and dissimilar spheres are located far apart. Ranked lists are used in a traditional way to display how similar the documents are to the original query. The combination of the two techniques provides additional information on how documents are related to each other.

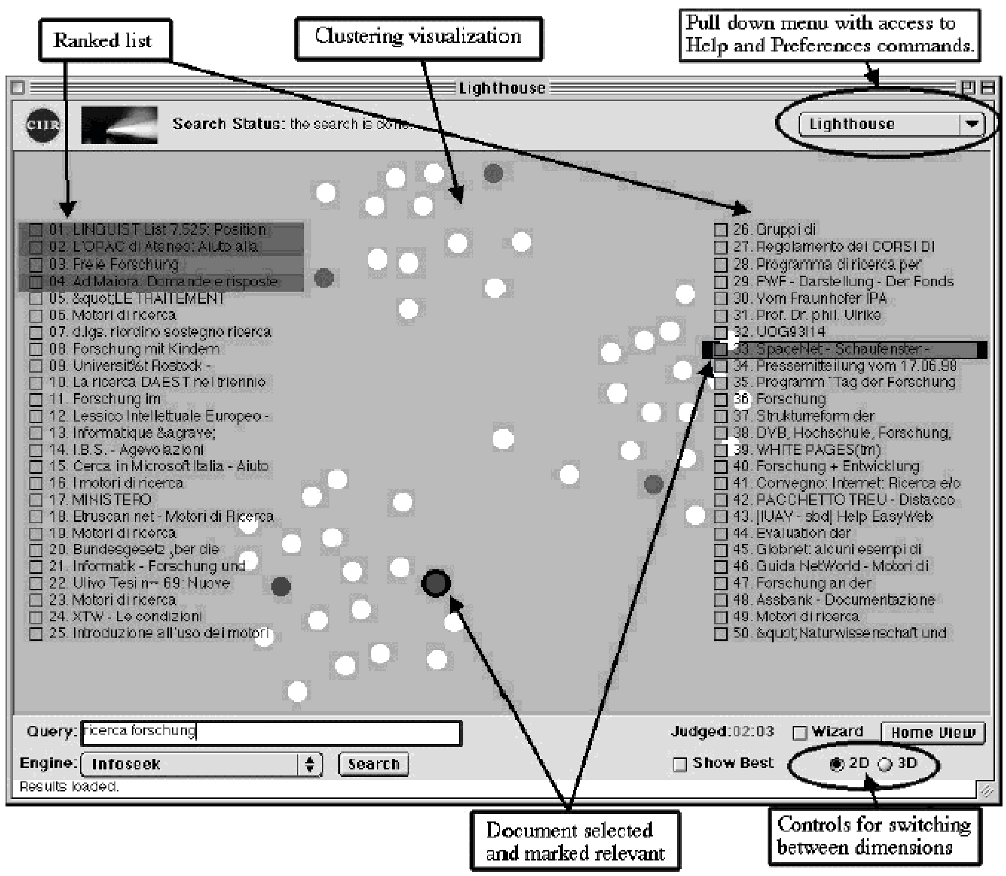


Figure 15. LightHouse Interface

III. Framework Analysis

In Chapter 2, background information on information retrieval techniques and systems was presented to help understand the fundamentals of the problem of this thesis. In this chapter, the emphasis moves away from these techniques and concentrates on the human cognitive side of information retrieval. The purpose for this is to build a framework of conceptual design issues and design corollaries that will guide the design methodology presented in Chapter 4. Therefore, this chapter is presented in the following manner. Section 3.1 investigates the high level cognitive strategies that are important in visual information retrieval and ends with a lower level summary of these propositions. Section 3.2 takes a different approach and examines a case study on the usage of an online information retrieval system. From this usage analysis, eleven design corollaries are formulated which builds the requirements framework needed for Chapter 4.

3.1 Cognitive Strategies

There is an overriding intuition in graphical interfaces that more is better. This has led to a long line of assumptions about technological advancements in visual systems that claim to provide greater assistance in the cognitive tasks. Some of these assumptions are listed in [35] as:

- Static pictures and diagrams are better than sentential representations.
- Three-dimensional representations are better than two-dimensional ones.
- Solid modeling is better than wire-frame modeling.
- Color is better than black and white images.
- Animated diagrams are more effective than static images.
- Interactive graphics are better than non-interactive graphics.
- Virtual reality is better than animation.

Are these valid assumptions? Generalizations about advanced graphical techniques over static or textual representations requires an understanding of the cognitive gain in providing more explicit, dynamic or interactive representations of the information, however, these cannot be assessed adequately from intuition alone. To be effective requires a number of independent factors to be considered: the level of experience the user has with the graphical representation, the knowledge domain, and the type of task. This suggests that evaluation of the merits of each requires a theoretically driven analytic framework that takes into account the cognitive processing that takes place when users interact with them. This is not well understood and accounts for why many visualization techniques have performed poorly in 'real world' situations. Molitor et al. [36] points out that a large number of studies have been concerned with the manipulation of the graphical representation for highly specific situation, and success or failure of the graphical representation is reported on how performance was affected. Winn [37] notes that it is difficult to generalize this precisely because of the idiosyncrasies involved with each graphical representation. Although these statements are not directed at visual information retrieval systems per se, the points remain valid. Studies on the effectiveness of visual systems, such as [38] have failed to provide support to claims that visual interfaces will outperform current traditional techniques.

This lack of graphical representation research on search strategies may be one reason why many visual information retrieval systems lack a solid grounding in cognitive processing theory, and why much of the empirical work has been centered on evaluation of ad hoc task specific questions. The authors of [35] advocate an alternative approach:

> *We need to ask what is the nature of the relationship between graphical representations and internal representations and to consider how graphical representations are used when learning, solving problems and making inferences. Such an enterprise means working towards a detailed description of cognitive mechanisms.*

Cognitive sciences advocate a deeper understanding of the relationships between the internal representations of visualizations, “knowledge in the head” [39][40], and external representations of visualizations, “knowledge in the world”. Vera and Simon [41] stress that, “A fundamental problem for cognitive modelers is to interleave internal and external states in order to achieve naturalistic behavior.” This is a fundamental change in the way cognitive modeling is now being approached and is of particular importance here since traditional techniques have always been centered around adapting internal modeling processes to the external representations. In the words of [39][40], the “knowledge in the head” has never mapped naturally to the “knowledge in the world”.

3.1.1 Cognitive Science in Visual Information Retrieval

The question now is how to bring a fundamentally new approach to the interaction of internal and external representations into the field of information retrieval. To begin with, it is necessary to understand the basic framework of *external cognition [35*]. Scaife and Rogers list four central characteristics from which to explicate aspects of external cognition:

- *Computational offloading* - refers to the extent to which different external representations reduce the amount of cognitive effort required to solve informationally equivalent problems.
- *Re-representation* - refers to how different external representations, that have the same abstract structure, make problem-solving easier or more difficult.
- *Graphical constraining* - refers to the way graphical elements in a graphical representation are able to constrain the kinds of inferences that can be made about the underlying represented world.

- *Temporal and spatial constraining* – refers to the way different representations can make relevant aspects of processes and events more salient when distributed over time and space.

Computational offloading is directed at the efficiency of solving a particular problem through direct perceptual recognition. If the perceptual models used are such that they must first be mapped to internal representations, then the cognitive effort required is much greater than those that are mapped directly. The use of well-known metaphors is an example of external representations that provide greater mental efficiency through computation offloading. This plays an important role in the selection of not only the graphical representations, but also the associated interaction and manipulation models.

Re-representation is a direct result of the efficiency of computational offloading. Representations that map closely to internal mental models are likely to provide greater efficiency in the cognitive processing functions than those that are not, even though they essentially have the same abstract structure. Zhang and Norman [42] provide an example of this by describing the process of multiplication using Roman or Arabic numerals. Both represent the same formal structure, and both could be used to provide the same solution. For many, however, the decimal system presents itself as an easier representation for manipulation. The difficulty is in knowing the target audience and then making the correct assumptions as to which representations work efficiently, and which do not. The idea is simple but the execution is enormously difficult. Can end-users be grouped in an appropriate manner to facilitate this process and if so, what effect does this have when they cannot?

Graphical constraining represents how the relationships between the graphical elements and their representations map onto the problem space in such a way as to restrict (or enforce) the interpretations that can be made. The aim here is to provide a tight coupling between the visual display and the information retrieval domain space to maintain tractable inference.

Temporal and spatial constraining of the external representations affects the cognitive processes of mental comparison and evaluation, shifts in visual attention, mental simulation and the appropriateness of proportional reasoning. This is a critical characteristic in visual information retrieval systems that attempt to constrain the temporal and spatial domains to further enhance the ability of end-users to make correct judgments and inferences about the external representation.

3.1.1.1 The use of diagrams

Diagrams are a natural evolution from sentential displays for a number of reasons. First, they provide simultaneous location information of objects and the inherent relationships between them, which can then be easily tracked and maintained. Second, they provide additional information necessary for the solution of search tasks. This is generally not so with sentential displays, which do not have the same external memory cues and must therefore force the user to formulate this additional information explicitly. Third, diagrams have been shown to be effective in maintaining a sense of how the problem is progressing. This too can be done with sentential displays but again at the cost of explicitly formulating additional information, which results in additional mental processing to maintain the same perceptions on the possible states of the problem.

Not all diagrams are equally effective in a particular situation. For example, Stenning and Tobin [43] claim that Euler's Circles are more effective than three dimensional cube diagrams in helping subjects solve logic problems because the geometrical constraints of the intersecting circles represent the logical constraints much better. In this example, the circles represent an effective use of graphical constraint, which limits the possible interpretations of the problem thereby guiding towards the correct solution. This was identified by Stenning's and Oberlander's [44] theory of specificity, which postulates, "graphical representations such as diagrams limit abstraction and thereby aid processibility". This leads to a fundamental point that diagrams are likely to be more effective in presenting a particular problem and a way of solving it than sentential displays because they are less expressive and reduce indeterminacy.

The implication now is that external representations can, and do, explicitly change the nature of the task through graphical constraints. By limiting the permissible states of the problem, through implicitly embedded rules, a form of cognitive offloading is achieved which allows greater internal memory to be used to focus on the next state of the search process. For this to be true, however, two factors are crucial in the construction of the external representation: the notation (symbols, icons) and the visual organization that structures them. The notation should be appropriate, intuitive, and easily understood to reduce the need to translate between external and internal representations. The structure should use canonical forms that activate recognizable 'readability rules' [45] by cueing appropriate kinds of inferences in the reader.

3.1.1.2 Conceptual Design Issues

The work of Scaife and Rogers [35] can be summarized into five conceptual design issues for graphical representations, of which four apply equally well for visual information retrieval systems. The applicable conceptual design issues, modified for use here, are listed as follows:

- Explicitness and visibility – facilitate perceptual parsing and inference, by directing attention to key components that are useful or essential for different stages of a problem-solving or learning task.

- Cognitive tracing and interactivity – support different kinds of cognitive tracing (highlights or mark-ups) and levels of interactivity to test new configurations.

- Ease of production – provide easy and intuitive creation or reproduction of external representations through readily accessible support tools.

- Combining external representations – use multiple external representations, such as static diagrams and text, to enhance the cognitive process.

3.1.2 Cognitive Propositions

Section 3.1.1 outlined the cognitive strategies that are important in shaping the design methodology in Chapter 4. For clarity, the important issues from this section are summarized below in the following twelve cognitive propositions:

Restrict the interpretations of the external model to enforce a tight coupling between it and the information retrieval domain space.

Proposition 1: Use two-dimensional static displays to provide external memory cues while maintaining graphical constraint.

Proposition 2: Maintain intuitive relationships between the graphical elements

Provide computational offloading through the careful selection of graphical representations and the associated interaction and manipulation models.

Proposition 3: Use carefully selected metaphors.

Proposition 4: Provide a natural interaction model within the metaphoric environment.

Proposition 5: Support different kinds of cognitive tracing (highlights or mark-ups).

Proposition 6: Provide interactivity to test new configurations.

Proposition 7: Provide easy and intuitive creation or reproduction of external representations.

Provide a graphical representation of the problem that maps closely to the internal mental models of most users.

Proposition 8: Use graphic representations (symbols, icons and visual organizations) intuitive to most users.

Proposition 9: Use multiple external representations (static diagrams, text) to enhance the cognitive process.

Constrain the temporal and spatial domains to enhance the correct inferences about the external representations.

Proposition 10: Constrain visual attention shifts.

Proposition 11: Provide a tight couple between the temporal and spatial domain to maintain consistent and progressive inferences about the representation.

Proposition 12: Direct attention to key components that are useful or essential for different stages of the retrieval process.

3.2 Design through usage analysis

In the design and evaluation of any query interface, it is important to have a clear understanding of the usage pattern expected and then design for the typical case. One method of conducting usage analysis is through transaction logs where predetermined usage metrics are collected over a period of time and then extracted for further analysis. This information on user behavior can be extracted automatically (through summary statistics) or manually (through text queries). The former method allows for overall statistical results to guide certain usage assumptions while the later allows for detailed examination of the semantic clues behind search motivations and search strategies. Both are important to gain some insight into typical usage patterns and the motivation behind them.

Although there is extensive literature on transaction log analysis of online public access catalogs (OPACs), these techniques have only recently been applied to digital libraries. The reason for this is that digital libraries are only just attaining usage levels suitable for log analysis[46][47][48]. Typically, transaction log analysis is conducted over a relatively short time period (such as a single day) and is based on a non subject-specific collection of documents. This tends to produce a diversity of results that are difficult to generalize for all users at all times. Of more use here is the study conducted by [49], which differed in two significant ways. First, the study was conducted over one years use, and second, the study was conducted on a subject-specific collection, computer science technical reports. This focus, both on the collection and the demographics of its users, resulted in finer-grained details that can be used to base the design criteria on. The caveat is that the study may be too specific to be used in a general case. The study does indicate, however, that although these users may be considered "best case" users of online search engines, given their familiarity with software and boolean logic, they experienced many of the same difficulties in searching and dealing with query languages that are reported for the general public.

3.2.1 Design Corollaries

The following section is dedicated to developing design corollaries based on the usage analysis experiment conducted in [49]. These design corollaries will formulate the second part of the analysis framework used to guide the design phase of this research outlined in Chapter 4. This section is organized by particular outcomes that resulted from this experiment. Each outcome is structured with a conclusion stated and a paragraph that summarizes the author's findings that lead to this conclusion. The paragraphs that follow the author's conclusions describe how the particular outcome relates to this research and then, the appropriate design corollaries are formulated.

3.2.1.1 Default Settings

Users rarely amend default settings.

The study indicated that given the opportunity to change default settings to tailor the search to an individual, the user was not likely to do so. The user had the ability to change the search model used, the proximity setting, case-sensitivity, stemming, and default result set size. Although evidence of some tailoring did occur, the occurrence of such action was not frequent.

Given that the target audience were 'experienced users', it is unlikely that this phenomenon would be more frequent for a different select group.

This leads to the interpretation that users tend to accept whatever the defaults were set to. For this reason, some statistical analysis may be required to determine what the default settings should be for a particular target audience. Simply giving the flexibility to the user to change them is not effective and should be avoided as a primary means of tailoring the search. This leads to the first corollary:

Corollary 1: The default setting should be set a priori and determined statistically based on user profiling.

3.2.1.2 Query Complexity

Queries tend to be short and simple

The average number of search terms over all the queries was found to be 2.43. Of this, approximately 80% contained three or fewer terms and 98% contained six or fewer terms. Additionally, users were unwilling to apply boolean logic to the search, possibly due to the added complexity or restrictive nature of the query language.

This is consistent with other findings that suggest users tend to submit ad-hoc queries of relatively few search terms [50][51]. An interesting point to note here is that TREC competitions, such as [52], tend to evaluate retrieval engines with long queries (50-85 words) which provide higher overall results because they are far more descriptive, however, they are not consistent with typical usage patterns. Of note here is that any visual interface employed need not be constrained to facilitating large and complex queries, instead it should be optimized around providing short and simple query specifications.

Corollary 2: Optimize the interface around short and simple query specifications.

3.2.1.3 Query Terms

Clusters of common query terms are apparent.

The study revealed that clusters of content bearing query terms became apparent and accounted for 13.8% of the terms used. This was consistent with other studies on web search engines and digital libraries, which also noted similar clusters of common search terms. The study indicated that many of these terms appeared as portions of phrases in the queries (such as "information retrieval"), which if used in isolation, would lead to false drops. The observation is made that phrase searching should be an integral part of the search tool and that users should be supported in constructing the proper phrases.

This leads to some important issues. First, the clusters of common search words are significant and have the potential to play a major role in the search. These common words, however, are domain specific and must first be identified for each target group of users. Doing this well is a significant task in its self. Second, the terms tended to make up common phrases but were not submitted properly (delimited by quotes) indicating that the tool must provide the ability to seamlessly allow phrase specification and be able to efficiently process those phrases. Third, knowledge of the common query terms opens up avenues for exploitation such as caching document abstracts, thematic coding of the query terms into categories and/or introducing predefined structure into the query.

Corollary 3: Identify the domain specific clusters of common search terms.

Corollary 4: Allow seamless phrase construction and efficient phrase processing.

Corollary 5: Use methods to exploit the most frequently used query terms.

3.2.1.4 Term Specificity

Users have difficulty selecting terms with the appropriate specificity.

The query terms themselves were correlated against the documents in the expanded collection. It was found that about one in twenty terms did not match any of the documents returned, most of which were due to spelling mistakes and the rest due to personal or product names or simply legitimate words that did not appear. At the other end of the scale, over 40% of the query terms were matched by 1000 or more documents with 3% of the query terms matched by over half of the collection. Therefore, the difficulty for the searcher was not that the terms could not be found, but that the terms were not selected with the appropriate specificity. This is symptomatic of the fact that users tend to specify the query in terms of broadly defining short queries, which tends not to be descriptive enough to narrow down the resultant collection.

Given the reluctance of users to expend more time in the query formulation, the question becomes whether overall results can be improved while still maintaining short query specifications. Relevance feedback [53] and automatic term expansion [54] are two methods proven to remedy this problem to some degree. Another interesting technique is the use of two simple constraints [55] specifying the query as a list of topics (converted into a conjunct of disjuncts) and imposing a subtopic-sized proximity constraint over the boolean constraint. In any case, term specificity is a significant problem and should be addressed by the search tool.

Corollary 6: Provide automated assistance to improve term specificity.

3.2.1.5 Erroneous Queries

A high percentage of malformed or erroneous queries are submitted.

The study defined a failed search as one that matched no documents in the collection (or zero hits). Although the converse is not true, the 'failed' search as defined, does identify the extreme case of an unsuccessful search. The study concluded that 10.5% of all submitted queries returned zero hits, with an average of 2.4 terms for the boolean query and 1.2 terms for the Ranked query. Of the ranked queries that returned zero hits, approximately 73% of them contained a quoted phrase. This indicates that a phrase search for the ranked query is less likely to be matched than an unquoted set of terms. Note that the experiment removed syntactically incorrect queries prior to being entered into the transaction log, so these were not considered.

There are some points worth mentioning here. First, no attempt was made to measure ‘failure’ at the satisfaction or motivational level of the user. Instead, the experiment followed the definition of previous studies to measure a failed search as ‘zero hits’. So in essence, the measure is not of semantic or syntactic correctness of the query, but rather a measure of the application specific rules for structuring queries. Second, as mentioned previously, the low term count of the query again indicates a correlation between poor specificity of the query and a failed search. This leads back to corollary 6 defined above. Third, phrase searching results in a very specific search. The fact that no hits are found on a particular phrase may not always be indicative of a ‘failure’, but rather a refinement process to ultimately a successful search. Therefore corollary 4 above remains valid.

This discussion does raise an important point, which leads into the next corollary. A significant proportion of submitted queries are unsuccessful due to poor user understanding, or familiarity, with the search application. This can be alleviated in a combination of two ways: provide user training, which is not always applicable, and/or design the application to be inherently intuitive and user friendly. The second point refers to mapping the external representation successfully from the internal representation to reduce the cognitive translation that needs to occur. This extension should allow a more natural interaction model and therefore reduce the element of unfamiliarity, which leads to the problems discussed here.

Corollary 7: Provide interaction models that naturally extend internal representations thereby reducing cognitive overloading.

3.2.1.6 User Sessions

Users submit few sequences of queries, spend short periods searching and are unlikely to repeat the visit.

The results of the usage patterns of users on the NZDL library are interesting. The study revealed that one fifth (21.51%) of the user sessions did not submit a query. Just over half of the sessions (51.68%) submitted only one or two queries. Slightly more than one fifth (21.69%) submitted three to six queries and 5.12% submitted seven or more queries. The average number of queries issued per session was 2.04. The length of the session revealed that 29.16% lasted less than one minute, 54.34% lasted five or fewer minutes and 66.43% lasted ten minutes or less. The number of repeat visitors to the site amounted to only one quarter of the total user base, indicating that three quarters of the users, who no doubt contributed to a significant proportion of the one or fewer query submittal, did not return over the full year of testing.

This has some interesting points and alarming consequences. First though, it is important to note that these results will differ from system to system especially between web and non-web based applications. It is expected that motivational trends in users of other applications would follow typical usage patterns such as these, so these results are useful here in providing general guidelines.

The first result indicates that not only do users not expend time in the development of queries, but they also do not expend time in refining the query to provide better results. In fact, the most common scenario was the submittal of a few short queries resulting in no documents being reviewed. This suggests that many users abandon their query process prior to having their information need satisfied. The second set of results reveals that users make rapid judgments as to the usefulness of the retrieval system. If the attention of a user cannot be maintained through effective retrieval, then users are likely to terminate the search prematurely. The last result indicates that not only is the window of opportunity to fulfill the needs of the user very short but once a user leaves, he is not likely to return.

These general usage patterns provide some useful indicators to apply to this research. What is needed here is an application capable of exploring search schemas in an automatic or semi-automatic fashion that is triggered on simple and concise queries. The underlying reasoning here is to open up as much of the collection space as possible, quickly and concisely, with minimal input from the user. This will promote further interest in the search and hopefully lead to additional exploration by the user. Further, the query reduction and expansion must be achieved intuitively, unobtrusively and almost as a natural extension of the interaction with the interface. Done effectively, the retrieval application may provide useful information almost immediately, which should help address some of the observations mentioned above. This leads to the next corollary.

Corollary 8: Exploit automatic and semi-automatic schema expansion and allow for exploration and substantiation of such schemas.

3.2.1.7 Query Refinement

Users refine their query.

An analysis of consecutive queries reveals that the majority (66.37%) of the queries issued by users had at least one term (a word or a phrase) in common with the previous query. Most often, queries had one or two terms in common (22.56% and 23.08%), some had three terms in common (11.34%) and 9.39% had four or more terms in common. The authors of the experiment go on to say that given that the average number of terms within a query is 2.5, and only a fifth of queries contain four or more terms, it is likely that the query refinement occurs in small incremental steps. Users will tend to make minor changes by adding a new term, or altering the existing terms.

So effective and efficient query refinement needs to be addressed. The refinement process, both restrictive and widening, need not be a complex autonomous operation but rather a mutable, incremental and basic one. Importantly, these changes are a common activity and simplicity of change will be the key to efficiency.

Corollary 9: Allow mutable, incremental and basic query refinement.

3.2.1.8 Result Viewing

Few, if any, documents are viewed from the returned list and those that are can be found at the top of the list.

Interestingly, the majority (64.2%) of the queries did not lead to users viewing document content, 19% resulted in one document being viewed, 12.7% viewed two, three or four documents and only 5% viewed five or more from the list. The study went on to investigate the length of the queries of those with documents viewed against those with none viewed and it was found that the length of the query was not necessarily a factor in whether users viewed any of the results documents. When users did view documents, 12.7% of the documents viewed were at the top of the result list (most common), 6.8% of the documents viewed were second on the list (next most common) and 73.2% of the documents viewed were found in the top 25 positions in the list.

The results discussed are particularly interesting. A majority of the queries did not lead to a document being viewed for its content. This could be the result of poor specificity from short submitted queries providing a ranked list of essentially non-relevant documents, in which case the query did not return a meaningful set. Or alternatively, it could be that the granularity of the document summary was insufficient to present information about the document that the user may have found relevant. If this is the case, then it is clear that the user is not willing to expend the time to discover the information in this way. In either case, the results suggest that users are not necessarily interested in large lists of documents returned from the search and that they essentially trust the application to rank the documents well and provide accurate summary representations even when queries are short and non-specific. It is unlikely that users expend the time to confirm the validity of this assumption explicitly but rather, this is done implicitly when their information need is not satisfied.

The implications of the last paragraph indicate that along with corollary 8, which can help with defining appropriate specificity of the query, large lists of documents returned do not necessarily add value to the search process. Also, when summary information is provided with the list of documents, this becomes the primary distinguishing criteria for relevance and not the information in the document itself. This leads to the following corollaries.

Corollary 10: Provide high level details initially, and lower levels of detail on demand.

Corollary 11: The details provided at each level should be the most accurate representation possible of the subsequent level without compromising clarity.

3.3 Summary

The purpose of this chapter was to build an analysis framework to guide the design phase of this research that follows. The chapter was presented in two distinct sections. The first section outlined the cognitive strategies important in bridging the gap between the internal and external representations of the mental models. The second section approached the problem with a different view by defining design corollaries based on the usage patterns of real people in a real environment. Both sections define the requirements framework required, which is summarized in Table 1 and Table 2 for clarity.

Cognitive Propositions	
Proposition 1	Use two-dimensional static displays to provide external memory cues while maintaining graphical constraint.
Proposition 2	Maintain intuitive relationships between the graphical elements.
Proposition 3	Use carefully selected metaphors.
Proposition 4	Provide a natural interaction model within the metaphoric environment.
Proposition 5	Support different kinds of cognitive tracing (highlights or mark-ups).
Proposition 6	Provide interactivity to test new configurations.
Proposition 7	Provide easy and intuitive creation or reproduction of external representations.
Proposition 8	Use graphic representations (symbols, icons and visual organizations) intuitive to most users.
Proposition 9	Use multiple external representations (static diagrams, text) to enhance the cognitive process.
Proposition 10	Constrain visual attention shifts.
Proposition 11	Provide a tight couple between the temporal and spatial domain to maintain consistent and progressive inferences about the representation.
Proposition 12	Direct attention to key components that are useful or essential for different stages of the retrieval process.

Table 1. Cognitive Propositions

Design Corollaries	
Corollary 1	Default setting should be set a priori and determined statistically based on user profiling.
Corollary 2	Optimize the interface around short and simple query specifications.
Corollary 3	Identify the domain specific clusters of common search terms.
Corollary 4	Allow seamless phrase construction and efficient phrase processing.
Corollary 5	Use methods to exploit the most frequently used query terms.
Corollary 6	Provide automated assistance to improve term specificity.
Corollary 7	Provide interaction models that naturally extend internal representations thereby reducing cognitive overloading.
Corollary 8	Exploit automatic and semi-automatic schema expansion and allow for exploration and substantiation of such schemas.
Corollary 9	Allow mutable, incremental and basic query refinement.
Corollary 10	Provide high level details initially, and lower levels of detail on demand.
Corollary 11	The details provided at each level should be the most accurate representation possible of the subsequent level without compromising clarity.

Table 2. Design Corollaries

IV. Design Methodology

The previous chapter explored the conceptual design propositions from a heuristic cognitive standpoint and defined some useful corollaries based on the observed interaction between users and a typical digital library. This chapter outlines the development of a novel information retrieval technique based on these design issues. The first two sections lead into a discussion on visual cues and dimensions. The third section outlines the design methodology for the visual information retrieval technique being developed. The final section discusses some additional techniques that may be added to improve interactivity with the new visual model.

4.1 Visual Cues

Currently there are two schools of thought in the information retrieval arena. The first is by far the oldest and most traditional method and the one that has enjoyed the most success. It represents the documents of a retrieval set in a single dimensional textual list, ranked and sorted by a particular retrieval model. Those that follow this discipline believe that the advancements are to be had in manipulating retrieval engines and following linguistic guidelines to extract inherent meaning from the written language. The second school of thought recognizes that one of the greatest benefits of data visualization is the human ability to rapidly interpret very large quantities of information and, with experience, draw conclusions from visual patterns. To this end many two, three, and multi dimensional displays have emerged, each with the promise of greater success than their predecessor. Those that follow this discipline do so with some caution. While visual techniques have promised a lot to the information retrieval community, they have yet to deliver a solution that is satisfactory to main-stream users. There are many reasons why this may be so, of which the learning curve required to learn the new technique, the user reluctance to do so based on a perception of limited gain and the subsequent poor results achieved in real world testing, are but a few. Whatever the reason, the benefits of visual cues for problem solving are undeniable and worth exploring.

4.2 Visual Dimensions

Three-dimensional (3D) displays are generally well suited for displaying physical data rather than abstract data. This is simply because of the smooth projections that occur naturally with physical data are not guaranteed to occur with abstract data. Human visual systems essentially draw from two-dimensional (2D) images to recreate the perspective, but this creates inevitable occlusion and the user must change the perspective to bring the occluded areas into view, at the cost of removing some information from the visible to the occluded area. Abstract data provides discontinuities that do not lend themselves to smooth 3D displays. Complex navigational requirements are also a concern with 3D displays, which tend to require either dedicated hardware (trackball, joysticks) or complex mouse motions.

Cognitive proposition 1 from Chapter 3 provided guidance on the selection of the visual dimension. The proposition identified the need to provide visual memory cues for the user while maintaining graphical constraint on the representation. Some of the problems of not constraining the representation (such as occlusion discussed above) can have a negative impact in the representation of the problem space. Therefore, a 2D static display has been used as the foundation for the visual display in this research.

4.3 Design Methodology

This section is dedicated to the step-wise development of a new visual information retrieval technique. It begins by defining the structure of a simple query and how the relationships and spatial positioning of the retrieved information is visualized against this structure. Then, query manipulation and the visualization of more complex nested queries are discussed. Finally, ontology models are introduced and integrated into the visual technique to provide automatic schema expansion.

4.3.1 The Simple Query

4.3.1.1 Query Specification

Corollaries 2, 4 and 9 provide some guidance when designing a system to specify queries. First, the specification of the query phrase should be seamlessly integrated into the visual system and provide the user with an efficient and natural input mechanism. Perhaps the most natural of inputs for users today is a text input typed from the keyboard. Therefore, the system should provide an input mechanism of this kind that seems naturally connected to the visual display. This requires a transformation between a natural language sentence and the visual structure. Second, the system should be optimized around short and simple query specifications. That is, enhanced clarity should be targeted at queries with two or three key terms, perhaps up to six, but with the ability to extend this as necessary. Third, the basic query should be easily mutable. That is, broadening or narrowing of the query should be a natural extension of the query specification process. In a text-based interface, this is achieved by either adding an additional word (or words) at the appropriate location in the phrase, (for narrowing of the query) or deleting a word (or words) in the phrase (for broadening of the query). The visual system should provide a mechanism to do the same operations with the same relative ease.

4.3.1.2 Visualizing the Simple Query

From the above paragraph, the underlying visual structure for the query must be able to display the query phrase in an intuitive fashion and be capable of mutable operations without loss of structure. Perhaps the simplest of shapes that will serve this purpose is the circle. The circle provides a simple uncluttered view that is capable of displaying nodes (or key index terms) symmetrically with optimal spacing. The nodes are simply spaced at $\frac{2\pi n_i}{N}$ where n_i indicates the position of the key index term and N is the number of index terms to be displayed. The obvious benefit of the circle here is that the structure does not change as N changes, hence it will support a mutable query phrase.

The query phrase can be specified as a typed written sentence, which is then broken into keyword nodes. These nodes can be placed symmetrically around the circle to represent the keywords of the query. This is an example of how multiple external representations (static diagrams and text) are used in conjunction to enhance the overall representation of the problem, the basis of proposition 9 from Chapter 3. An example of this is shown below in Figure 16. Using this view, the query can be broadened by removing nodes from the structure, or narrowed by adding nodes to the structure.

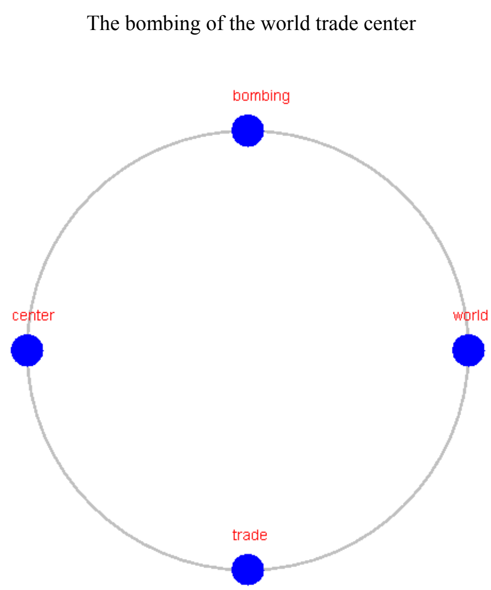


Figure 16. Structure of a simple query

One particular problem with the circle is the finite number of nodes that can be displayed by the structure. The trivial case is when $N=1$. Although the structure supports this, no discernable relationship information can be displayed (see next section). From section 3.2.1.2, the case when $N=6$ represented 98% of the queries processed. That is, only 2% of the queries exceeded this case.

The case of *N=10* is still quite discernable, as shown in Figure 17 below. Although it is possible to display queries where *N>10*, the limiting factor here becomes discerning the relationship information and not the structure itself. This will be explained further in the next section.

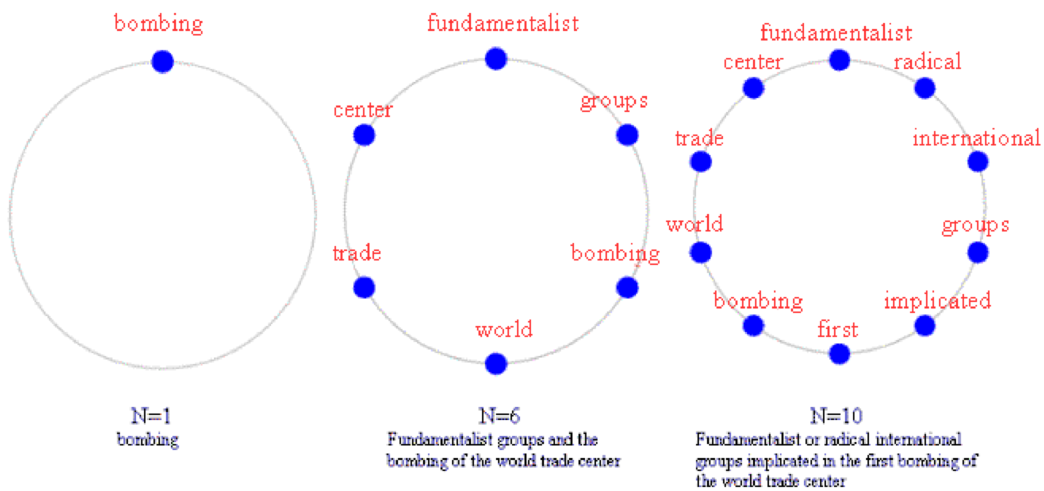


Figure 17. Examples of Simple Queries

4.3.2 Visualizing Relationships

4.3.2.1 Coding

Coding of the documents returned allows users to rely on their visual reasoning skills to quickly discern the relationships between the icons (or documents) and the keyword nodes of the structure. The key here is to identify coding principles that are understood by most users, that is, the symbols, icons and their visual organizations are intuitive and familiar. As well, the relationship between the graphical elements must also be intuitive if information is to be inferred from this relationship. This was identified in propositions 2 and 8 from Chapter 3. Spoerri's InfoCrystal [25] uses simple geometrical shapes such as rectangles, triangles and squares to represent a collection of documents with a specific set of keyword relationships and spatial positioning to indicate a particular characteristic of that relationship. The coding principles used by Spoerri fit well into this technique because not only are geometrical shapes (and the number of sides that make up that shape) an inherent part of early childhood learning and therefore familiar to all, the circle represents an element of this set which maintains some consistency in the representation. As well, the spatial relationship between the icons and the nodes, that is, the closer the icon is to the node the greater the influence that node has on the icon, is also a familiar concept since it represents a basic physical phenomenon.

The following is a list of the coding principles used in this research:

- **Shape** - Initially introduced by A. Spoerri's InfoCrystals, the shape of the icon indicates the number of keyword nodes that the contents are associated with. That is, a circle can show relationship to one node, a rectangle to two, triangle to three, square to four and so on.

- **Proximity** – The closer the icon resides to a node, the stronger is the relationship between the icon and that node.

- **Size** – The greater the size, the higher the ranking of the icon.

- **Text** – A number resides next to each icon to indicate the number of documents associated with the icon. The node can also display a number to indicate how many documents are associated with the node.

- **Color** – Color is used to show information on demand such as which icons are associated to a particular node or group of nodes, which icons have previously been investigated, and so on.

- **Shading** – Regions are shaded to show which combinations of nodes are related to the icon.

An example of some of these coding techniques in use is shown in Figure 18. The first row shows the icon relationship to the nodes through shape, proximity and shading. Both Figure 18(a) and 18(b) show approximately equal weighting of keyword terms, whereas Figure 18(c) shows a heavier weighting on the node 'trade' with equal weighting of the terms 'world' and 'center'. Spatial positioning is investigated in more detail in the following section. The second row shows an example of coding by color. Figure 18(d) shows two nodes, 'bombing' and 'world' selected. The blue icons indicate that they are completely matched by the two selected nodes, the green icons show icons that are only partially matched by the selected nodes and the black icons show no matching to the selected nodes. Figure 18(e) uses color in a different way, this time to show state history of the icon. The icon color changes to indicate which have been previously investigated, and which have not.

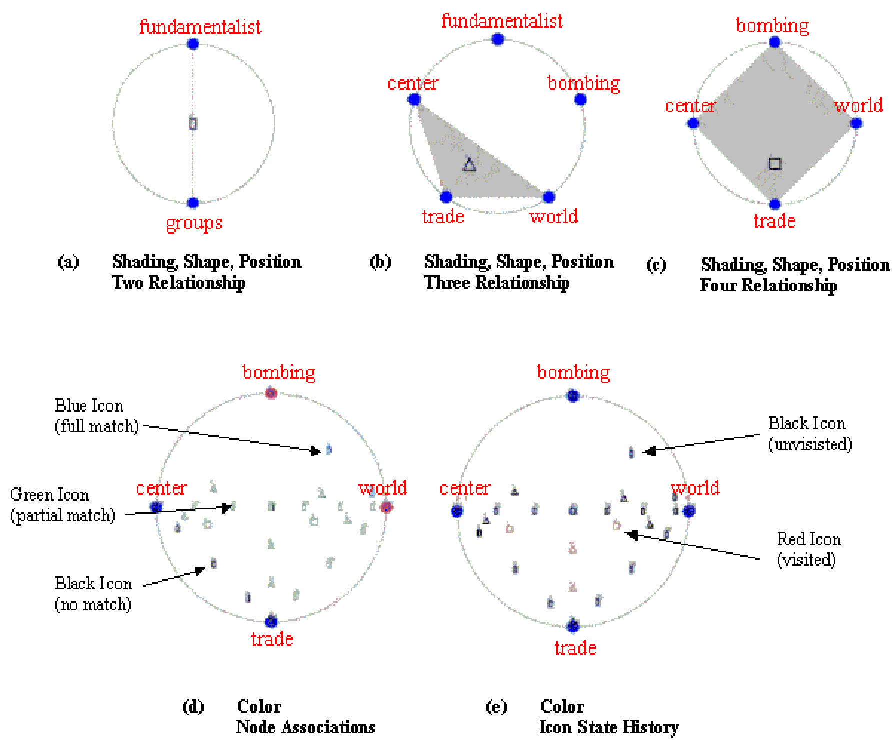


Figure 18. Demonstration of some coding techniques used

4.3.2.2 Spatial Positioning

The vector Model was used as the information retrieval search engine primarily because of its simplicity in design and high recall ratios achieved. That aside, it provides a convenient tool to apply two-dimensional spatial positioning on individual documents because essentially, they are vectored into position. The equation that achieves this was described in Chapter 2 and is shown below for clarity:

$$sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} * w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} * \sqrt{\sum_{j=1}^{t} w_{i,q}^2}} \qquad (16)$$

In this case, the sim(d_j,q) represents the similarity value for document d_j when applied to query q. The result is a single value that can be used to rank documents in a descending relevance list. Of more importance here, is how each keyword index weight can be broken down into normalized vector components as shown below:

$$\alpha_{i,j} = \frac{w_{i,j} \times w_{i,q}}{\max_l w_l} \qquad (17)$$

where $\alpha_{i,j}$ *represents the normalized scalar vector component* i *of document* d_j *and*

$\max_l w_l$ is the maximum value over all components.

The question now is how to apply the scalar components to a vectored position. A simple but effective technique is to consider each node, n_i, to have an attractive force equal to its normalized scalar component. The icon then settles at the position of equilibrium between these nodes. This can be visualized simply by considering the icon being connected to each node via invisible springs with a tension equal to the attractive force. The icon will rest at the point were the resultant force is zero.

There are two points worth mentioning with this spatial positioning technique. First, only relative strength relationships are visible. Consider figure 18(a) above. The position of the icon indicates an equal attractive force between 'fundamentalist' and 'groups', but there is no indication of relative strength of this force. Therefore, this technique must be supplemented with additional coding to provide a visual cue on absolute strength. This is achieved by incorporating size coding to represent absolute strength of the attractive force. This is easily implemented by relating size to $\sum a_{i,j}$, but note the complication. Now documents with exact term relationships and spatial positioning, but different size must now be either differentiated or ignored. That is, either multiple icons of different sizes will try and occupy the same space, or only one icon will exist but the ranking information of the document is lost. Since trying to map many icons to the same space cannot work, for obvious visual clarity issues, the documents must be grouped into a single icon where the size of the icon represents the highest ranked document within the icon collection. This solution works for the following reasons. First, icons can still be compared against each other to determine the rank order, the only occlusion occurring is the exact number of documents that are ranked higher in one icon compared to that of another. Second, documents are still in rank order within the icon so document level ranking is not lost just relative ranking across the whole collection. Third, the probability of this occurring reduces significantly as the order of the shape increases and as the order of the shape increases so does the probability that the ranking will be higher. In other words, higher-ranking documents tend to be located in icons with diminishing collection sizes.

The second point worth mentioning is that there is no immediate cue to which nodes provide an attractive force to the spatial positioning of the icon, and which do not. Again, an additional coding technique has to be included here. The technique used by A. Spoerri was to align each icon such that the faces pointed to the nodes of influence. This would have only limited success here, since the shape of the icon would have to be determined not only by the number of nodes causing influence on the icon, but also the spatial positioning of that icon. The more complex the relationship, the harder it is to discriminate the relationships in play. For this reason, this technique was not used. Instead, an object-in-focus technique is used. The icon in focus develops a shaded polygon with each vertex represented by a node having a positive influence on the position of the icon. In this way, the user can not only discriminate which nodes have influence on the icon, but also gains clarity about the relative positioning of the icon to each of these nodes. Figure 18(a) thru 18(c) shows this technique.

4.3.2.3 C-Factor

As the number of documents being mapped to the two-dimensional structure increases, two effects become more prominent. First, and perhaps most obvious is that the clutter of icons rapidly increases to the point that individual icons can no longer be distinguished from one another. The second point is that icons, regardless of the nodes influencing their position, can be mapped to the same spatial point. In this case, without a third dimension, the icons would perfectly overlap each other causing occlusion.

A natural progression to this problem is to represent groups of documents (repositories) within single icons instead of an individual document representation. However, care must be taken to maintain the integrity of the icon representation. To do this, the merging of a document into an icon is forced to obey the following simple rule:

Document d_j can merge into Icon I_m containing document d_k iff

$$\sum_{i=1}^{t} \alpha_{i,j}\phi_{i,m} = \sum_{i=1}^{t} \alpha_{i,k}\phi_{i,m} \; where \, \alpha_{i,j}, \alpha_{i,k} > 0 \qquad (18)$$

where $\phi_{i,m}$ *represents the angle from the position of Icon* I_m *to node* n_i

This merging rule groups documents into icons that have identical term relationships and spatial positioning and excludes those that don't. Icon occlusion is not completely prevented using this method, however, the probability of two documents with differing relationship criterions being spatially mapped to the same exact point is significantly less than that of two documents with the same relationship criterions.

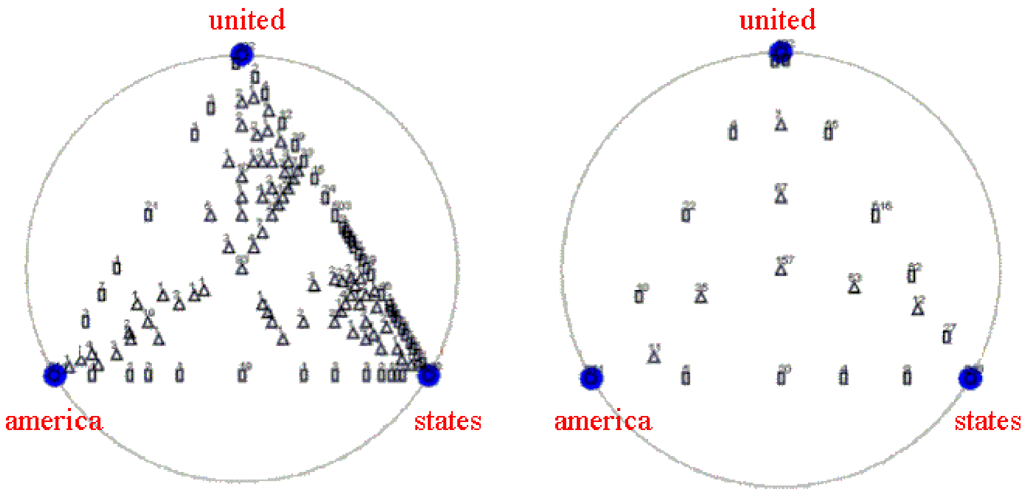
The problem of icon clutter remains even with the grouping scheme discussed above. This is simply because only perfect matches in relative relationships can be grouped. To alleviate this problem, a neighborhood zone around each existing icon is defined. If a document containing the same nodal relationships as an existing icon is placed within this icon's neighborhood zone, then the document can be merged as before. This essentially reduces the restrictions of the document merging rule above to incorporate near hits. The merging rule therefore becomes:

$$\left|\sum_{i=1}^{t} \alpha_{i,j}\phi_{i,m} - \sum_{i=1}^{t} \alpha_{i,k}\phi_{i,m}\right| \leq r \qquad (19)$$

$$where\, \alpha_{i,j}, \alpha_{i,k} > 0$$

and r is the radius of the neighborhood zone.

The size of the neighborhood zone is known here as the C-Factor, for the consolidation of icons. The diagram in Figure 19 shows an example with and without the use of neighborhood zones. The use of neighborhood zones does distort the spatial positioning of the documents but individual fidelity is traded for clarity. It is not expected that this loss in fidelity will significantly alter the users ability to reason about the relationships of the documents to the nodes. It will, however, increase the repository size of the icon (number of documents merged into a single icon) as less and less detail is displayed. This is congruent to corollary 10 and 11 of the previous chapter, which identified that higher-level details should initially be displayed in the most accurate representation possible and that lower levels of detail should be displayed on demand without compromising clarity. Details on demand will be discussed further in a subsequent section of this chapter.



(a) Without Neighborhood Zones

(b) With Neighborhood Zones (high C-Factor)

Figure 19. The Effect of Neighborhood Zoning of Icons

4.3.3 Physicalization

Before the mechanics of query manipulation are examined, it is worth digressing for a short time to discuss propositions 3 and 4 from Chapter 3. These propositions suggest that the environment in which the user is immersed in should be crafted around carefully selected metaphors, and the interaction within this environment should be a natural extension of the metaphors chosen. By doing this, the cognitive load of translating between the internal mental model and the external representation is reduced, and the interaction becomes a natural extension of the problem.

The circular structure can be viewed as an information bubble, that represents the query construct. The nodes, representing the keywords of the query, can be viewed as the orbs suspended tangentially at the surface of the bubble providing the attractive forces that are applied to the internal contents. The contents (termed icons) are suspended in equilibrium within the bubble and do so according to the attraction of the orbs. Submitting a query creates an environment at equilibrium. Changing the environment, such as bursting or creating new orbs or bubbles causes the equilibrium to change to reflect the new state.

4.3.4 Manipulating the query

The concept of corollary 9 is to allow queries to be mutable in an incremental manner. This reflects the nature of the search pattern of users when submitting queries to an information retrieval system. That is, users begin by specifying a query (query creation), and then adapt the query in an incremental narrowing and broadening fashion (query refinement) until the resultant document set matches that of the users need. This technique is typically used in an exploratory fashion since the user is generally disassociated with the contents of the collection and must therefore 'learn' how to formulate the question to retrieve the appropriate results.

The query is manipulated interactively within the environment. To construct a query, a bubble is created and orbs are placed around the bubble at an equidistant position maintaining equilibrium. Icons are created as the query is submitted and drift to their equilibrium positions within the bubble. To narrow the query (increase term specificity), users add additional keywords. Adding additional orbs to the bubble, which causes new icon sets to be created and new equilibrium points to be found, does this. To broaden the query (decrease term specificity) users remove keywords. Removing orbs from the bubble, which again creates a new icon set and new equilibrium points to be found, does this. The same can be said for the bubbles themselves. To remove whole queries, the bubble can be removed by 'bursting' the structure, and to add whole queries, new bubbles can be created.

In the next section, the creation of complex nested queries is described in some detail. For now, this concept will be described metaphorically as additional bubbles attaching themselves to existing bubbles at the point of an orb. At this finite point of contact between two bubbles, the contents of the icons that have some attraction to the orb are free to pass through the node into the next bubble. In reality, the contents (documents) do not migrate but rather replicate and then undergo a transformation into new icons. The point of this may be subtle but is significant. This provides a filtering mechanism across the nesting of the queries. The point of nesting defines the filtering characteristic. Again, this is defined in more detail in the next section. For now, it is enough to say that the new nested bubbles have the same interactive characteristics as previously described, however, they exhibit a hierarchical influence on the lower level nesting.

4.3.5 Visualizing complex nested queries

The knowledge crystallization process introduced in Chapter 2 essentially defines the process in which one person gathers information for some purpose, makes sense of it by constructing a representational framework or schema, and then packages it into some form of communication or action. If the data does not fit the current schema, then the search continues for a better schema, all-the-while removing non-relevant data through the abstraction process. The problem is how one visualizes the schema relevant to the search at hand. For a single dimensional textual interface, the schemas are stored in the users cognitive thoughts, which for all but the simplest of problems, soon becomes overwhelming due to cognitive overloading. Perhaps the most fundamental argument for a visual system is the ability to visualize these schemas and provide this much needed cognitive off-loading.

Schemas are a partial representation of structure nested on a particular object. For instance, consider the boolean query below:

Query: Fruit OR A OR B
Where A => Vegetable (Carrot OR Potato)
And B => Cereal(Wheat)

The nested query would be shown as:

Query: Fruit OR Vegetable (Carrot OR Potato) OR Cereal (Wheat)

Where Vegetable and Cereal pertain to a particular schema.

In this particular case, the nesting points are ‘Vegetable’ and ‘Cereal’, that is, a schema defines a structure below each of these objects. Figure 20 shows how these schemas are incorporated into the bubble structure previously described.

“Fruit OR Vegetable (Carrot OR Potato) OR Cereal (Wheat)”

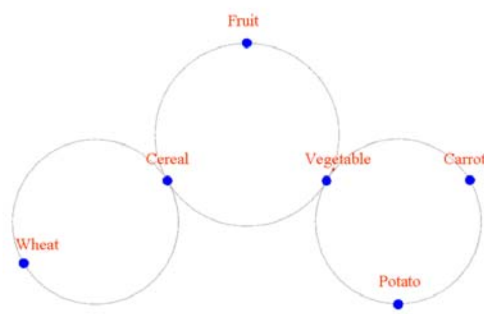


Figure 20. Representation of a simple schema

The diagram above clearly defines a root query (Fruit OR Vegetable OR Cereal) and two schemas (Vegetable – Carrot OR Potato: Cereal – Wheat). Returning to the knowledge crystallization process described earlier, a person could now investigate each schema independently or as a whole, manipulate them through incremental query changes (adding or removing orbs) or create new schemas (adding or removing bubbles) depending on where the exploratory path takes them.

4.3.5.1 The Boolean Filter

In the above paragraphs it was shown how schemas could be developed by combining bubbles at the point an orb is located. This concept was shown using a simple boolean example. However, icons are located within the bubbles using a vector model and not through boolean operators. In this section, this concept is explored further to explain how the two models can be fused together to provide an effective hybrid model.

One of the biggest criticisms of the vector model is that words are treated as pair-wise independent, which in many cases is not realistic. Consider the terms 'White House'. If the terms are considered independent, then the returned document set is not likely to be of much value to the user. Now consider terms physically separated but still not independent. An example of this might be 'President Reagan Wife'. It is likely that a document having some information about President Reagan's wife would contain all three words, but proximity matching may not help here. The boolean model would help in this situation, however, this model is too restrictive and there is no provision for ranking. Mixing the boolean and vector method provides a viable solution.

Consider the following example. A person wishes to discover what information the collection has on the presidents of the United States, and if possible, something about their wife. At the beginning of the search, the names of the presidents of interest are unknown, so the query shown in Figure 21 is issued:

"President of the United States"

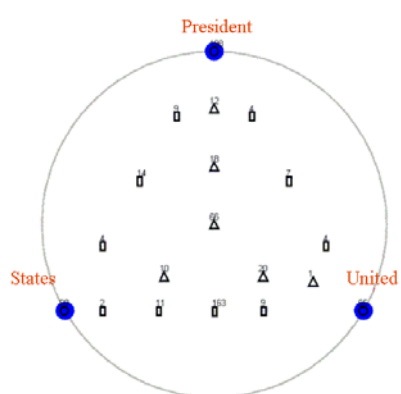


Figure 21.Boolean Filter Example – Step 1

As the browsing and searching continues, it turns out that Presidents Reagan, Truman and Kennedy are of particular interest and the schema of Figure 22 is created.

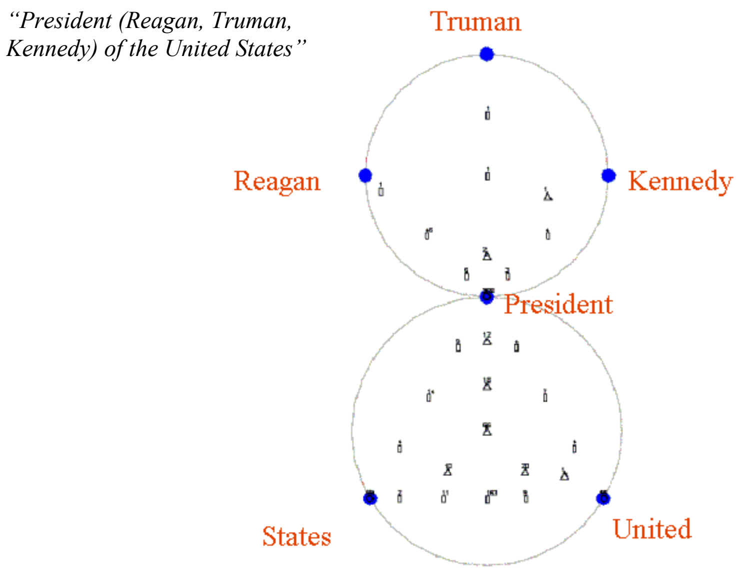


Figure 22. Boolean Filter Example – Step 2

In this case, only the documents matching (UNITED OR STATES OR (UNITED AND STATES)) NOT PRESIDENT are excluded from replicating to the above bubble. Visually, this is seen as the documents lying on the line connecting the orbs 'United' and 'States'. All other documents are replicated and make up the document set for the second bubble. The vector model is again executed across this document set and the icon collection is created and placed accordingly. Now documents with PRESIDENT AND NOT (REAGAN OR TRUMAN OR KENNEDY) are trapped at the connecting orb 'President'. All other documents are placed into icons that show some relationship to the schema of interest. Information about these presidents can now be investigated in this second bubble with greater confidence of relevance than a search on REAGAN, TRUMAN or KENNEDY alone. To continue this example, information about the respective wives is now sought, so the schema develops as shown in Figure 23.

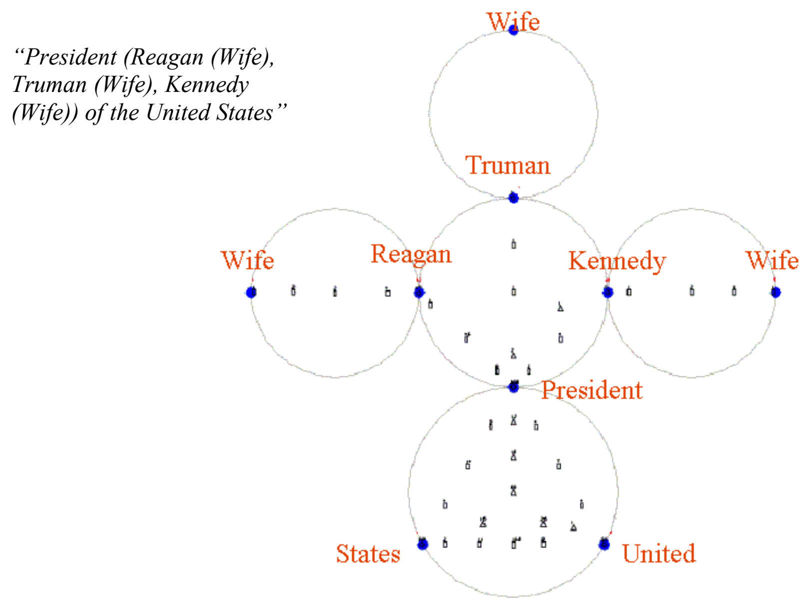


Figure 23. Boolean Filter Example – Step 3

Again, the documents found in the left most bubble now contain the words 'President' and 'Reagan', and only those that have the word 'Wife' will be placed away from the connecting orb. What is immediately obvious is that the number of documents reduces quickly as the document sets filter through each node. This potentially reduces the number of documents that need to be searched in further detail. Investigating the new schema bubbles reveals that the name of President Reagan's wife is Nancy, there are no references to President Truman's wife and President Kennedy's wife is Jackie. This information is now included into the schema as shown in Figure 24.

Now, the documents found in the left most bubble have the words 'President', 'Reagan' and 'Wife' and these documents are ranked according to the vector model on the query 'Wife Nancy'. In this example, the three documents found in this bubble all refer to Nancy Reagan, the wife of President Reagan.

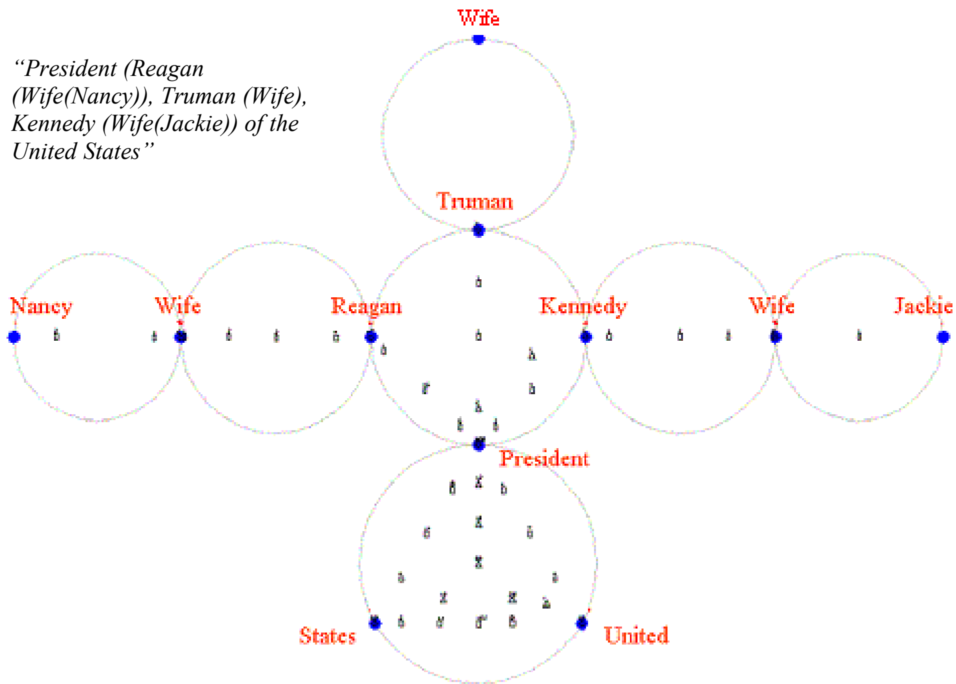


Figure 24. Boolean Filter Example – Step 4

This technique looks promising, but a couple of problems need to be addressed. First, relevant documents may prematurely be excluded through the boolean filtering stages. In the example above, documents that reference Nancy Reagan may never make the left most circles if the terms 'President and Wife' do not appear. This is indicative of the restrictions that accompany the boolean model, however, this can be overcome by adapting the bubbles appearing further up in the schema hierarchy. For example, the terms 'Nancy Reagan' can be added to the root query, investigated, and then removed accordingly. This essentially switches between the boolean model and the vector search models as needed. Second, it may not be clear which term to filter on. For instance, say the query was "President in the White House". A bubble 'House' could be added to 'White' or 'White' to 'House' however, this is somewhat non-intuitive. Perhaps a better technique is to allow filtering on proximity terms or phrase matching. This is investigated further in the next section.

4.3.5.2 The Proximity Filter

The proximity filter is included here as an extension to the boolean filter, where instead of a single term to filter on, a phrase or collection of terms may be used. In the previous paragraph, the query: "President in the White House" was given as an example. In this case, the terms 'White' and 'House' are explicitly linked and should not be considered independent terms. It would make no sense to filter on the term 'White' or 'House' alone, so the query is translated to a single bubble with two orbs, 'President' and 'White House'. Documents replicated through the latter orb must contain the phrase 'White House' and not just the terms 'White' and 'House'. An example of this is shown in Figure 25.

This concept can be extended again to include terms within a specified proximity of each other. For instance, 'Computer Network' with a specified proximity distance of one, would allow documents containing the phrase 'Computer Support Networks' to be replicated through the orb. To visualize this, a period between the terms would specify the proximity distance, for instance 'Computer .. Network' would specify the proximity distance as two, 'Computer … Network' as three and so on.

"President 'White House'
(Security Protocol)"

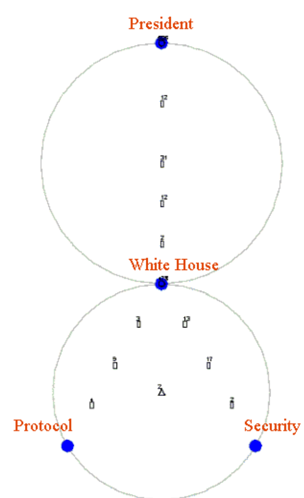


Figure 25.Proximity Filter Example

4.3.6 Interaction

With any medium to large scale collection size, one of the most important considerations is how best to hide details and provide high level abstract views of the data initially, yet provide an intuitive interaction model that allows the user to 'drill down' into the data until the primitive view of the data is reached (the document itself). Previous sections described how grouping could be achieved using neighborhood zoning and in doing so the initial exploratory view was kept uncluttered. In the next section, this exploration is broken down into three distinct environments: (1) the exploratory view, the overall view of the collection based on the created schemas, (2) the icon view, the environment to explore the repository collection of an icon, and (3) the document view. The next section is dedicated to explaining the overall interaction model, and the individual environments.

4.3.6.1 Overview

It is clear that without icon grouping the abstract view of the document collection would become overwhelming and the selection of an individual document from this view would be difficult. For this reason icon grouping was introduced to allow clusters of documents with the same relational positioning to be grouped together. This was extended further to include neighborhood zoning in which documents with 'nearly the same' relational positioning could also be drawn into a nearby grouping. Now the initial exploratory view has been de-cluttered but at the expense of larger grouping sizes. This creates a need to be able to interact with the icon repository collection to eventually reach a document to view.

Before proceeding with the design methodology of the interaction model, it is worth considering propositions 10, 11 and 12 from Chapter 3.

Proposition 10: Constrain visual attention shifts.

Proposition 11: Provide a tight couple between the temporal and spatial domain to maintain consistent and progressive inferences about the representation.

Proposition 12: Direct attention to key components that are useful or essential for different stages of the retrieval process.

These propositions provide guidance in how the interaction model is to be crafted. In summary, to maintain a consistent cognitive flow the exploratory view should change: (1) only if it provides a meaningful extension to the browsing process, (2) only at the right time and place, and (3) should maintain the attention of where the view came from.

To achieve this goal, the interaction model is limited to three distinct environments: the exploratory view, the icon view and the document view. The exploratory view provides the overview of the document collection depicted by the schemas that are defined. That is, not all documents are seen at all times, but only those that fit into the current need of the user. The icon view provides a means to further explore the repository collection of an icon. The obvious interaction link between the exploratory view and the icon view is the icon itself, so by selecting the icon the user moves from the exploratory view to the icon view. Note that this is consistent with the guidance from the propositions above since, by selecting the icon, the user has depicted a particular area of interest and now full attention should be directed at this point. This can be achieved with a complete attention shift provided some means of relating back to the previous view is maintained. Once in the icon view, the user will ultimately wish to progress to see an actual document. Again, this is a convenient point to provide an attention shift since, for a moment in the browsing phase, full attention to a particular document is the only concern. Therefore, selecting a document icon in the icon view will cause the user to move from the icon view to the document view. This interaction is depicted in Figure 26.

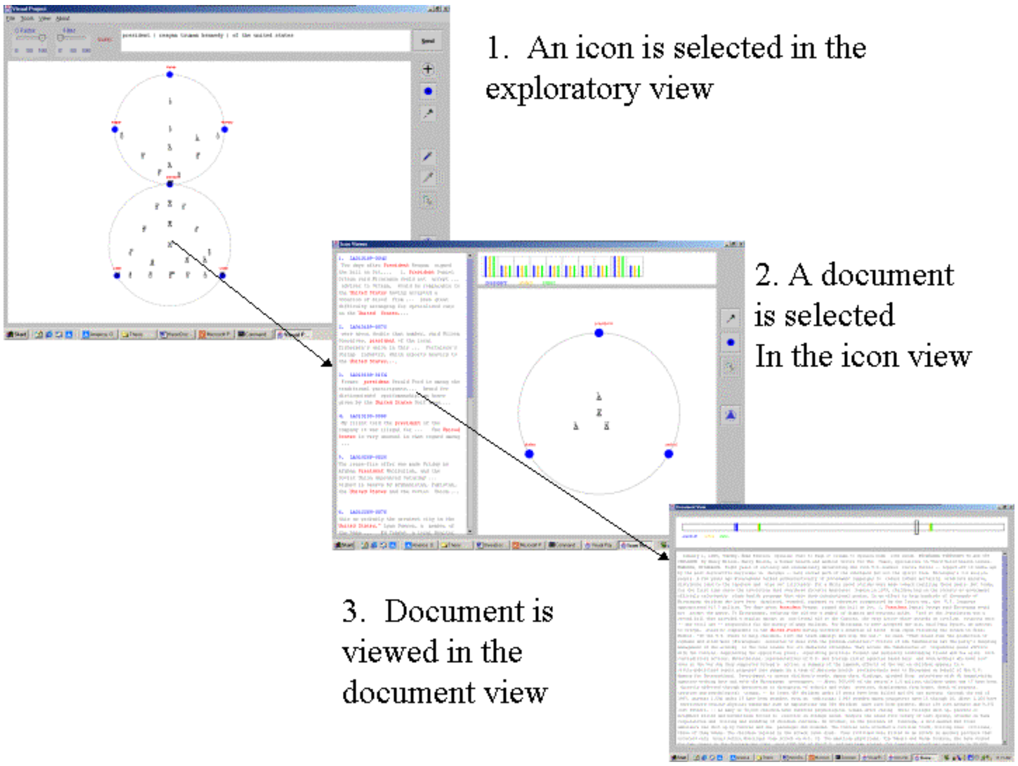


Figure 26. The three interaction views

Interaction from one view to the next is bi-directional. That is, once in the icon view the user could chose to return to the exploratory view or progress to the document view.

4.3.6.1.1 Exploratory View

Propositions 6 and 7 from Chapter 3 stated that the interaction environment should provide a means to easily and intuitively test and create new representations of the view. This is the purpose of the exploratory view. In this environment, schemas can be built, modified and destroyed quickly to gain an understanding of the underlying collection set without requiring detailed knowledge of the domain beforehand.

The exploratory view can be considered as a window to the collection. The window can be moved so that every document in the collection will, at some time, be visible but not all documents will be visible at any one time. Moving the window around is achieved by the act of creating (and destroying) the root query. Building schemas from this root query is likened to closing the curtains so that only a small subset of the collection seen through the window is now visible at the ends of the schema. Of course, for each curtain that is closed, there exists another curtain that can be closed further until eventually the window is totally covered and no documents can be seen at all at the ends of the schema. Note that closing the curtains does not have the same effect as moving the window. The curtain can always be re-opened to reveal what was previously hidden (destroying part of the schema) but moving the window (redefining the root query) will change the document view, not just hide it.

Consider the example shown in Figure 27. In this case, the root query produces a set $S_1 \subset U$ where U represents the full document collection. Now, as the schema is developed, documents from S_1 can replicate through the node 'President' to form set $S_2 \subset S_1$. Again, the schema continues so that $S_3 \subset S_2$, $S_4 \subset S_2$ and $S_5 \subset S_4$. It naturally follows that $S_5 \subset S_1 \subset U$.

4.3.6.1.2 Icon View

At this stage the collection view is limited to the repository of documents within the icon. The size of this subset collection however, may still be quite significant especially if the initial collection size is large. Therefore, just as it was not appropriate to view individual documents in the exploratory view, it may not be appropriate to view all the documents individually in the icon view. This comes back to the scalability of the problem. To overcome this, the icon view is designed as an abstract manipulation layer that sits between the view of the collection and the view of the document.

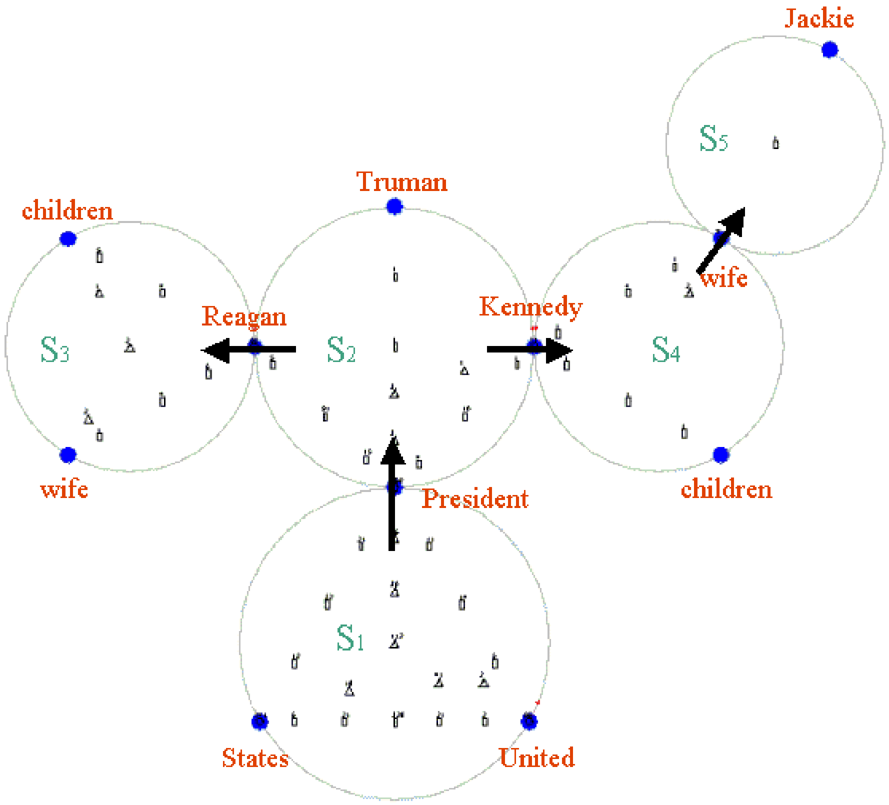


Figure 27. The formation of collection sets

This layer works as follows. The icon view layer is initially created around the icon selected in the exploratory view where only the orbs that had influence on the icon are reconstructed. An example of this is shown in Figure 28(b). In this view, additional orbs can be added (or removed) to break open the icon(s) and redistribute them, shown in Figure 28(c). When a particular icon of interest is identified (by relative positioning as before) then it too can be selected causing an additional layer of abstraction to be created around that icon, shown in Figure 28(d). The document collection contained within this newly selected icon is passed through the layer, and the view is reconstructed around this new icon. This manipulation phase can go on for an indefinite number of abstract layers, and the layers themselves can be transitioned up or down as they are created. This creates a browse and filtering environment, which can help locate a particular document for viewing.

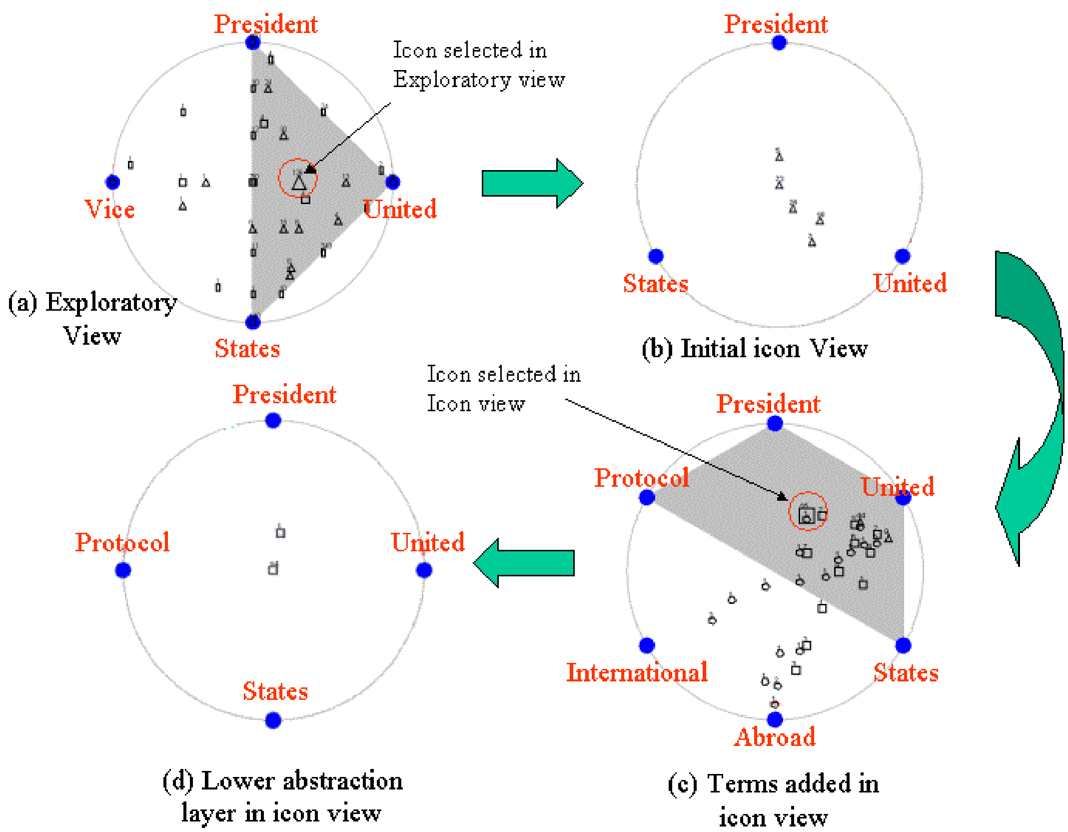


Figure 28. Transitions through the Icon View abstraction layers

4.3.6.1.2.1 Additional Browsing Tools

Transitioning through the abstraction layers of the icon view was described in the previous section. At each layer, however, a decision must be made by the user to either continue proceeding further down into the next abstraction layer, move back up to the previous layer or discontinue that search altogether. To assist in making the decision, some additional tools need to be incorporated into the icon view environment.

There are two tools that can be useful in quickly learning something about the documents of a collection, the first is a brief summary about the documents of the collection (based around the key words of interest) and the second is the frequency of the keywords within the document (i.e. high frequency of particular words may be more important than moderate frequencies of all key words). Therefore, these tools are added to the icon view along with the ability to transition through the abstraction layers as previously described. Figure 29 shows the full icon environment. Note that trying to display this information for all documents represented at each layer is again impractical for the same reasons that required the concept of the abstraction layers to be introduced. Therefore, since ranking information on the documents is available at each layer, the summary descriptions and frequency graphs are displayed only for a small subset of the top ranked documents. The rest are ignored at this layer, and require further transition down the abstraction layers before they become visible in this way.

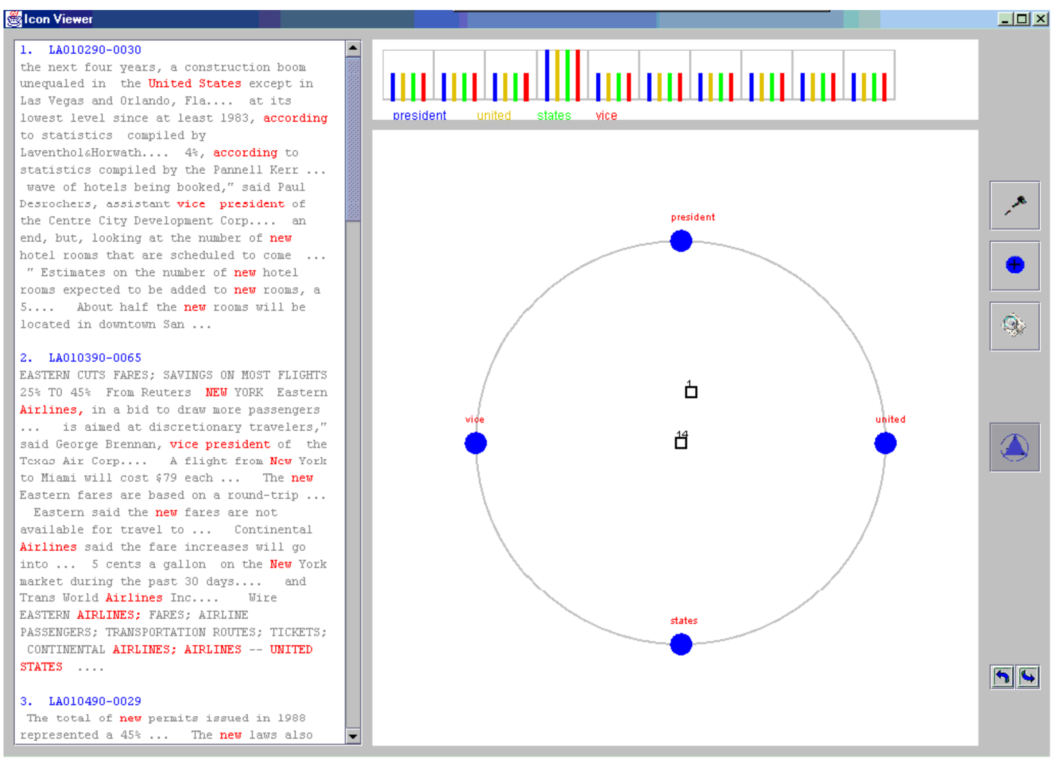
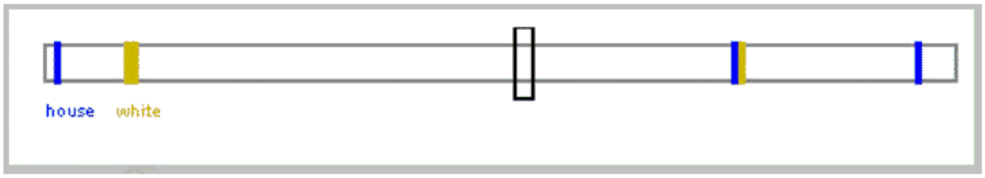


Figure 29. Icon View Environment

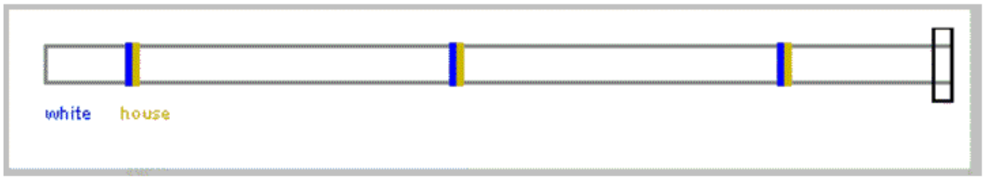
4.3.6.1.3 Document View

The document view is reached when transgressing an arbitrary number of abstraction layers leads to a document of interest. At this stage, and because of the interactive filtering that has occurred, it is likely that the document being viewed is of some relevance, and therefore some interest, to the user. Unfortunately, like any information retrieval system, there are no guarantees that the document is indeed relevant since the complexity of the lexical grammar of our language allows for redundancy of meaning using otherwise similar groups of words. Therefore, the document viewer represents the end of only one search path and perhaps the beginning of another. The search/browsing continues by transgressing from this point up the icon view abstraction layers until other icons of interest are located.

The role of the document view is therefore two-fold. First, the document view must present the actual text of the document. Second, the document view must present an environment that can quickly alert the user of the potential relevance of the document. If the document turns out not to be relevant, or at least only partially relevant, then the user must be able to see this quickly without having to read the full document text. To achieve this, a "query term hits within document content" approach is used [1:289]. Put simply, an overview of the location of keywords within the document is displayed so that some proximity information between the words can help identify the context of the document. For instance, consider the example in Figure 30 below in which information on the "White House" located in Washington DC is sought. In Figure 30(a) it is clear that there is no semantic correlation between the words 'white' and 'house' since a yellow and blue bar are not located together (in that order). In this case the user could infer that the document is not relevant to their search and move on. In Figure 30(b) another document clearly shows a blue and yellow bar located together (in that order) several times throughout the document. There is a significant chance that this document has a high semantic correlation between the two words. In this case the user has some confidence that the document may be relevant and would continue by reading the sentences around the keyword pairs to gain context information.



(a) Example of no semantic correlation



(b) Example of semantic correlation

Figure 30. Query term hits within document content

4.3.7 Ontology Models

Exploratory browsing is a fundamental technique used in retrieving textual information from large information sources. The process of browsing is complex however, and it is not always intuitively obvious to the user on how to proceed at any given point. For instance, if the user is not an expert in the domain they are searching, they may have some difficulty in expressing their need in the language provided to them by the system they use. Even if they could express this need, the user may wish to organize related information into common schemas, but exactly how these schemas should be specified may not be known.

The importance of ontology to specify conceptual knowledge and its relationships has been identified as an important tool to be used in knowledge based systems and information retrieval systems [56,57]. Ontology provides semantic structure to an otherwise unstructured information environment. Of importance for this research is how this semantic structure can be integrated into the need specification process.

An active area of research has been to develop ontology based knowledge models that simulate users' exploratory behavior by matching their need with the necessary information required to carry out the task at hand. For instance, if the user is exploring information about a 'bus terminal' then related input information on bus schedules, routes and ticket prices could be used to support the user's need specification. The next section investigates how this information can be integrated into the information schemas described in section 4.3.5 of this thesis.

4.3.7.1 Integrating Ontology Models into Schemas

Consider another example, this time a hierarchical diagram of an aircraft structure. A common method of displaying this type of information might be as a connected tree diagram shown in Figure 31. In this diagram, each sub-category is represented as a child to the parent category with the relationships specified by lines and arrowheads.

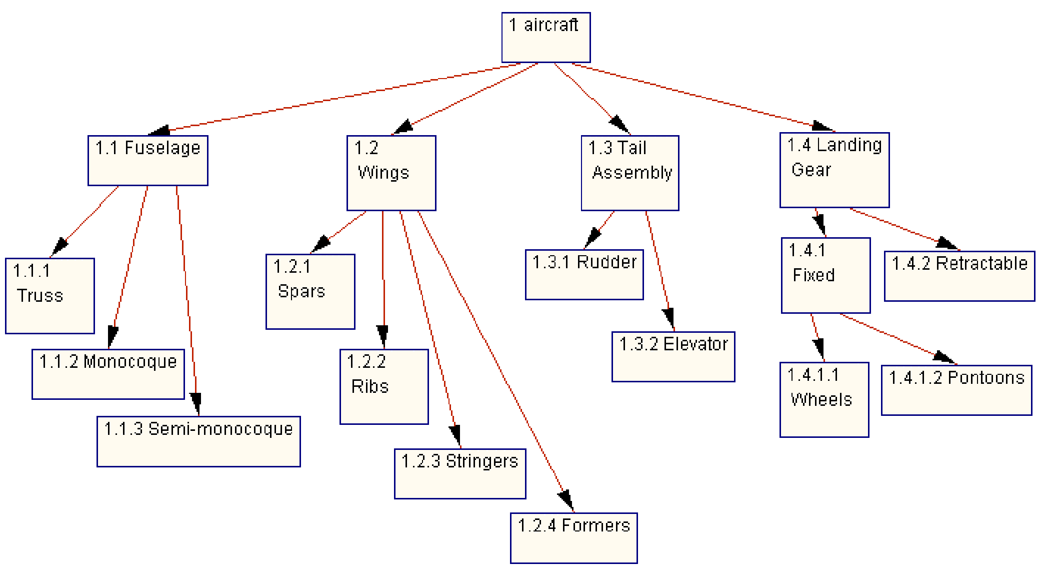


Figure 31. Example of an Ontology Model of an Aircraft

A distinctive property of such a diagram is the inherent structure it represents. For instance, the diagram shows that aircraft has a fuselage, wings, tail assembly and landing gear. The fuselage has types truss, monocoque, and semi-monocoque and the wings are made up of spars, ribs, stringers, and formers. In each case, a category can be identified with a number of subordinate categories. This diagram is shown again in Figure 32 with the categories identified.

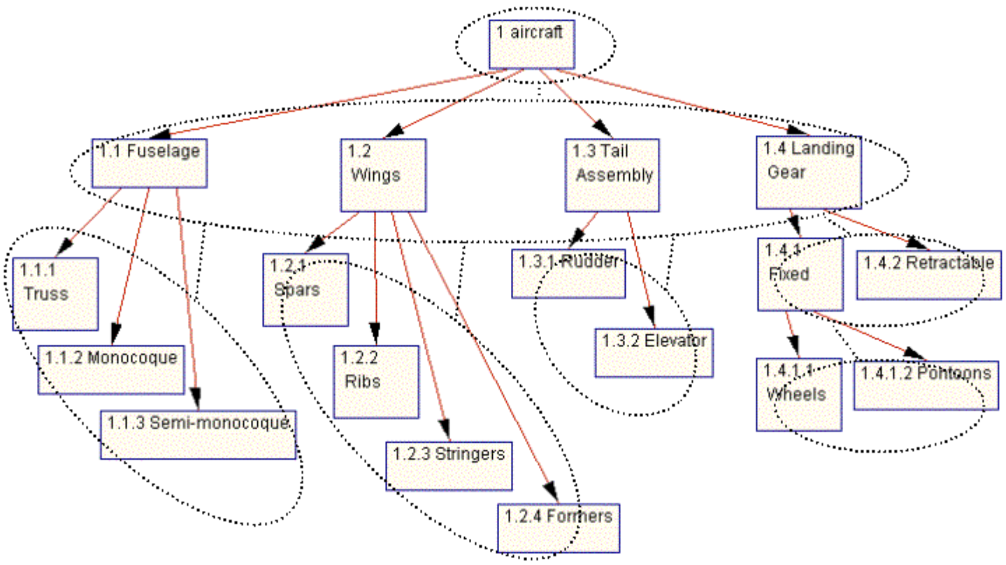


Figure 32. Aircraft Ontology with Identified Categories

To integrate this diagram into the schema diagrams shown previously in this chapter (bubbles and orbs), it is a simple matter of identifying the ovals as bubbles, and the connection between the ovals as the orbs. Figure 33 shows the completed transformation of the aircraft ontology into this representation.

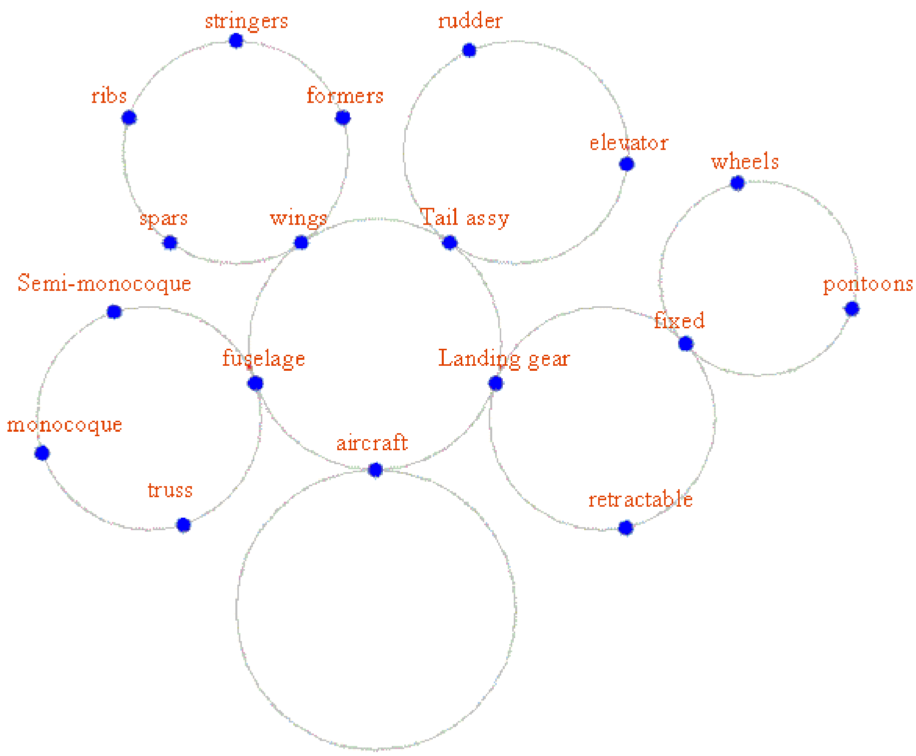


Figure 33. Example Transformation of the Ontology Model of an Aircraft

Since ontology models can be easily converted into the form of Figure 33, it follows that automatically expanding a user's schema is also possible, which is the essence of Corollaries 5, 6 and 8 from the previous chapter. If ontology models can be pre-defined based on expert knowledge of the domain, then automatic schema expansion can provide a means to guide the user into areas of the collection that may otherwise have been hidden from the search.

Consider the example of exploring bus terminals as discussed above. In this case, the user may have any number of reasons and needs to explore bus terminals. For the purpose of this example, a domain expert has determined that most users investigating bus terminals are interested in bus schedules, bus routes, and ticket prices. Furthermore, those looking for ticket prices are usually interested in cheap or discounted fares. Without automatic schema expansion and some knowledge of the domain, schemas would be created and removed in somewhat of an ad-hoc manner until the exploration revealed some useful results, or the user grew tired of looking. With schema expansion, the user is guided into browsing particular areas of the collection that may be of more relevance. Consider the same user browsing for bus routes in the area they live, north of the city. The schema in this example was not detailed enough to provide this information, however, the user is now guided to extend the schema off 'bus routes' and continue extending the schema until the browsing process has been completed. Figure 34 shows this example where the highlighted bubble indicates the evolution in the schema. The important point to gain from this example is that ontology models presented in this framework need not be completely specified to function adequately. Instead, if the models can provide high level guidance into particular areas of the collection, then individual user's can be guided into those areas and then refine and modify them according to their individual need.

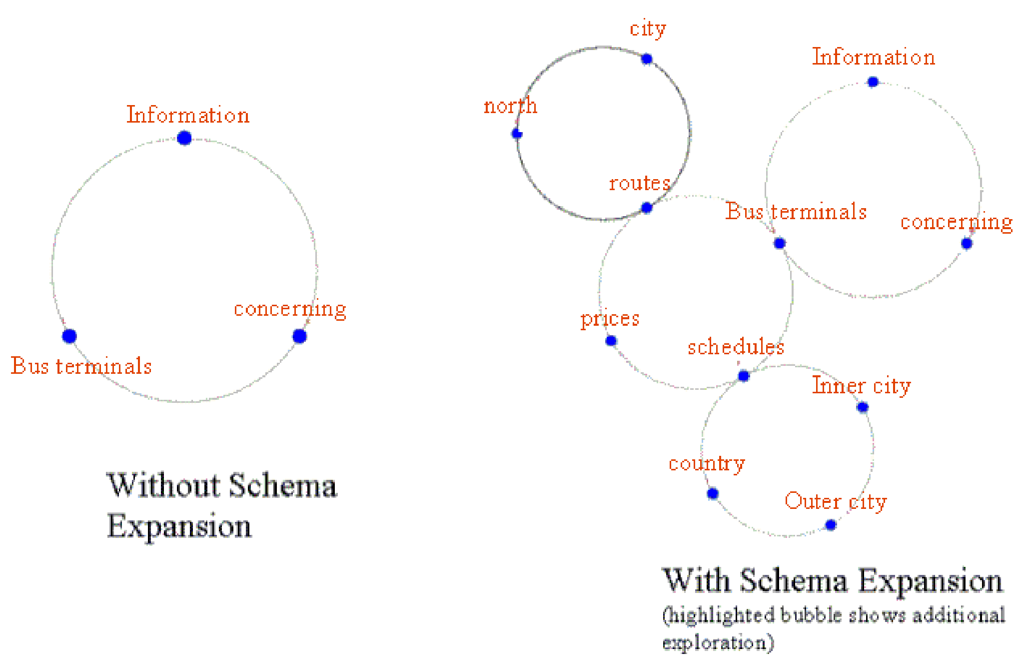


Figure 34. Example of Automatic Schema Expansion

4.3.7.2 Thesaurus

One of the difficulties with specifying queries is the consistency of the terminology. Consider the previous example, which specified the hierarchical structure of an aircraft. Although this may be totally appropriate for some documentation, others may refer to the 'tail assembly' as the 'empennage', or yet others may refer to the 'elevator' as the 'horizontal stabilizer'. Previous research has shown that the probability of matching between the terms in the domain and the vocabulary used by the user is less than 20% [58].

One of the many benefits of ontology models is that they are specified a priori, based on expert domain knowledge of the information collection. This means that not only can structural information about a particular model be specified, but the opportunity to specify semantically equivalent terminology is also possible. Adding a thesaurus capability to the model is essentially adding the OR portion to the boolean model described earlier in section 4.3.5.1 of this thesis. Consider the aircraft ontology example again. If 'empennage' is added to the thesaurus collection of 'Tail assy' then the documents found in the set S would satisfy the boolean equation: *Aircraft AND (Tail assy OR empennage).* This example is shown in Figure 35.

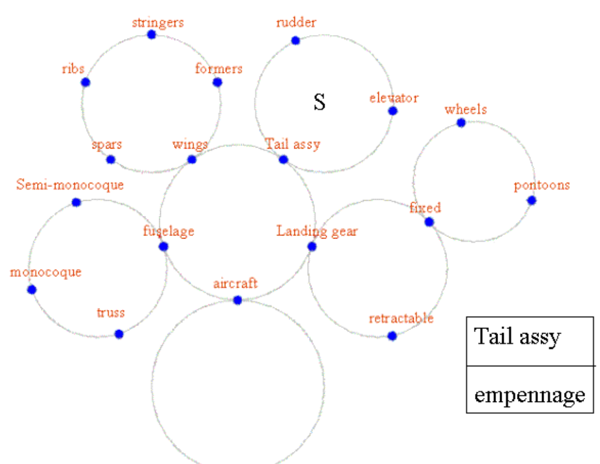


Figure 35. Adding Thesaurus Words To the Model

V. Experimental Evaluation

5.1 Introduction

New information retrieval systems are generally tested against standard retrieval strategies in controlled experiments. The experiments are generally set around standard information retrieval test collections (such as TREC [59]), and performance is evaluated based on conventional techniques such as precision and recall. There are some misgivings with this technique, however, that cause doubt as to the validity of the results. For instance, the test queries are generally constructed to represent high specificity and syntactical matching resulting in 'information tasks' rather than 'navigational tasks'. Here, 'information tasks' refer to those problems that can generally be solved with a single query. 'Navigational tasks' are those that require multiple queries to formulate enough about the schema of the problem such that it can be answered. Unfortunately though, no simpler alternative is currently available to quantitatively compare one retrieval system to another.

Attempting to test visual information systems in this manner has many problems. First, in most experimental environments in which these systems are tested, only naïve or briefly trained test subjects are used. Therefore, many of the benefits of the visual information retrieval system tend not to be effectively utilized which leads to poorer performance and satisfaction ratings than simple textual interfaces. Although poor usability design is not condoned here, the fact remains that almost all new tools require some lead time to use effectively. Second, many visualization techniques are designed as exploration tools that 'help' navigate users through the mass of information available. To this end, the tools tend to perform better in solving 'navigational tasks' rather than 'information tasks', which is contradictory to how current precision and recall evaluations are conducted. The difficulty here is how can 'navigational tasks' be specified into a query without it becoming an 'information task'? The queries themselves either become designed around the textual interface or simply become non-representative of real world problems. Third, just what metrics should be used for the evaluation is not always clear. For instance, when evaluating the performance of the visual information retrieval system, should the visual retrieval technique be removed from the visual interface, or should the system as a whole be evaluated? Should the system be evaluated against usability, task performance or both, and how does this affect the results when directly comparing systems?

The rest of this chapter is dedicated to the design of a user study in an attempt to evaluate the visual information retrieval system outlined in this research work. The next section introduces related work in the evaluation of visual systems and discusses some of the difficulties and problems encountered with each approach. The second section describes the design behind the experiment and the execution of the experiment. The third section analyzes the results obtained from the experiment. The final section discusses these results in further detail and draws conclusions based on these results.

5.2 Related Research

Usability studies in visual information retrieval systems are predominantly conducted to provide feedback for system enhancements and as such, no attempt is made to evaluate performance against another information retrieval system. Despite this, there have been a number of experiments conducted that attempt to do this with limited success. Some of the more significant studies are summarized below in the following paragraphs.

Newby [60] tested Space IR with a test population of 20 users. In this experiment, users performed two information retrieval tasks: a closed-ended question that was based on key-term synonymy and an open-ended task based on a vague statement of information need. The Space IR system was evaluated against a traditional text based IR system (Prism). Although Newby was able to demonstrate considerable learnability of the Space IR system and high user ratings, the comparisons showed that users preferred the traditional system.

Spoerri [61,62] tested Infocrystal's interface with a standard textual interface accepting boolean operators. Ten users were asked to perform query recognition tasks and query generation tasks. The result showed that new users could use the Infocrystal interface after a short tutorial and that it performed as well as the standard boolean interface. However, Spoerri identifies a difficulty in maintaining a balanced experiment, which did not disfavor the boolean mode, but in doing so actually biased the experiment towards it. The results showed a statistically significant difference between the two query languages in favor of the boolean mode.

Koshman [63,64] conducted extensive user testing on the comparisons of VIBE and a commercially available text-based information retrieval system called AskSam. Test subjects were made up of 15 novices, 12 online search experts and four experienced VIBE users. Subjects performed seven tasks that were chosen to represent 'normal' user IR tasks. However, these tasks tended to be a selection of 'information tasks' as apposed to 'navigation tasks' since the VIBE interface could not realize many of the latter. Although the scenarios were constructed to provide a naturalistic information retrieval setting, the tasks tended to be of a boolean nature e.g. how many documents contain (all, one or two) terms. Results showed that there was minimal difference in performance between the two systems, which was probably influenced by the boolean nature of the questions. Unexpectedly, the results also showed a preference for the text-based interface over the graphical interface.

Kim [65,66] conducted three experiments to examine the usability of BIRD. The first compared the correlation between the performance with BIRD and reasoning skill test scores. The second experiment examined users' performance in remembering how to use the system. The third experiment compared users' performance in constructing boolean queries using BIRD and an informationally equivalent text-based interface for both constrained and unconstrained data sets. The results showed unexpected difficulties in using the visual interface, which resulted in preferences favoring the text-based interface. A variety of features (or lack of) were identified as possibly related to these difficulties.

5.3 Experimental Design

The aim of the experiment for this research is to compare the prototype information retrieval system developed in this thesis against a representative text-based interface in a realistic 'real world' information retrieval environment. This section is dedicated to the design of that experiment. The first section outlines the hypothesis tested against to formally concretize the aim of the experiment. The second section outlines the test platform used throughout the experiment. The third section outlines the design behind the queries that made up the tasks that user's were asked to perform and the final section discusses the methodology of the experiment.

5.3.1 The Hypotheses

The aim of the experiment is to test the prototype information retrieval system against a representative text-based interface. The hypothesis for the experiment is that users of the prototype system will, with minimal training, perform at least as well as users of the more familiar text-based interface. The resultant conjecture is that if the hypotheses turns out to be true, then there is strong evidence to support the next hypothesis that user's of the prototype system will, with appropriate training, outperform user's of the more familiar text-based interface. The second hypothesis is not tested against in this experiment, as the setup and execution required is outside the scope of this research. However, the first hypothesis can be more formally defined as:

The experiment was conducted to test the following hypotheses:

$$H_0: (\mu_1 - \mu_2) = 0 \text{ against the alternatives:}$$
$$H_a: (\mu_1 - \mu_2) < 0 \text{ (lower tail alternative)}$$
$$H_a: (\mu_1 - \mu_2) > 0 \text{ (upper tail alternative)}$$

Where μ_1 and μ_2 represent the means of set 1 and set 2 respectively.

The hypothesis states that given the particular metrics to test against, the average between the two experimental groups are statistically the same. The Two-Sample t Test (explained later) is used to conduct this test. The alternatives are that they are not statistically the same, and in this case, the groups are tested against group one being statistically lower than group two (lower tail alternative) or group two being statistically lower than group one (upper tail alternative).

The experiment was conducted on two metrics:

- Time required to perform the tasks (time metric in seconds), and
- Accuracy achieved in doing the tasks (accuracy metric as a percentage).

The time metric is based on the time that the user spent solving each particular task (in seconds) and the overall time to complete all tasks assigned to the user (in seconds). The accuracy metric is a ratio between those tasks in which the user answered correctly (based on a priori results) and the overall number of tasks assigned. This result is an accuracy percentage for that particular user.

Other metrics are included for detailed analysis of the results, these include:

- Executed Queries: the queries submitted throughout the experiments by a particular user to perform the tasks given.
- Documents investigated: the documents opened by the user throughout the course of solving the tasks.
- Qualitative Assessments: the results of a survey to measure the user's thoughts on the performance of the system to solve the tasks.

Finally, the test candidates are broken down into two groups: Group 1, the control group using the text-based interface which is used to gain benchmark statistics for the experiment and Group 2, the experimental variable that is used to test the hypothesis against.

5.3.2 Test Platform

The graphical interface was implemented in the JAVA programming language (version 1.3.0) and deployed on a standard desktop computer (Intel PIII, 500MHz 256K RAM PC running Windows 2000 OS). For the experiment, a purpose built textual interface was incorporated within the graphical interface software, which had a similar form and function as traditional textual interface that one might reasonably expect to see used commercially. This included relative matching indication (based on a vector model search engine), ranked summary document listing, and keyword highlighting, but did not include phrase matching or boolean filtering.

The reason for the integration of both the textual and graphical interface was simple. It allowed for a controlled experiment where the independent variables are essentially the interfaces themselves. This mitigated the risks of uncontrolled (and undesirable) variables having an effect on the experiment and perhaps biasing the results obtained.

In addition to this, the software package was instrumented to capture user statistics throughout each experiment. For this experiment, these were limited to capturing the queries that were issued, the documents opened and the time taken for each question. These were deemed to be the minimum subset of metrics needed to test the hypotheses above.

The document collection used for the experiment was a sub set of the LATIMES data available on disk 5 of the TREC data collection. The collection of 4457 documents was indexed a priori, to produce 74516 unique keywords. This collection was selected based on its neutral subject content and its unilateral relevance. The size of the collection represented a trade off between the time required to index the collection, system performance (based on the test platform described above), and a reasonable representation of an information source.

5.3.3 The Queries

In formulating the queries for the experiment, it is important to understand the types of queries that make up reasonable representations of 'real world' situations. For instance, asking a

50-85 word query on a search engine when users are reluctant to type more than 3-5 words at a time is not reasonable. Equally, queries that tend to have perfect syntactical matching with the documents generally occur less than 20% of the time (as discussed in chapter 3), so although this is useful sometimes, again it is not a frequently occurring situation.

This leads to identifying the type of queries that can be issued. In the introduction to this chapter, ‘information tasks’ and ‘navigational tasks’ were identified. Information tasks are those that can generally be solved with a single query, they tend to have a definitive solution, and they can be analytical in nature. Navigational tasks are generally considered ‘real world problems’, in which the results of a number of queries are compared and analyzed before a final solution is reached. The former type is often used in static testing of information retrieval models and user testing because of its simplicity. The latter type is not generally tested formally through analytical means, but rather through user evaluations of operational systems.

To handle these two types of tasks, this research creates two scenario’s. Scenario 1 tasks are the ‘information tasks’ that require short and specific answers. Scenario 2 tasks are the ‘navigational tasks’ that require more involved navigation through the information collection to answer. Scenario 1 tasks have been broken down further into types based on syntactical matching and term specificity. Syntactical matching is how well the syntax of the words of the query match the document considered relevant to that query. Term specificity is how well the query semantically specifies the problem. For instance, the query ‘vehicle’ would be poorly specified, but the query ‘vehicle used in the armed robbery of the Dayton First National Bank on Saturday the 23rd November, 1996’ would be well specified. The four Scenario 1 types are listed and described below:

- **Type 1: Good term specificity and good syntactical matching.** Requires limited browsing. Usually the document is well scored and appears high in the retrieved relevance list.

- **Type 2: Poor term specificity and good syntactical matching.** Returns good hits on the documents from the query but quite often scores poorly because of poor specificity. Queries of this type generally fall lower in the relevance list and require some browsing to find.

- **Type 3: Good term specificity and poor syntactical matching.** Good description of the problem, however, only minor matches are made with the document. Again, scoring on this document is poor unless some additional techniques are used to match semantically similar terms.

- **Type 4: Poor term specificity and poor syntactical matching**. Could be the first step in the information gathering phase. Documents returned are generally not relevant, however, they help the user in refining the query into another type above.

Type 1 queries are not frequent and generally only occur if the user knows exactly what document they are after and what is contained in that document. To construct this type of query, it is a simple matter of specifying the query well and providing a good syntactic match between the query words and the document where the solution is to be found. The bolded words represent a keyword match to the document.

1	Find two documents that talk about the **electric motor car**?
2	What is the most **popular registered boat** in **Italy** and **France**?
3	Is it a **felony offense** to **donate blood** if **infected** with the **AIDS virus**?

Table 3. Type 1 Queries

Type 2 queries represent a typical query that a user might issue if they know their subject matter. That is, the query is not well specified but the keywords of the query will provide a good syntactic match. The problem with this type of query is that syntactic match does not guarantee semantic matching and if the keyword being matched is popular in the collection, the returned relevance will either be poor or hidden by irrelevant documents. Examples are shown below, where bolded words represent a keyword match to the document.

4	Find a document that talks about the **electric bicycle**?
5	Who did **Charles White** play for and what was his position?
6	Enter the query '**white house press bush'** and do not change this. Now browse the documents until you find one that talks about Lady Barbara Bush suffering from persistent double vision and swelling around her eyes.

Table 4. Type 2 Queries

Question 4 is selected because it has no answer, yet the keywords electric and bicycle are popular words in the collection. The user has to make a definitive decision to end the search hopefully before all documents are investigated. Question 5 provides good syntactic matching on the terms 'Charles White' but very little information is given about who he is or what he does. Question 6 extends this concept further by forcing the user to browse on a query that matches syntactically but is not well specified. This question is intended to model a user browsing for interesting information without knowing too much about the specifics. For instance, in this question, the user may be looking for something interesting about the Bush family that has been issued by the White House press media. The interesting article would be about Lady Barbara Bush and her medical condition.

Type 3 questions are generally very difficult to find, but represent perhaps the most common type of query that would be issued if the user had a good understanding of what they wanted, but had little knowledge of the specific terms that may be used. That is, the problem is generally well specified but there is poor syntactic matching of the keywords. In this case, users generally have to try and improvise by guessing which query terms may give good document returns. Examples are shown below, where the bolded words represent a keyword match to the document.

7	Find a document about a lady who commanded **troops** into **battle**?
8	A well known deep **sea** adventurer collaborates with a **revolutionary** photographer to find a vessel destroyed in WW1. What was the name of the photographer?
9	There are two documents that have the words '**president kennedy assassination**' but actually make no references to President Kennedy's Assassination. List these two documents.

Table 5. Type 3 Queries

Question 7 and 8 above show well specified problems but with very few key word hits in the document. This is done by carefully changing the syntax of the problem but keeping the semantics the same. For instance, in question 7 of Table 5, 'lady' and 'commanded' could be replaced with 'woman' and 'command' to achieve a higher syntactic match. Question 9 shows an example of a query that is well specified but has poor syntactical matching to differentiate between the documents that are about President Kennedy's assassination, and the documents that are not.

The last of the query types for Scenario 1 is Type 4, those that have very poor syntactic match and are not at all well specified. This type of query is generally produced when users begin with a vague idea of what they are searching for and with no domain knowledge to help with syntactic matching. Needless to say, this type of query generally never produces a good relevance list for the user, however, it does help the user to re-specify the query iteratively until the query is of one of the other three types. For instance, a user specifying the term 'wireless' may not have a good relevant list of documents returned, but in browsing the documents that do return, the same user may now try 'radio' or 'wireless radio' with more success. Type 4 queries will not be tested here directly, because the nature of the problem is such that they are difficult to specify without inadvertently making it one of the other three types. The Scenario 2 questions, however, will encompass some Type 4 search methods, and for this reason and to simplify nomenclature, Scenario 2 questions will be referred to as Type 4 queries from this point forward.

Scenario 2 (Type 4) questions are those that require more complex schemas than those of Scenario 1 tasks. That is, users are required to browse information to find answers to problems, and then by piecing these answers together, are able to answer larger questions. Again, these are difficult questions to formulate and are likely to have somewhat limited success in a formal testing environment. The reason for this was alluded to previously, in that by formulating the question the user inadvertently learns something about the problem, which then guides the user into search patterns rather than browsing patterns. Having said that, it is still important to investigate this type of task in this environment as a precursor to further user evaluation testing. Scenario 2 (type 4) examples are shown in Table 6:

10	You have been asked to do an assignment of three United States presidents. To make research easier, you will pick the three that appear to have the most documents mentioning them (and therefore the most research material). To this end, find at least five presidents and determine which three of these presidents appear to be mentioned in the most documents.
11	Are there any documents that mention all three of the presidents that you chose together? If so, list up to five documents mentioning all three presidents in the same document?
12	To finish off, you feel it would be nice to know the first name of the wife of the presidents that you have chosen. If you are able, list these.

Table 6. Type 4 Queries

Again, all three questions are related, and require some browsing to find information to piece together to solve the larger problem, however, there are some problems with this type of question. Users with some knowledge of American history may try to skip the browsing portion of the question and move on to the search portion immediately (effectively answering this question as a number of Scenario 1 queries). This problem is somewhat mitigated by the fact that the collection is limited in size (approximately 4500 documents) and covers only a very short period of time (1989-1990). Therefore, searching in this manner may not provide good results. Secondly, the answer to all three questions may differ from one user to the next depending on the five presidents listed in question 10. This is fairly typical of what happens with ‘real world’ browsing since all users are different and have different backgrounds. However, having said this, this is still useful in observing how users browse for information and whether the graphical interface is able to assist in this process.

5.3.4 The experiment

5.3.4.1 Introduction

Before outlining the experiment, it is useful to investigate some of the risks that may adversely affect its outcome. First, using naïve users for testing a graphical interface has problems. As mentioned before, it is unreasonable to expect users to perform well with any new package until they have had a chance to become familiar enough with it to be considered proficient. This, however, is almost always impossible to achieve since testing time is generally very limited. To mitigate this risk, the test subjects were selected from the Electrical and Computer department of the Air Force Institute of Technology (AFIT) graduate programs (master students). Therefore, all test subjects should have had an advanced knowledge of computers and be proficient at using traditional text based information retrieval engines. In this way, only the graphical information retrieval package was unfamiliar to the subject, which reduced the lead-time before the experiment. Next, training packages in the form of instructional videos were used to teach the user about the interface prior to the experiment. Again, this was limited in time and usefulness, but allowed for a consistent approach. Finally, the users were given some time to use the basic functions of the interface prior to the experiment to increase familiarity.

Second, since there are only a limited number of test subjects that can be called on for user trials, it would have been advantageous to have each user perform the textual and graphical part of the experiment. This had two major problems, the testing time required for each subject now doubles and any residual knowledge on the first test will taint the results of the second test. Therefore, test subjects were randomly chosen to perform only one of the experiments, either the textual part or the graphical part.

Third, the questions that make up the experiment may bias either interface. As has been the experience of others before with similar comparison testing, the questions can bias the textual interface if they are chosen to ignore many of the advantages of the visual systems or they can bias the graphical interface if the converse is true. This research has addressed this problem by carefully identifying the type of queries that represent ‘real world’ situations. The queries were chosen by type, as discussed in the previous section, and not by a particular function that can be emphasized. To mitigate this risk further, users were asked to fill out a questionnaire at the conclusion of the experiment, in which one question asks specifically whether they though the questions represented ‘real world’ situations.

Fourth, to control the variables of the experiment, the textual interface was incorporated within the graphical interface package. This essentially means that the underlying search engine, ranking models and any other similar functions are consistent between both interfaces. The only difference is how the data is displayed, and the interaction with the data that is provided. The obvious problem with this approach is that the textual interface provides only basic functions for searching, that is, no phrase or proximity matching is provided and no boolean filtering is available. This is satisfactory for this experiment because the functionality that is provided represents that typically used by average users of mainstream information retrieval engines. That is, relevance scoring, summary listing, keyword highlighting, limited list sizes, and links to the full document text on demand are all provided. This is not to say that either method couldn’t be improved by adding any particular function to it, but this is not the purpose of this experiment.

Fifth, ontology models, although a large part of this research work, were not used in all but one question of the experiments. This was to ensure that the results were not unfairly biased towards the graphical interface due to this assistance. Question 9 (of the test sheet) was the only question that had a simplistic model associated with it (‘lady’ was mapped to ‘women’ and ‘female’). This was to show that ontology models could be seamlessly integrated into the process without user knowledge, and to see if limited assistance such as this could significantly change the results for the better. This question is removed from the overall results and is investigated separately.

5.3.4.2 Method

Test candidates were obtained from volunteer students in the Engineering and Management School of the Air Force Institute of Technology (AFIT). Each candidate was selected for the textual interface component of the experiment or the graphical interface component of the experiment based on 'walk in' random selection. The candidate was required to view a short video describing the purpose of the experiment, functions of the interface they were about to use, and how the experiment was to be conducted. The video for the textual interface was a five minute presentation, and the video for the graphical interface lasted 20 minutes. Appendix A describes the tutorial in detail by showing a text version of the presentations given, and screen shots of the examples used. After the presentation concluded, the program was run and the appropriate interface was presented to the candidate. The candidate was then asked if they felt comfortable with the interface they were about to use and whether they had any questions about that interface. They were then allowed to experiment with the interface until they were happy to proceed. This generally took a few minutes for the textual interface and 5-10 minutes for the graphical interface.

The candidates were told that the test was one of accuracy but that they would also be timed. They were also told that some questions would be easy to answer, some more difficult to answer, and some had no answers. If at any time the candidates believed that the answer was not in the collection given, they could indicate this on the question sheet and move on. If they felt they simply could not find the answer, they indicated this and moved on. At this point, the experiment was initiated. This involved candidates typing their last name at the beginning of the experiment (to set up the log file) and then, sequentially reading and completing each question prior to moving on to the next. Throughout the trial, each candidate was logged for the queries issued, documents opened and time required to answer each question. At the end of the experiment, the candidates were asked to fill out a short questionnaire. The questions used for the experiment, the testing answer sheet, and the questionnaire can be found at Appendix B.

5.4 Experimental Analysis

5.4.1 Preliminary Discussion

The results of the experiment are presented in Appendix C. Question 12 was removed from the analysis for two reasons. First, the logger used to capture statistics during the experiment did not capture this question on the first four candidates. After this, the error was detected and fixed. Second, the difficulty of the question was directly proportional to the answers given in Question 10 and 11. That is, one subset of answers would lead to a simpler answer than another subset. It was observed that candidates either brushed over this last question quickly, or spent considerable effort trying to find an answer that wasn't there. For this reason, it was removed.

Some outlier results were also removed. In the graphical interface test results, two candidates scored accuracies significantly lower than the rest. Both these candidates did not speak English as their first language so these results were removed. In the textual interface results, one candidate scored an accuracy significantly lower than the others because accuracy was traded for speed. Since the candidates were clearly told that the test was accuracy based, this result was also removed.

Question 9 (from the test sheet) was also removed from the overall results because an ontology model was associated with the question. As described above, this could possible bias the result towards the graphical interface, so it was removed. This question is analyzed separately later.

5.4.2 Two-Sample t-test

The following statistical designs and formulas used to analyze the experimental results are based on the text book by Mendenhall, Wackerly and Scheaffer [67]. For the following analysis, the two-sample t test was used because of the low sample collection of the experiment. That is, this test is somewhat more robust against the assumption of a normally distributed data set for small collection sizes than say the z-curves.

In this analysis, the results of the graphical interface and the textual interface experiments can be tested against the null hypothesis H_0: $\mu_1 - \mu_2 = D_0$ for some fixed value of D_0. If $D_0 = 0$, then the test will show if there is sufficient evidence to indicate a difference in true mean times between the two samples taken.

Test H_0: $(\mu_1 - \mu_2) = 0$ *against the alternative* H_a: $(\mu_1 - \mu_2) \neq 0$.

The t-test statistic is given by:

$$T = \frac{(\overline{Y}_1 - \overline{Y}_2)}{S\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \qquad (20)$$

where Y_i is the sample mean of the population set i, and the variance S is given by:

$$S^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2} \qquad (21)$$

where S_i^2 is the variance of the population set i, and n_i is the population size.

5.4.3 Mean Average Times

Testing at the $\alpha = 0.05$ level of significance, the two-tailed test statistic for average times between the samples is calculated as follows:

$Y_1 = 3122$ seconds (mean average of group 1 total time)

$Y_2 = 2979$ seconds (mean average of group 2 total time)

$S_1^2 = 1405807$ (variance of group 1 total times)

$S_2^2 = 127302$ (variance of group 2 total times)

Therefore, the two group variance S is calculated as:

$S = 966$ where $n_1 = 9$ and $n_2 = 8$

The t-test statistic is calculated from the above information and equation to yield:

$T = 0.286$

This value does not fall in the rejection region $(|T| < 1.771)$ and hence the null hypothesis is not rejected. There is not sufficient evidence to indicate a difference in the mean between the two results. Therefore, there is no statistically significant difference between the two sample mean times, which supports the null hypothesis.

5.4.3.1 Mean Average Accuracy

Testing at the $\alpha = 0.05$ level of significance, the two-tailed test statistic for the average accuracy values between the samples becomes:

$Y_1 = 0.76$ (mean average of group 1 accuracy)

$Y_2 = 0.89$ (mean average of group 2 accuracy)

$S_1^2 = 0.004132$ (variance of group 1 accuracy)

$S_2^2 = 0.008855$ (variance of group 2 accuracy)

Therefore, the two group variance S is calculated as:

$S = 0.0855$ where $n_1 = 9$ and $n_2 = 8$

The t-test statistic is calculated from the above information and equation to yield:

$T = -3.13$

This value falls outside of the rejection region $(|T| > 1.771)$ therefore the null hypothesis is rejected. There is sufficient evidence to indicate a difference in the mean between the two sample mean accuracies. Furthermore, it is a lower tail alternative H_a: $(\mu_1 - \mu_2) < 0$ providing support that the mean accuracy for the results of the graphical interface is statistically superior to that of the results for the textual interface.

5.4.4 Additional t-test

In this final test, and for the purpose of further discussion later, only the mean average times for Type 1, 2 and 3 queries are analyzed. For this subset of results the following is calculated:

$Y_1 = 2372$ (mean average times for group 1 Type 1, 2 and 3 queries)

$Y_2 = 2015$ (mean average times for group 2 Type 1, 2 and 3 queries)

$S_1^2 = 905850$ (variance of mean average times for group 1 Type 1, 2 and 3 queries)

$S_2^2 = 160840$ (variance of mean average times for group 2 Type 1, 2 and 3 queries)

Therefore, the two group variance S is calculated as:

$S = 558178$ where $n_1 = 9$ and $n_2 = 8$

The t-test statistic is calculated from the above information and equation to yield:

$T = 0.983$

This value does not fall in the rejection region $(|T| < 1.771)$ and hence the null hypothesis is not rejected. There is not sufficient evidence to indicate a difference in the mean between the two results. Therefore, there is no statistically significant difference between the two sample mean times, which supports the null hypothesis.

5.4.5 Analysis and Discussion of Results

Put simply, the results of the previous section imply that the differences in times taken between the two sample sets are not statistically significant, but that the differences in accuracy are. This is useful as an overall statistic, but of more importance here is to understand exactly what this means based on the questions asked and the query types.

5.4.5.1 Comparison By Times

Figure 36 shows the direct comparison between the average time results of candidates using the textual interface versus the average results of the candidates using the graphical interface.

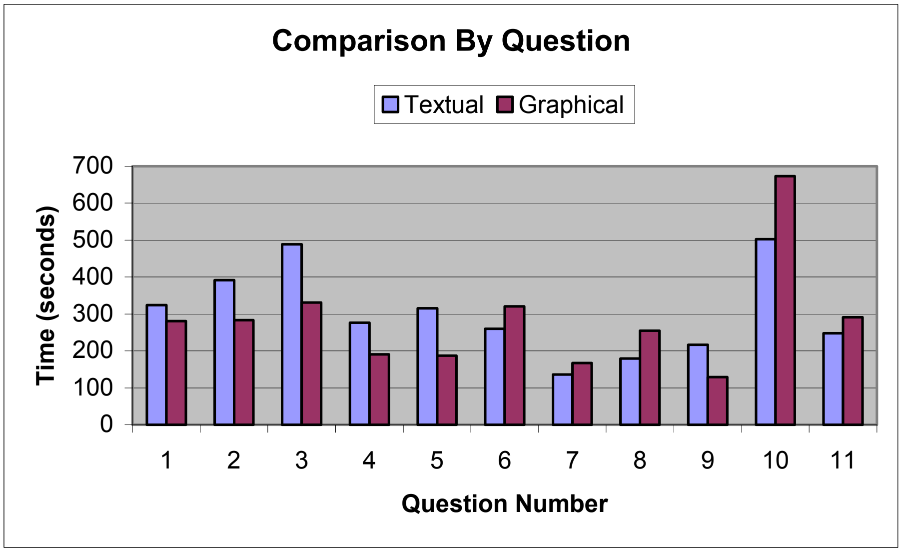


Figure 36. Comparison of Average Times by Question

Figure 37 shows the direct comparison between the average time results of candidates using the textual interface versus the average results of the candidates using the graphical interface for each query type.

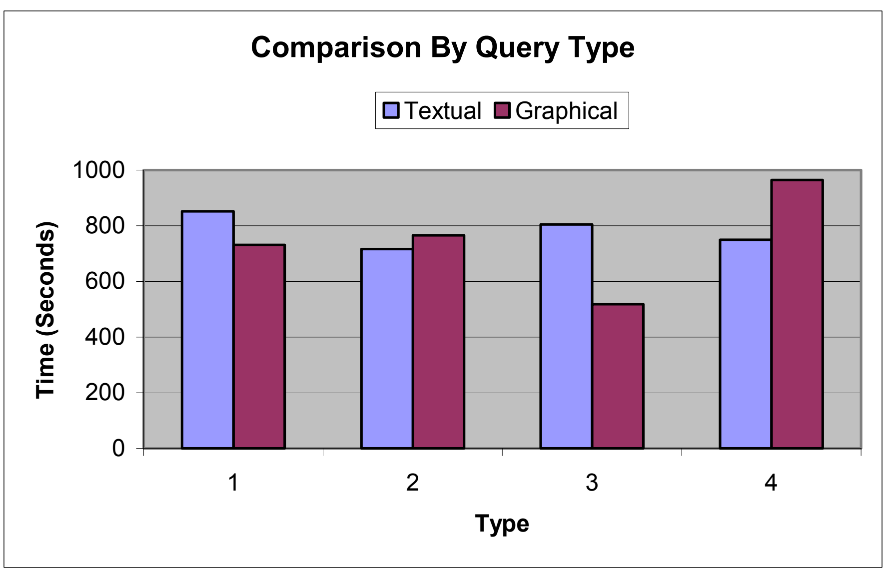


Figure 37. Comparisons of Average Times by Query Type

The results show that there was a significant decrease in time required to answer type 1 and type 3 queries for the graphical interface compared to the textual interface, a comparatively equal result for type 2 queries and a significant increase in time to answer type 4 queries (which is defined as Scenario 2 task).

The results of the type 1 queries are interesting. These queries represent those that have good term specificity and good syntactic matching (as described above) which generally allowed the documents to rank well in the relevance list. It was initially thought that documents ranking high in the document list would be found quicker than selecting icons from the graphical display, however this was not the case. The reason for this was that documents matching the type 1 queries were displayed with higher order icons (i.e squares, pentagons, …) which are easily distinguishable across the collection. Therefore, the number of documents that were scanned for content (by the user) was generally fewer than in the textual interface. For example, in the ranked list of the textual interface, a document with five keyword hits may rank lower than a document with four resulting in the later document being scanned before the former.

The type 2 queries showed similar results for both the graphical and the textual interface. Type 2 queries are those with poor specificity but good syntactic matching. This was to be expected since the relevant documents are easily 'lost' in the noise of the collection. For instance, in the textual interface, the document may rank low in the relevance list especially if the small number of keywords that match are popular in the collection. A similar problem occurs in the graphical interface. The document can easily get lost in the noise of lower order icons (say rectangles) that have large document collections within them. With experience, users may have been able to reduce the search time by considering spatial location of the icon more closely than they did. For instance, question 6 asked the following:

*Enter the query '**white house press bush**' and do not change this. Now browse the documents until you find one that talks about Lady Barbara Bush suffering from persistent double vision and swelling around her eyes.*

When the query was executed in the Bubble World interface, the result shown in Figure 38 occurred.

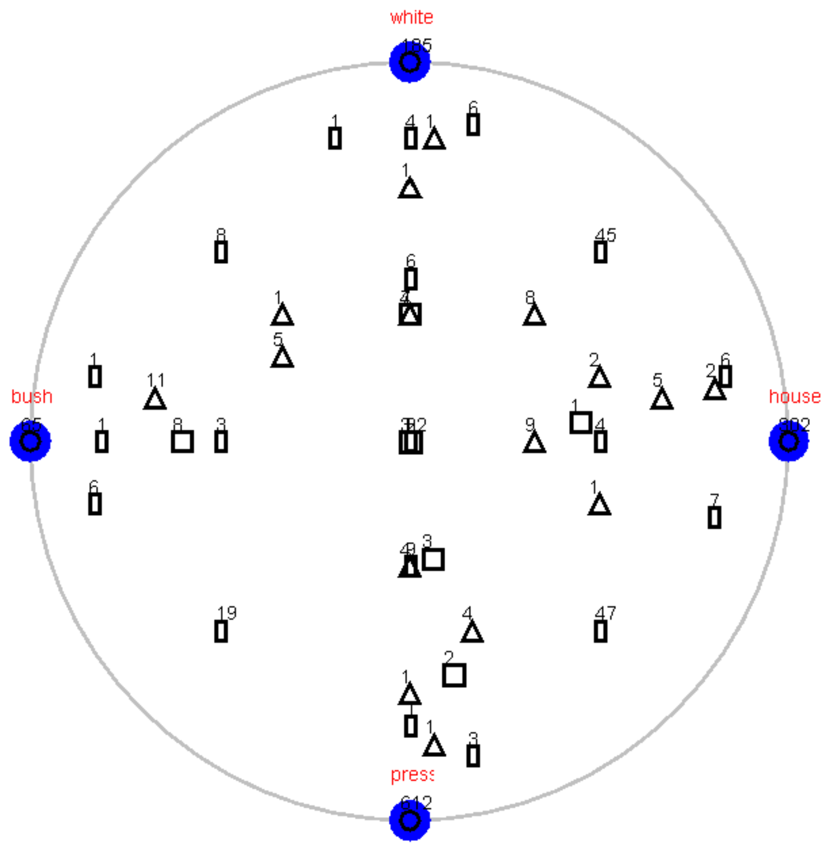


Figure 38. Result of Issuing Question 6

Most users attempted to answer this question by selecting the squares (almost randomly) from the circle above. In doing so, quite a number of documents were scanned before the document containing the answer was found. However, if one considers that the question is about Lady Barbara Bush, one might conclude that the term 'Bush' may be a significantly weighted term.

Therefore, the square found close to the node 'bush' may have been a more appropriate icon to select than any other. The document that contained the answer was indeed found in this icon. Again, deductions of this type come more naturally to experienced users of this interface than inexperienced ones.

Type 3 queries showed the most significant reduction in time between the graphical interface and that of the textual interface. Type 3 queries are those that contain good specificity but poor syntactical matching. Again, this generally resulted in the relevant document being ranked poorly across the collection. This type of query generally required the user to manipulate the query, perhaps incrementally, until keywords are struck that allow the document to rank higher in the relevance list. The results showed the users of the graphical interface were able to see the results of this incremental change more clearly than those using the textual interface. That is, words that had little or no relevance to the document collection could easily be distinguished and therefore replaced with other words until the document ranked well enough to be promoted to higher order icons, thereby becoming more visible. Those using the textual interface did not have this luxury and relied on having to scan a large list of documents for every incremental change of the query, resulting in considerable frustration and fatigue for the user.

The last of the query types, Type 4, provided interesting results as well. These queries required multiple schemas to piece together the overall solution. This type of query resulted in a 28% average increase in time for those candidates using the graphical interface to those using the textual interface, which by any means is a significant result. The reason for this came down to inexperience by the user. The users, although explained in the instructional video, did generally not attempt constructing multiple schema queries in the graphical interface, or if they did, it was not done effectively. In fact, most of the users that attempted this problem in the graphical interface tried to solve the problem using a single schema (single bubble). By guessing the answer they attempted to remove the browsing component of the problem and go directly to the search component. The problem with this technique is that it cluttered the search space and made it difficult for the user to ascertain what was relevant, and what was not. An example of one of the queries issued by a test subject is shown in Figure 39. Although the information is visible, it is not clear on exactly how to proceed from this point.

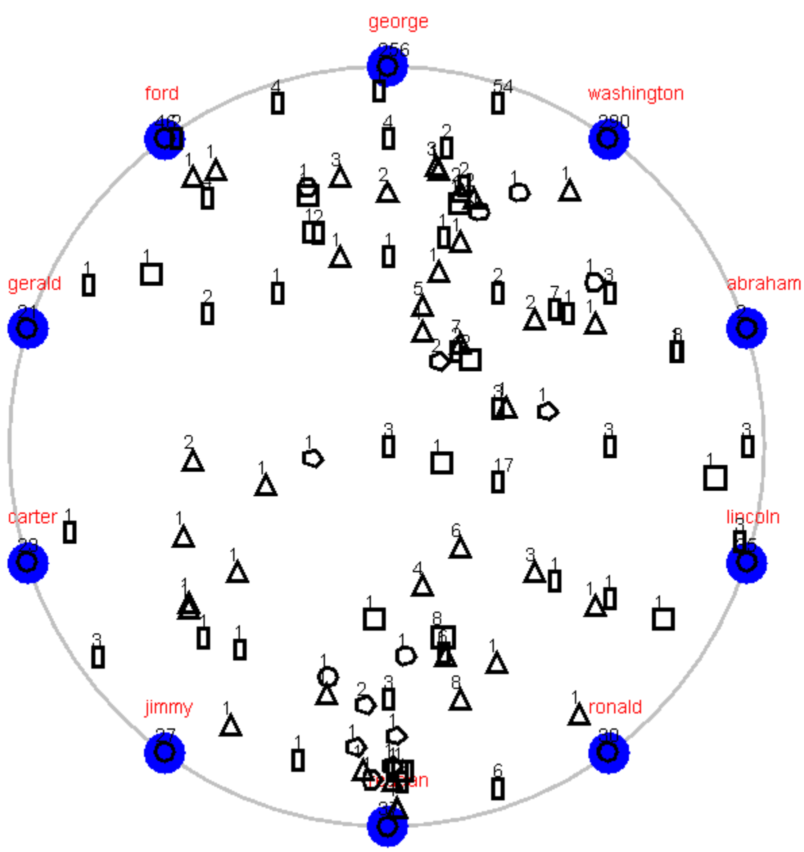


Figure 39. Test Case Example, single schema

The next example from another test subject shows the use of multiple schemas to answer the same problem as that in Figure 39. This is shown in Figure 40, modified from the original query to show a single representation. In this case, multiple schemas provided a clearer view of the problem and all three questions can be answered from this single view.

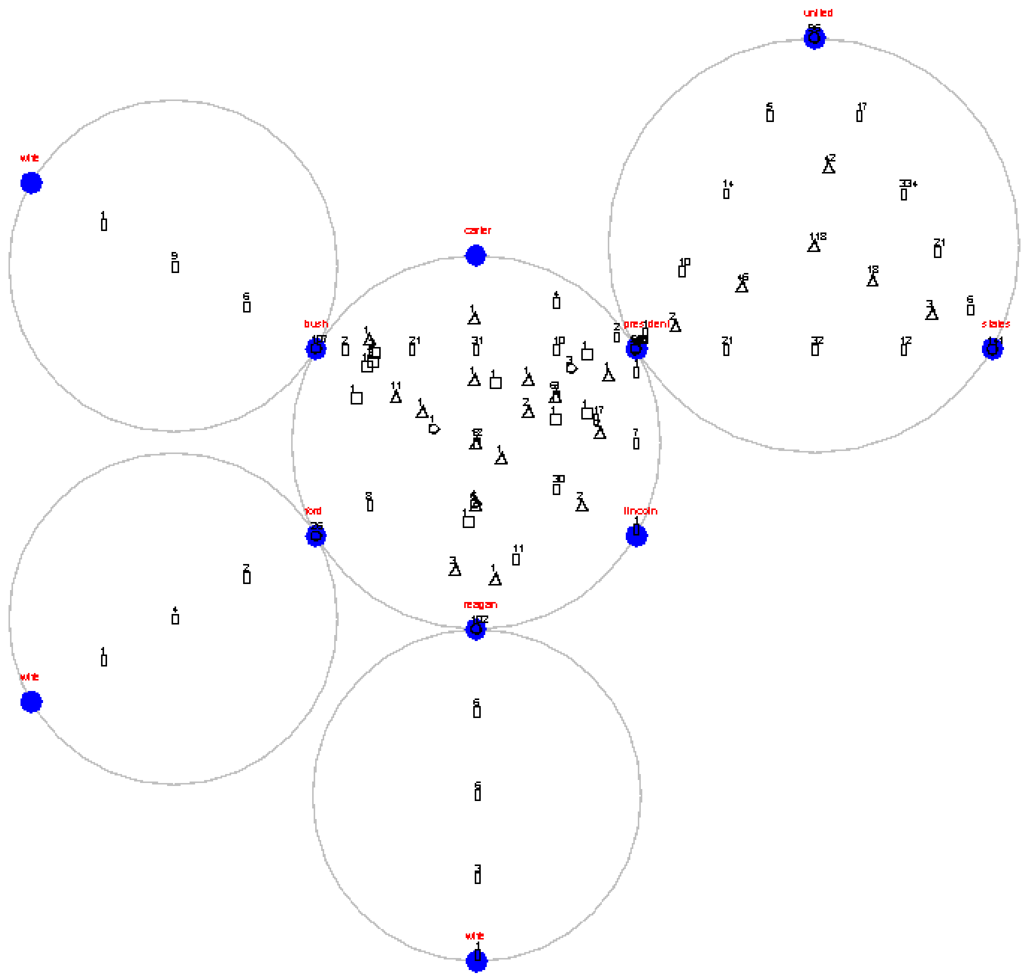


Figure 40. Query used to answer Questions 10, 11 and 12

5.4.5.2 Comparison By Accuracy

Figure 41 shows the direct comparison between the average accuracy results of candidates using the textual interface versus the average results of the candidates using the graphical interface by question. Figure 42 shows the same comparison but this time by question type.

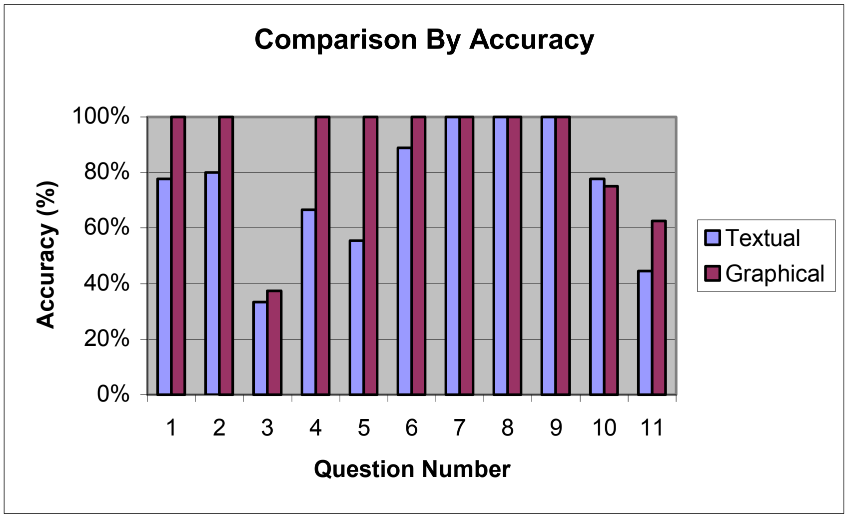


Figure 41. Comparison of Accuracy by Question Number

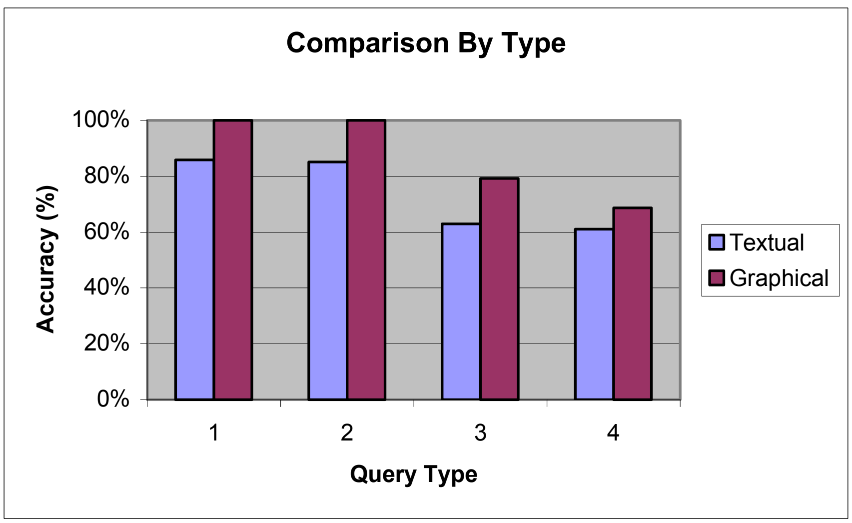


Figure 42. Comparison of Accuracy by Question Type

It is clear that the accuracy of all query types for the graphical interface is significantly higher than that of the textual interface. Of particular interest here is the fact that both Type 1 and Type 2 queries show an accuracy of 100% for the graphical interface. This shows that with good syntactic matching, the relevant documents are promoted well enough to higher orders of icons to be found. The only variance here was the time taken to find the document.

5.4.5.3 Inclusion of Ontology Models

To this point, the inclusion of ontology models has been ignored because it is impossible to include this in the graphical interface when direct comparisons are being made between it and the textual interface. It may have been useful to add limited features of these models into both interfaces, however, that is not the purpose of this research.

Examining how ontology models could be seamlessly integrated into the query without user knowledge is an important observation and one worth investigating. For this purpose, Question 9 (of the question sheet) was included in the experiment but not in the overall summary results. This question is shown below for clarity.

Question 9: Find a document about a lady who commanded troops into battle

In this question, the word 'lady' was mapped to 'women' and 'female', so in a strict sense this was nothing more than adding a thesaurus. However, users were free to formulate any query they wished, and were not told of the significance of the word 'lady'. Furthermore, the assistance provide only one additional keyword match for the total query. The results of Question 9 are shown below:

Average Time for Textual Interface:	217 seconds
Average Time for Graphical Interface (with model)	129 seconds

A 40% reduction in time even for moderate assistance such as this!

Note that this result is far from conclusive. Its usefulness is simply to show that the model can be incorporated without user knowledge, and this assistance can help without any additional overhead being added to the user. An interesting observation to note was that not all users chose to use the term ‘lady’. Some users immediately substituted the word for ‘woman’ and others chose not to use the term at all.

5.5 General Discussion

It is evident from the results previously discussed that although there was no statistically significant improvement in the time required by users of the graphical interface, there is a significant improvement in the accuracy of the solutions. A reasonable conclusion to draw from this is that users of the graphical interface appear to be more willing to keep searching for the solution as long as they think they have a chance to find it. The graphical interface is able to do this by showing the whole collection space (as pertained to the query). While relevant icons are still not searched, the user is willing to continue. Contrast this to the textual interface where after searching through tens of documents, the user is not sure whether continuing the search would help, or whether there really is no answer.

Another observation made with the study is that users tend to experiment more with the queries in the graphical interface. The user logs revealed that, on average, users of the graphical interface issued 46 queries across the 12 questions compared with 32 queries for the textual interface. This was probably due to the fact that users could obtain more meaningful feedback from the graphical display than from the textual list and were therefore more willing to try new queries. This is one explanation as to why the accuracy varied between the two systems so significantly. The fact that the average time difference between the two groups did not change significantly, even with an increase in the number of queries issued, suggest that users were able to make quicker inferences from the results of each query than they were from the textual list.

As discussed before, there is a significant increase in the average time (+28.5%) to solve Type 4 queries in the graphical interface over the textual interface. Although it was thought that the graphical interface would outperform its counterpart in these queries, the results are not surprising. Constructing the nested queries represents the most difficult task required of the user, and knowing just how to nest the queries effectively takes a little experience. Investigating the user logs, of the

10 candidates that used the graphical interface, only five attempted nested queries as instructed in the training video, and of those, only two did so effectively. The rest attempted to answer the question in a similar fashion to those using the textual interface. That it, guess the answer and search for documents using a single schema approach. The two candidates that did use the nested queries effectively answered the question correctly and did so with a 16.9% improvement in time over the overall average for the group using the graphical interface. If Type 4 questions are removed from the analyses of mean average times and the two groups again compared, then there is a real mean average decrease in time of (15%) from the group using the graphical interface to the group using the textual interface. Note that although this is still not statistically significant (see section 5.4.4), it is expected that increasing the population size of the experiment would have decreased the variance and ensured a more normalized set of data. From this discussion, it is reasonable to postulate that an increased effort in training (more than the 25 minutes afforded in this experiment) would result in a reduction in the mean average time of the users. Given that the test candidates can be considered expert computer users, it is unlikely that the same additional training in the textual interface would result in any increased performance.

5.6 User Feedback

At the end of each experiment, test candidates were asked to fill out a short questionnaire about the experiment. The first four questions were rated from 1 to 7 where 1 represented the least favorable position and 7 represented the most favorable position. Question 5 was a yes/no answer, and Question 6 required general comments. The results of this questionnaire are summarized and discussed below.

Question 1: How would you rate your proficiency in using information retrieval search engines (such as Google)?

Rating:	Textual:	5.4
	Graphical:	5.6

The purpose of this question was to ensure that there was no biasing of experience between the two groups tested. The results of this question suggest that this was the case.

Question 2: How well do you feel that you were able to answer the questions using the interface provided?

Rating:	Textual	4.7
	Graphical	5.4

This rated the perceived usefulness of the interface to answer the questions. The lower rating for the textual interface resulted in the frustration of having to search for documents that did not rate well with the query (ie. Type 2, 3 and 4 queries). The higher rating for the graphical interface indicates that these users felt that they had more control for answering these questions than users of the text-based interface.

Question 3: Do you feel that the questions represented reasonable 'real world' problems that one might use a search engine to find?

Rating:	Textual	5.3
	Graphical	6.2

This question was designed to get some feedback on how relevant users thought the questions were. The aim of the experiment was to try and match a more realistic 'real world' scenario to gain a perspective of how well the graphical interface might do in this environment. The results show a positive response from both groups, however there were some negative comments about question 6. Some users felt it was unreasonable not to be able to change the query, however once it was explained to them that it was crafted to test a particular concept, they seemed satisfied.

Question 4: What was your overall feeling on the usefulness of the interface you used?

Rating:	Textual	3.3
	Graphical	6.4

There was a significant difference in attitude between the two groups in the response to this question. Those using the textual interface believed that the documents were not ranked well enough, and would have preferred to see some additional techniques such as proximity filtering and phrase matching incorporated into the interface. They did feel, however, that the keyword matching and document summaries helped. Those using the graphical interface were extremely positive and believed that the interface was a definite advantage in answering the questions. Some comments about phrase matching were made on this interface as well.

The observations made by the textual users were accurate. Many documents did not rank well because they were not Type 1 queries, therefore requiring users to search for them. Also, because of the limited collection size, there were far fewer partially relevant documents which again made searching more difficult. Overall though, users did feel capable of answering the question in both environments.

Question 5: Would you use this interface again if available as a front end to a digital library or web search engine?

Rating:	Textual	50% said yes
	Graphical	100% said yes

This question was essentially targeted at those that used the graphical interface, however the comparison between the two is useful. Half of the users using the textual interface indicated they would not use this interface again because others were available with more advanced functions. Again, the difficulty of query types had a significant effect on this result. In contrast, every user of the graphical interface said they would definitely use the interface again and believed it was very easy to use. There were no negative responses from the group that used this interface.

Question 6: Do you have any other comments about the interface that you used?

Many of the users indicated that phrase matching would have enhanced both interfaces. Some suggested that hyperlinks from the document summary would have been useful. A couple of users indicated that they popped bubbles when they really wanted to pop nodes. Other suggestions including capabilities for dragging the lens in the document view instead of having it follow the cursor and for having the frequency charts labeled so that it would be easier to see which document they represented. Overall, the comments for the graphical interface were positive such as “nice”, “cool idea” and “on to a good thing”.

5.7 Conclusion

It was not possible to test all of the useful features of Bubble World in this user trial since many of these would simply have biased the results, however, the experiment did highlight some useful results. First, it was shown that even after only a brief introduction to the graphical interface (lasting only 20 minutes), users were able to outperform their counterparts that used the textual interface. Second, although statistically there is no evidence that a reduction in time was achieved using the graphical interface, there is strong evidence to suggest that this was the case and that the additional time was used to increase the accuracy of the results. Also, based on the examination of Type 4 queries, there is strong evidence to support the notion that additional training and familiarity would improve the results even further. Third, although ontology models were not added to the results of the experiment, it was shown that these models can be seamlessly integrated into the search of the user without additional overhead imposed on them and, it appears that this results in significant improvements to the search pattern of the user. This result has promising implications for future research. With correct domain (or user) profiling, these models could be used to significantly enhance relevant document retrieval. Fourth, the feedback from the experiment was extremely positive with regards to the graphical interface. This suggest that not only do users believe that the interface would significantly help them in retrieving relevant documents, but also that the interface was intuitive and easy to use. These are encouraging responses for the first version of Bubble World. Fifth, the experiment provided some useful feedback about the software package itself, and highlighted some possible future improvements.

VI. Conclusion and Future Research

6.1 Conclusion

Today, information domains are expanding at an alarming rate. The freedom of the general public to access and publish this information has rapidly become mainstream and has undeniably been the catalyst for this enormous growth. It has also created a paradigm shift from well-formatted information into essentially a 'free-for-all' collection of text and multimedia. Along with this enormous growth of information comes the expectation of users to be able to retrieve relevant material effectively and efficiently.

Traditional information retrieval techniques have been around for some time and although they come in many different styles and formats, they essentially all reduce to matching a query to an information collection and then displaying that collection as a list of documents in rank order. The misgivings with this approach have always been that their underlying interaction model does not align with the cognitive process of the person using it. That is, the user has always had to map internal representations of the problem into a one dimensional, single schemata process.

The last decade has seen many new visual information retrieval systems emerge into the research arena, each with its own unique style and focus. The reason for this is simple: visualization offers a much larger bandwidth channel between the user and the system than traditional text based systems. Therefore, many rely on this to transfer high data densities in an attempt to amplify the users cognitive thought processes to effectively improve information retrieval. To date many of these systems have not been formally tested against traditional techniques, and those that have, have not delivered results that would suggest a paradigm shift in information retrieval any time soon.

On one hand, users are comfortable with text-based interfaces even though they provide very little cognitive amplification to assist in the information retrieval process. On the other hand, visual interfaces exploit the large bandwidth channel of the human visual system to assist users in forward browsing, but users are generally unable to use them more effectively than text-based interfaces. This research begins by identifying why this calamity exists. It is theorized in this thesis that this is due to the following fundamental issues, (1) that many visual information retrieval systems lack a solid cognitive framework to efficiently and effectively map internal mental models of the information retrieval problem to an external view, (2) that their interaction models are not based on a natural extension of how humans solve problems, and (3) that observed user interaction patterns with existing information retrieval systems seldom influence the design of the next visual system. For this reason, considerable emphasis is placed on this component of the research, the end result of which is a generic framework of cognitive propositions and design corollaries that provide a useful guideline to the development of any information retrieval system.

With a solid framework to base the design on, the next stage of this research was to develop a visual technique that transforms the internal mental representation of the information retrieval problem to an isomorphic external view, and then through visual cues, provide cognitive amplification at key stages of the information retrieval process. Understanding the key stages of information retrieval came from discovering how the knowledge crystallization process of the human mind integrates into the interaction model of the user. Once this is known, providing the mechanisms to incorporate complex search schemas into the retrieval process, both manually and automatically through the use of pre-defined ontological models, is a natural extension of this search and browse process. Using these predefined models, a domain expert is now able to guide users to search relevant areas of the information collection that may not have been immediately visible to the user.

Bubble World is the prototype system developed around this new and powerful visual information retrieval technique. The environment allows simple and complex nested queries to be specified visually and the system then integrates the resultant returned relevance set back into the query relationships. In this way, it is both a visual tool and a visual query language. The system is capable of maintaining an overall spatial overview of the relevant set from a large data collection while still providing visual cues about the contents of that data. This is achieved by grouping 'similar' documents into specially coded icons and then, spatially moving these icons into equilibrium positions (position of zero net attractive force on the icons). Complex nested queries can be defined to create individual schemas of the search problem. Each schema replicates the root query view of the data collection through a series of boolean and/or proximity filtering mechanisms. This does several things for the user: (1) it forces documents with a higher probability of relevance to filter to the ends of the schema, (2) it excludes documents with lower probability of relevance to replicate across the schemas and (3) it provides the mechanism for the user to explore new configurations, maintain consistent and progressive inferences about the collection, constrain their visual attention shifts and direct attention to only the key components of the search that are useful (or essential) at that time, thus providing cognitive offloading. The interaction model provides the mechanisms to allow efficient mutable changes to the query to allow the user to search and browse, thereby promoting quick and easy exploration of the information space. Throughout this search/browse phase, users are encouraged to develop problem specific schemas to facilitate perceptual parsing and inferencing on the data collection.

With the visual techniques defined and the prototype system developed, the research moves focus to the experimental verification and validation phase to demonstrate their effectiveness. This is achieved in the following manner. First, background research is performed on experiments previously conducted with similar aims, in an attempt to pollinate those ideas into this research. Second, the query construct itself is examined to identify what makes a 'real world' scenario in information retrieval. This leads to a representative set of twelve 'real world' queries across four different query types, which become the foundation to the experiment. Third, the experiment is designed around these test queries and conducted on two select groups: the control group, which use the text-based interface to perform retrieval tasks and the experimental group, which use the new visual interface to perform the same tasks.

The user study was conducted to compare a representational text-based information retrieval system with Bubble World. The test candidates were required to retrieve information based on a number of queries using either one of the two interfaces. Although the scope of the test is unable to explicitly test many of Bubble World's novel features (such as automatic query and schema expansion), the experiment is useful in a number of ways. First, it shows that users of Bubble World could outperform users of the textual interface with minimal training, in the particular environment tested. Second, it provides positive feedback in the usability of the system. The key findings of the study are summarized as follows:

- Although there is no statistically significant improvement in the time required by users of the graphical interface, there is a significant improvement in the accuracy of the solutions. A reasonable conclusion to draw from this is that users of the graphical interface appear to be more willing to keep searching for the solution as long as they think they have a chance to find it.

- Users tend to experiment more with the queries in the graphical interface. This was probably due to the fact that users could obtain more meaningful feedback from the graphical display than from the textual list and were therefore more willing to try new queries.

- An increased effort in training is likely to result in a reduction in the mean average time of users of the graphical interface, but it is unlikely to result in a reduction in the mean average time of users of the textual interface.

- Users of Bubble World were all extremely positive about the interface they used and believed that the interface provided a definite advantage in finding information over a standard text interfaces

The study shows that users with minimal training can interact with the Bubble World environment to perform information retrieval tasks and do so more effectively than users of a text-based interface. It also shows that the technique of building multiple schemas to browse problems is not immediately intuitive and this component of the interface requires additional familiarity to use effectively. Overall, user feedback on the interface is very encouraging and has help to identify additional improvements in the future.

6.2 Future Research

Visual information retrieval is a relatively new field of study offering plenty of opportunities for further research. While the results of this research are very encouraging, the techniques are still very new and like any new research, could benefit greatly with further work. This section is dedicated to providing some ideas into where such research could be directed. The section is broadly broken down into three distinct areas: the first is application enhancements, the second is further application research, and the last area discusses different applications that may benefit from these techniques.

6.2.1 Application Enhancements

One of the positive outcomes of the user study conducted in this thesis is substantial user testing and evaluation through user feedback. The visual interface was used by ten individuals representing a total of 8.2 hours of actual usage time. Therefore, the feedback provided by these individuals is a valuable source to pinpoint future enhancements to the package.

The highlights of the feedback are as follows. First, there is substantial support for the incorporation of phrase and proximity matching for the queries. This was discussed in chapter 4 of this thesis but was never implemented due to time constraints. Phrase and proximity matching can easily be added into the existing framework by allowing multiple words to describe each orb and specifying distance using the "word1..word2" notation discussed in section 4.3.5.2 of this thesis. Second, hyperlinks from the document summaries to the document contents would be a more intuitive approach to opening documents than using the frequency graphs alone. Third, allowing more than ten documents to be listed in the icon view was a concern of some who used the interface. Although this was purposely done to try and prevent users returning to a single dimension text-based searching pattern, it did add to some frustration for the user. Fourth, a more intuitive link between the document summary and the frequency graphs would aide the navigation process. At the moment, users are forced to count across the graphs to locate the document to open. Although this is not a significant problem it could have been more intuitive. Fifth, some users had difficulty popping bubbles when they really just wanted to pop orbs. Bubbles are highlighted to indicate they have been selected but because of the close proximity of orbs to these bubbles, it is sometimes difficult for the user to select one without the other. Sixth, some users suggested that dragging the lens in the document view rather than have the lens follow the cursor may be more beneficial.

6.2.2 Further Application Research

It was discussed in chapter 5 that one of the limitations of the experimental testing conducted in this research is that it was only able to provide limited feedback on some of the more interesting features of the application, namely the use of ontology's to provide automated user assistance, the use of data mining tools to provide similar assistance, and the use of multiple schemas to enhance the search and browse patterns of the user.

This leaves the door open for some interesting future work. For instance, the next step to evaluating this technique could be to apply it to the front end of a functioning digital library with substantial document collection and user subscription size. This would allow some evaluation that could not be accomplished in the current test environment. For one, users would now use the technique in a real information retrieval situation instead of a simulated one. In this environment, ontology models could be carefully constructed against the domains of the library (and perhaps user modeling) to provide some useful feedback into how well this automated assistance works. At the same time, more advanced data mining techniques could be developed to extract useful patterns in the collection that could then be incorporated into the search and browse pattern of the user. The concept of integrating a data mining technique into the interaction model of the user has been shown in this research, however, the experiment was not designed to provide feedback on the usefulness of this approach. It is postulated in this research that careful use of ontology models and useful data mining techniques will significantly improve the current effectiveness of the visual interface.

6.2.3 Other Applications

The visual technique itself can be applied to much broader applications than digital libraries containing ‘text’ documents. Essentially, any media that can be ranked by a standard information retrieval engine can effectively be visualized by this technique. The following is a brief description of some additional applications that may benefit from the techniques discussed in this thesis.

6.2.3.1 Web Search Engine

Web searching is an obvious application area that could benefit from this technique since there are many similarities between the current tested environment, and the environment found online. Essentially, only the terms differ: documents are now called pages, document numbers are now links and XML is now HTML. Having said that, there are some specific problems that would have to be addressed. First, in the current application, documents are linked hierarchically through schemas and icons in what amounts to a form of directed graph. Online documents do not have this structure since a document may link to other documents that self-reference back in complex cycles. Online documents are essentially linked as an undirected graph. Therefore, how best to view this self-referencing is an issue yet to be solved. Second, how pages are indexed will affect how well the visualization technique works. For instance, it may not always be feasible to index the full text of each page, since this may amount to more resources than one would like to expend. Even if full text is incorporated, care would have to be given to ensure page biasing does not affect the visual technique. Page biasing is where invisible lines of text are used to rate a page more significantly on some key terms than others, to increase the ranking of the document. In either case, careful use of indexing and possible different heuristic views of the data may be required.

6.2.3.2 Email

An interesting application that is perhaps not so obvious is email. After all, email is essentially text media with similar attributes that can easily be parsed and ranked like any other document. The trick is how to specify the root query so as not to exclude any email from view while maintaining a meaningful form so that useful schemas can be built from it. To do this, the root query could be specified from the email fields (such as To, From and Cc) and spatially distributed using a different set of coding schemes (such as date and time of email). Once the root query is specified, pre-defined schemas can be used to sort the email into “pigeon holes” grouped on the context of the email, and not by the email fields alone.

6.3 Summary

The future foretells an inevitable increase in the size and number of information sources available and in the importance of retrieving the right information at the right time. As the military, industry and the general public place greater and greater emphasis on sourcing and gathering this information, so must they decide on how it will be retrieved. Current text-based approaches provide a mechanism for information retrieval today, however, their interaction model is simplistic and unable to provide the cognitive clues that may be required for information retrieval tomorrow.

The goal of this thesis was to bring together previous research concepts, the human cognitive thought process of problem solving, and their inherent interaction with the external representations of these problems, to build techniques that successfully amplify and assist the user at each stage of the information retrieval process. This thesis has been successful in contributing to this goal in a number of important ways. First, this thesis has demonstrated how the visual techniques described in this research can significantly benefit information retrieval. By mapping internal representations of the problem to an external view through simultaneous isomorphic transformations of the query and results, and by providing a consistent interaction model that maps the natural knowledge crystallization process of the human mind, the user is able to cognitively offload to the system and subsequently increase performance. Second, it provides useful information and feedback on constructing an experiment for comparatively examining traditional information retrieval techniques against visual techniques, of which there is little currently available. Third, it provides directions for further research both within this current application and other applications that may benefit from this particular visual technique.

Appendix A. Tutorial Extracts

A.1 Textual Interface Tutorial

Welcome to the user testing phase of Bubble World, a new graphical interface for information retrieval. Today you have been selected to use the textual interface to gain some benchmark statistics for the trial. Before we begin with this trial, it is necessary for me to take you through some of the features that you will use. The interface in front of you is very similar to any other textual search and retrieval interface that you may have previously used before.

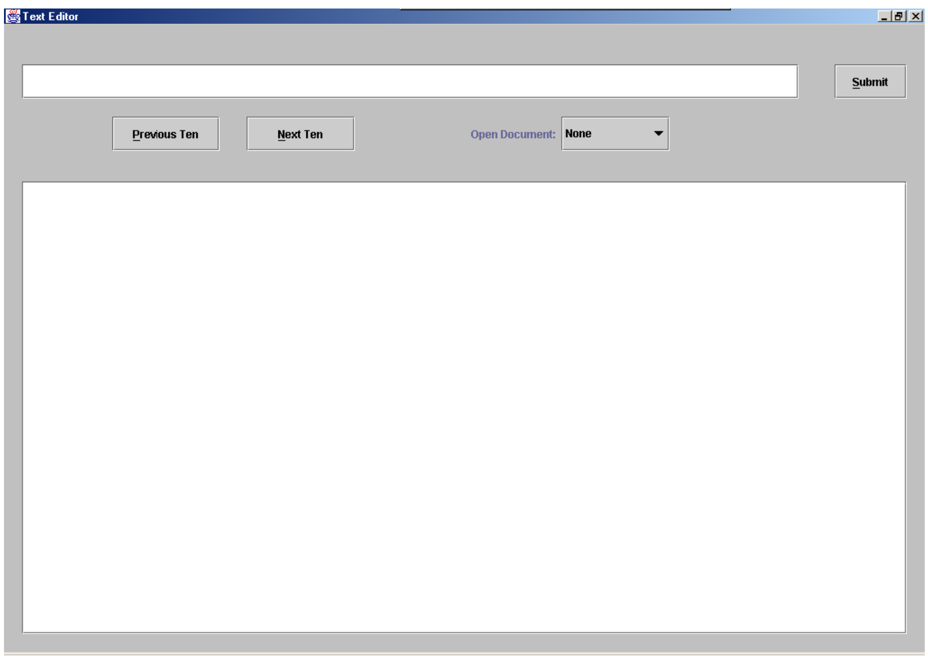


Figure 43. Text Interface

Lets begin by typing in a query on Navy Army and Marine planes. To submit the query, use the submit button found on the right hand side. When the query has been submitted, a list of documents is presented in ranked order from highest to lowest. The ranking is displayed as the green percentage located to the left. The blue text immediately to the right of this, is the document number. The text below is a short summary of the document and the key words that match your query are highlighted in red. The idea here is that you scroll down the list and read the summary of the documents until you find one that you may be interested in. When you locate this document, use the drop down menu to the right and select the document of interest from the list. Note that the order of the documents in this list is the same ordering as the main list so locating the document is simply a matter of locating its position and then verifying the document number matches the one you are about to open.

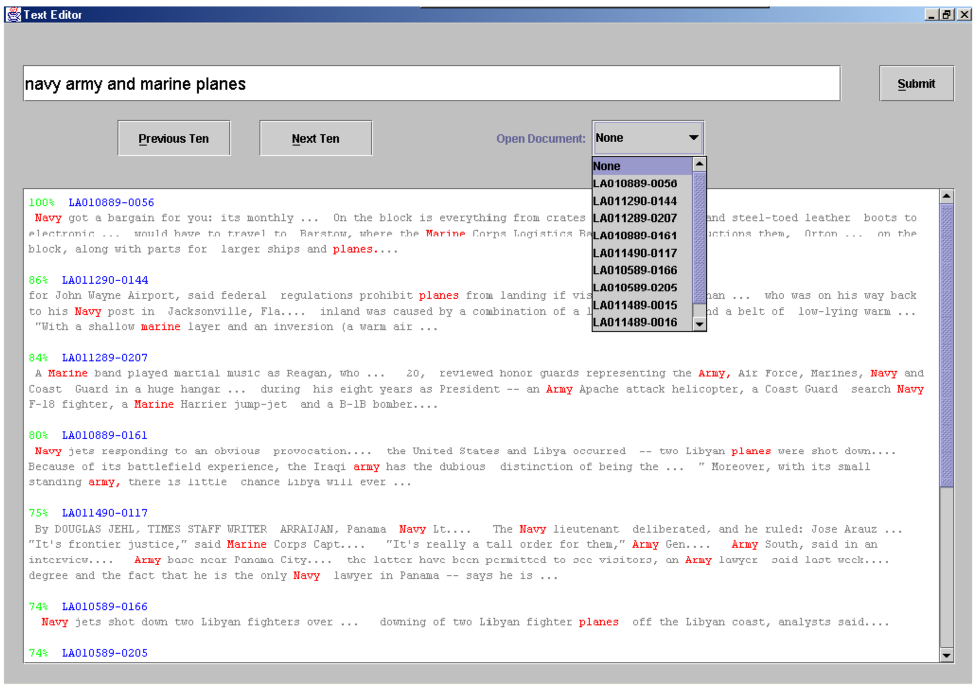
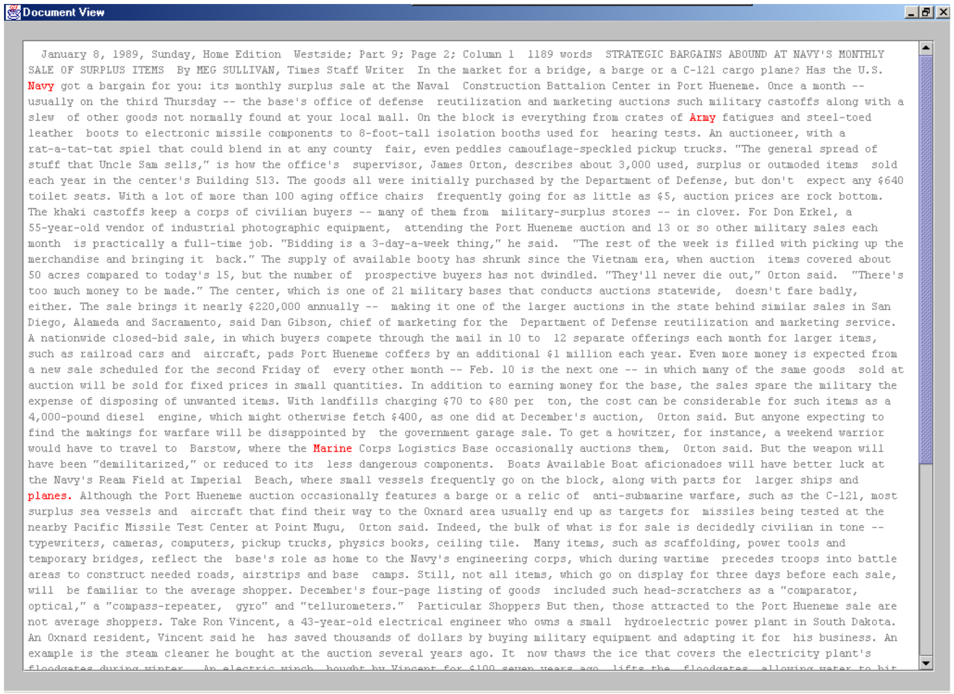


Figure 44. Text interface with Example Query

When you click on the listing, the full document appears. Note once again that the key words matching your query are highlighted. This will help you locate sentences that may be of interest to you quickly and without having to read the whole document. When you are finished simply close the window and continue as before.


Document View

January 8, 1989, Sunday, Home Edition Westside; Part 9; Page 2; Column 1 1189 words STRATEGIC BARGAINS ABOUND AT NAVY'S MONTHLY SALE OF SURPLUS ITEMS By MEG SULLIVAN, Times Staff Writer In the market for a bridge, a barge or a C-121 cargo plane? Has the U.S. Navy got a bargain for you: its monthly surplus sale at the Naval Construction Battalion Center in Port Hueneme. Once a month -- usually on the third Thursday -- the base's office of defense reutilization and marketing auctions such military castoffs along with a slew of other goods not normally found at your local mall. On the block is everything from crates of Army fatigues and steel-toed leather boots to electronic missile components to 8-foot-tall isolation booths used for hearing tests. An auctioneer, with a rat-a-tat-tat spiel that could blend in at any county fair, even peddles camouflage-speckled pickup trucks. "The general spread of stuff that Uncle Sam sells," is how the office's supervisor, James Orton, describes about 3,000 used, surplus or outmoded items sold each year in the center's Building 513. The goods all were initially purchased by the Department of Defense, but don't expect any $640 toilet seats. With a lot of more than 100 aging office chairs frequently going for as little as $5, auction prices are rock bottom. The khaki castoffs keep a corps of civilian buyers -- many of them from military-surplus stores -- in clover. For Don Erkel, a 55-year-old vendor of industrial photographic equipment, attending the Port Hueneme auction and 13 or so other military sales each month is practically a full-time job. "Bidding is a 3-day-a-week thing," he said. "The rest of the week is filled with picking up the merchandise and bringing it back." The supply of available booty has shrunk since the Vietnam era, when auction items covered about 50 acres compared to today's 15, but the number of prospective buyers has not dwindled. "They'll never die out," Orton said. "There's too much money to be made." The center, which is one of 21 military bases that conducts auctions statewide, doesn't fare badly, either. The sale brings it nearly $220,000 annually -- making it one of the larger auctions in the state behind similar sales in San Diego, Alameda and Sacramento, said Dan Gibson, chief of marketing for the Department of Defense reutilization and marketing service. A nationwide closed-bid sale, in which buyers compete through the mail in 10 to 12 separate offerings each month for larger items, such as railroad cars and aircraft, pads Port Hueneme coffers by an additional $1 million each year. Even more money is expected from a new sale scheduled for the second Friday of every other month -- Feb. 10 is the next one -- in which many of the same goods sold at auction will be sold for fixed prices in small quantities. In addition to earning money for the base, the sales spare the military the expense of disposing of unwanted items. With landfills charging $70 to $80 per ton, the cost can be considerable for such items as a 4,000-pound diesel engine, which might otherwise fetch $400, as one did at December's auction, Orton said. But anyone expecting to find the makings for warfare will be disappointed by the government garage sale. To get a howitzer, for instance, a weekend warrior would have to travel to Barstow, where the Marine Corps Logistics Base occasionally auctions them, Orton said. But the weapon will have been "demilitarized," or reduced to its less dangerous components. Boats Available Boat aficionadoes will have better luck at the Navy's Ream Field at Imperial Beach, where small vessels frequently go on the block, along with parts for larger ships and planes. Although the Port Hueneme auction occasionally features a barge or a relic of anti-submarine warfare, such as the C-121, most surplus sea vessels and aircraft that find their way to the Oxnard area usually end up as targets for missiles being tested at the nearby Pacific Missile Test Center at Point Mugu, Orton said. Indeed, the bulk of what is for sale is decidedly civilian in tone -- typewriters, cameras, computers, pickup trucks, physics books, ceiling tile. Many items, such as scaffolding, power tools and temporary bridges, reflect the base's role as home to the Navy's engineering corps, which during wartime precedes troops into battle areas to construct needed roads, airstrips and base camps. Still, not all items, which go on display for three days before each sale, will be familiar to the average shopper. December's four-page listing of goods included such head-scratchers as a "comparator, optical," a "compass-repeater, gyro" and "tellurometers." Particular Shoppers But then, those attracted to the Port Hueneme sale are not average shoppers. Take Ron Vincent, a 43-year-old electrical engineer who owns a small hydroelectric power plant in South Dakota. An Oxnard resident, Vincent said he has saved thousands of dollars by buying military equipment and adapting it for his business. An example is the steam cleaner he bought at the auction several years ago. It now thaws the ice that covers the electricity plant's

Figure 45. Document view of Text Interface

You will notice that only ten documents are displayed at any one time. To view the next ten, click the ‘next ten’ button found above. You can move though all the document by clicking on these two buttons as shown.

You are now ready to start the user trial. At the bottom of the screen you will see a window called ‘Questions’. Click on this once and a new window appears. To begin the trial, type your last name in the box provided and click the button found directly below. From that point onwards, the questions will appear sequentially to you. Please answer each question carefully before preceding to the next. The emphasis here is on accuracy however you will be timed. To answer each question you will be required to open the document that has the information in it, and write the answer and document number in the answer sheet next to you. Please note that not all questions will have an answer. If at any time you feel there is no answer to the question, check the ‘NO ANSWER’ box next to the question and proceed. If you feel you cannot find the answer, check the ‘ANSWER NOT FOUND’ check box and proceed to the next question. If unsure, assume that the answer was not found and check this box.

At the end of the trial, you will be asked to fill out a short questionnaire about your experiences with using this interface. Please make sure your name appears at the top of this questionnaire. When you are ready to begin notify the tester. End of instructions.

A.2 Graphical Interface Tutorial

Welcome to the testing phase of Bubble World, a new graphical interface for information retrieval. Today you have been selected to use the graphical interface to gain some statistics on how well you do compared to that of others using the textual interface.

Before we begin with this trial, it is necessary for me to take you through some of the features that you will use. I will focus on teaching you the basic functions that will be required throughout this trial. More advanced functions will not be needed here. At the end of the video, you will be given an opportunity to ask questions and try this out for you self so don't worry if you don't remember everything.

The interface in front of you is essentially where queries are issued and manipulated. There is a white canvas area where queries and their results will be displayed, and buttons to the right hand side where queries can be manipulated. To create a query, click on the 'bubble button' found as the top button to the right hand side. Type in the query you wish to submit. For this demonstration we will begin with 'fundamentalist religious groups'. When your finished, click on the submit button and you will notice that the mouse pointer turns into a bubble. Using the left mouse button, click on the canvas once and the query will be issued. What you see in front of you now is a bubble with blue nodes. The bubble represents the query, the nodes represent key words that make up the query.

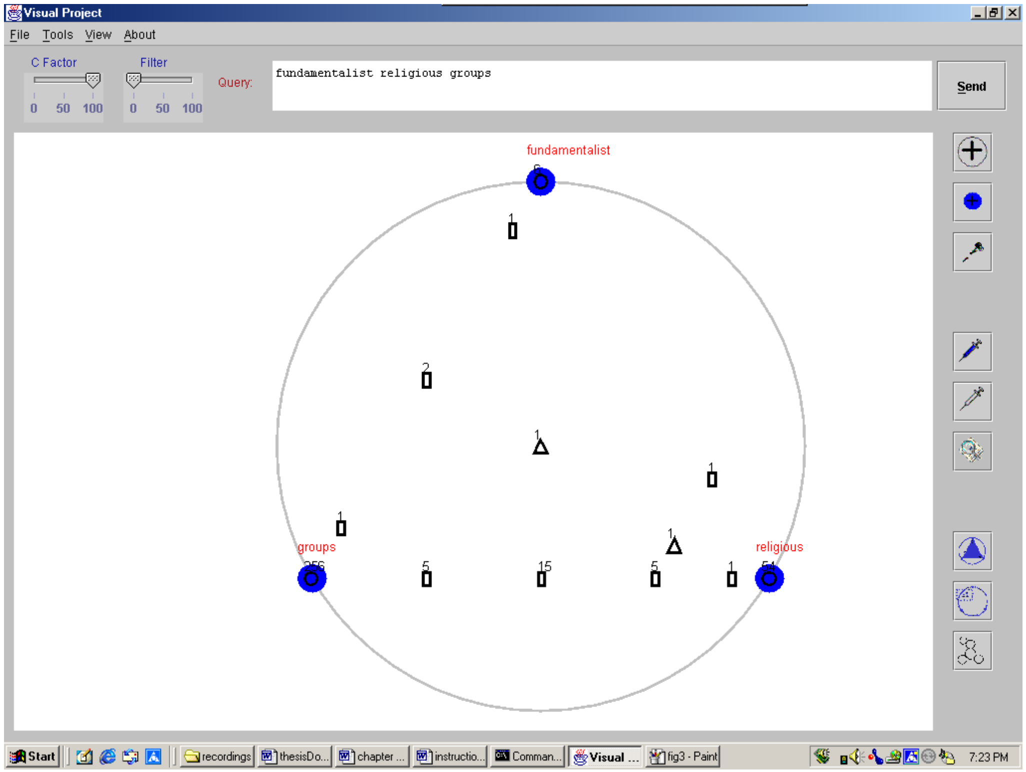


Figure 46. Bubble World Exploratory View

Before we continue, note that you can navigate around the screen using several different functions. First, by dragging the mouse with the left mouse button you can pan left and right, up and down. By dragging the right mouse button up you can zoom into the center of the screen. By dragging the right mouse button down, you can zoom out from the center of the screen. By using the zoom select button on the bottom right, you can select an area to zoom in. This is done by selecting an area from left to right in which to zoom to. At any time, you can select the bird eye view by selecting the bottom right button which will optimize the view to fit the screen. In summary, drag left mouse button to pan left right up and down. Drag right mouse button up to zoom in, down to zoom out. Select an area to zoom to and reset the view at any time.

Now look at the query we issued earlier. You will see a number of icons placed within the circle. These icons represent a collection of documents with certain characteristics. The number above the icon shows the number of documents within the icon, the shape of the icon shows how many key words of the query are found in the documents within the icon (eg rectangle means two words, triangle three words, square four words ect) The position of the icon represents the relationships the documents have with the keywords they contain.

For instance, consider the first triangle. It shows there is only one document inside the icon, the document has all three key words and there is an equal relationship between all these keywords. Consider the second triangle. Again there is only one document inside the icon, the document has all three key words however now the document has more influence on the word ‘religious’ than ‘fundamentalist’ or ‘groups’ and if you look carefully, it is just off center so ‘groups’ is slightly more important than ‘fundamentalist’. Consider the rectangle. It contains 15 documents that have two key words, ‘religious and groups’ but do not have the word ‘fundamentalist’. It is equally spaced between ‘religious’ and ‘groups’ showing an equal relationship between the two words. The icon to the right shows five documents weighted about 2/3 more to ‘religious’ than ‘groups’. The icon to the left shows five documents weighted about 2/3 more to ‘groups’ than ‘religious’.

If we wish to change the query, we can do so by clicking on the blue node button found at the top right of the screen, type the new word to add, say ‘Christian’, then hover over the node that you wish to add it next to. The new node will always be added to the right of this node. To remove any nodes, use the pin button, hover over the node to remove and use the left mouse button to pop this node.

Now lets investigate this query. If we are interested in ‘religious fundamentalist groups’ then the center icon showing an equal weighting of all key words may be of most relevance to us, so we can select this using the left mouse button. What you see now is an intermediate view of all the documents found in the icon. To the right, is the canvas showing the query as it relates to the documents. To the left you see the first ten documents in summary form listed in ranked order from highest to lowest ranking. There is currently only one document, so this is all you see. The blue text indicates the document number, and the red text indicates the words that match the nodes around the bubble. To the top of the screen you see the same ten documents but this time as a graph of the frequency of the key words within the document. For instance, the key words ‘fundamentalist religious groups’ all appear equally in this document. To select the document simply click on this frequency graph and the full text of the document is displayed.

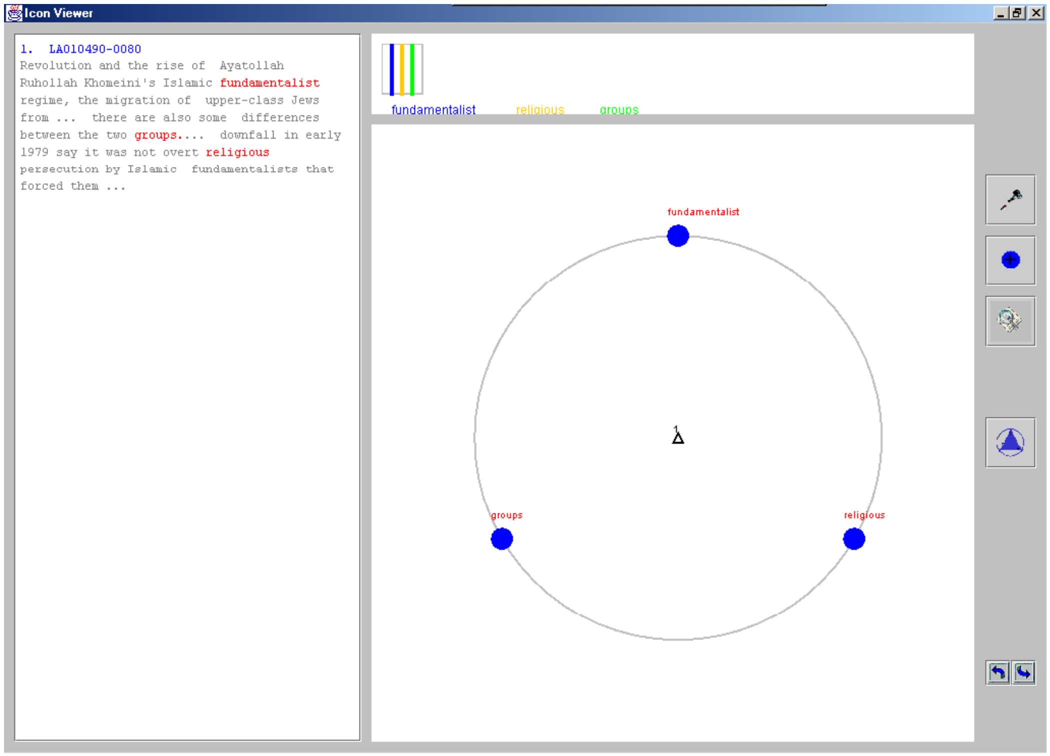


Figure 47. Bubble World Icon View

At the top you will see a rectangular bar indicating the full length of the document. Placed along the rectangular bar are the location of individual key words with the documents. To locate a particular key word within the document, identify the color of the word, then move the curser until the lens hovers over this hit. You will notice that by doing so, a selected text field moves along the full text to indicate the text that is located within this lens. This helps you locate particular key words or multiple key word patterns in the document quickly and easily. Note also that the key words are also highlighted in red, so alternatively you can scan the document for hits in this way.

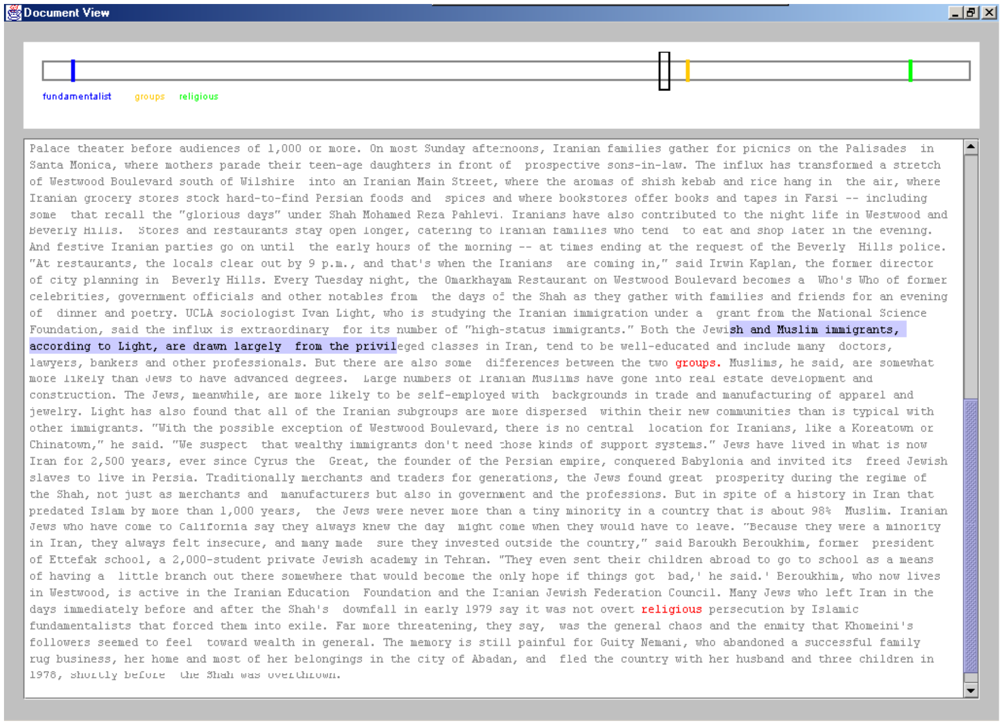


Figure 48. Bubble World Document View

Back to the query at hand. We can close this document when we are done, and then close the icon viewer as well. Notice that the icon just investigated now turns red to indicate that we have investigated the documents within it. Now continuing with the search, we may decide that the other triangle may be the next most relevant so we select it. Again we can read the summary to the left. If we feel this summary is relevant, we can select the document to view, identify the key words in the document, read the surrounding text to see if it is indeed relevant to us and then return to the query when we are done.

If we are still searching for documents then note that only rectangles are left. We must asked the question now, are we more interested in 'fundamentalist religious' 'religious groups' or 'fundamentalist groups'. We may decide that 'fundamentalist groups' is more appropriate so again we can select the icon, investigate the summaries and documents and continue as needed.

We can dispose of this query using the pin button and clicking on the bubble.

Lets consider another query. This time we wish to investigate documents about ‘army marine and navy planes’, so we add the query with the bubble button and use the left mouse button to submit it. Quickly scanning the query we see one square icon indicating it has hits on all four key words so we can zoom in and select it. We can read the summary on the left side and decide it is of not much interest, so we go back to the query. Now we must decide which document we want to investigate from those that only have partial matching. To help us here, we know that the document must at least have the word planes in it so select the node. This immediately tells us that 58 documents have the word ‘planes’ in them and these icons are illuminated green for us. Now we can investigate either ‘army navy planes’ or ‘marine navy planes’ or indeed ‘army planes’, ‘marine planes’ or ‘navy planes’.

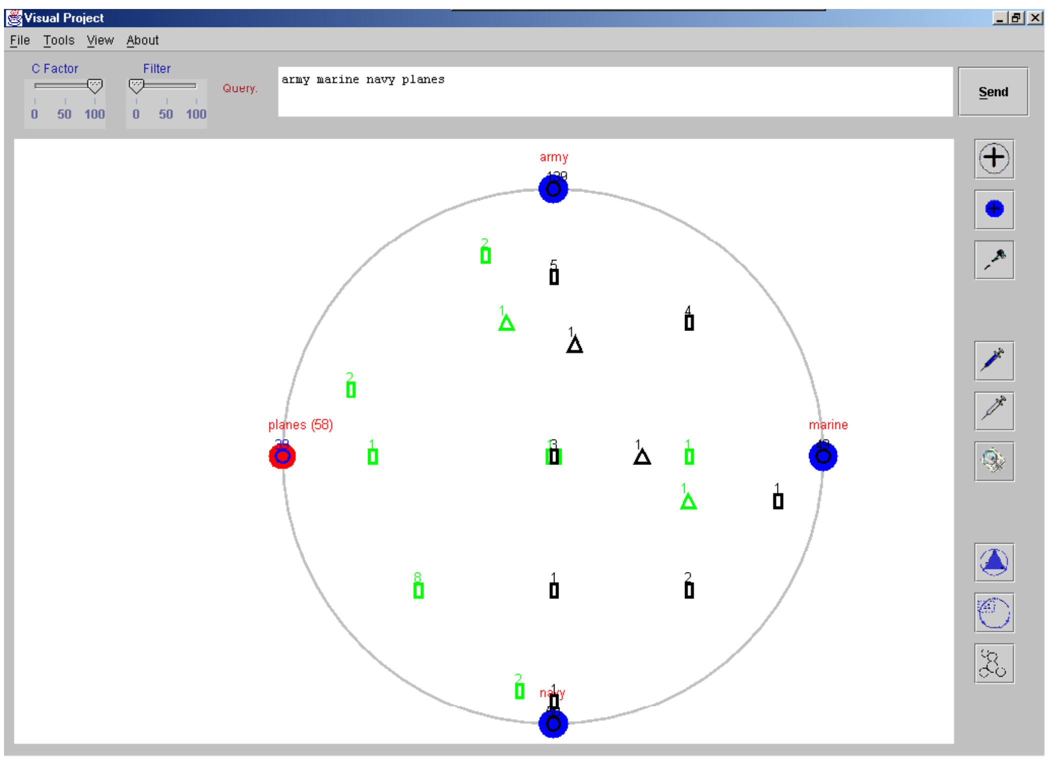


Figure 49. Example of Selecting an Orb

When we are done looking dispose of the query using the pin button as before.

Lets look at the query ‘lady tennis player’. The first thing you will notice is that the word lady has changed blue. This indicates that there is an ontology model behind this word that may provide some assistance in locating the correct documents. To show an example of this, lets look at the triangle icon near player. Now when you look at the summary of the documents you will notice that ‘lady’ did not actually appear in the document but rather ‘woman’ and ‘female’ did. This was the help that the ontology model was able to provide. For this user trial, you need to know nothing more than some assistance is being provided to you.

Consider a more complex question that requires us to answer multiple questions to come up with a final solution. In this next demonstration we will try and find documents about presidents of sporting clubs and then see if they tell us how much they get paid. We begin by issuing the query ‘president of a sporting club’. Now browse the documents that appear to be the most relevant first (that would be the triangles). As we find what type of sporting clubs are out there, we can add another bubble to the keyword ‘president’. We continue to browse and slowly build up this new query as we find the information we are after. On each of these clubs, we can now add the query ‘salary’ as such.

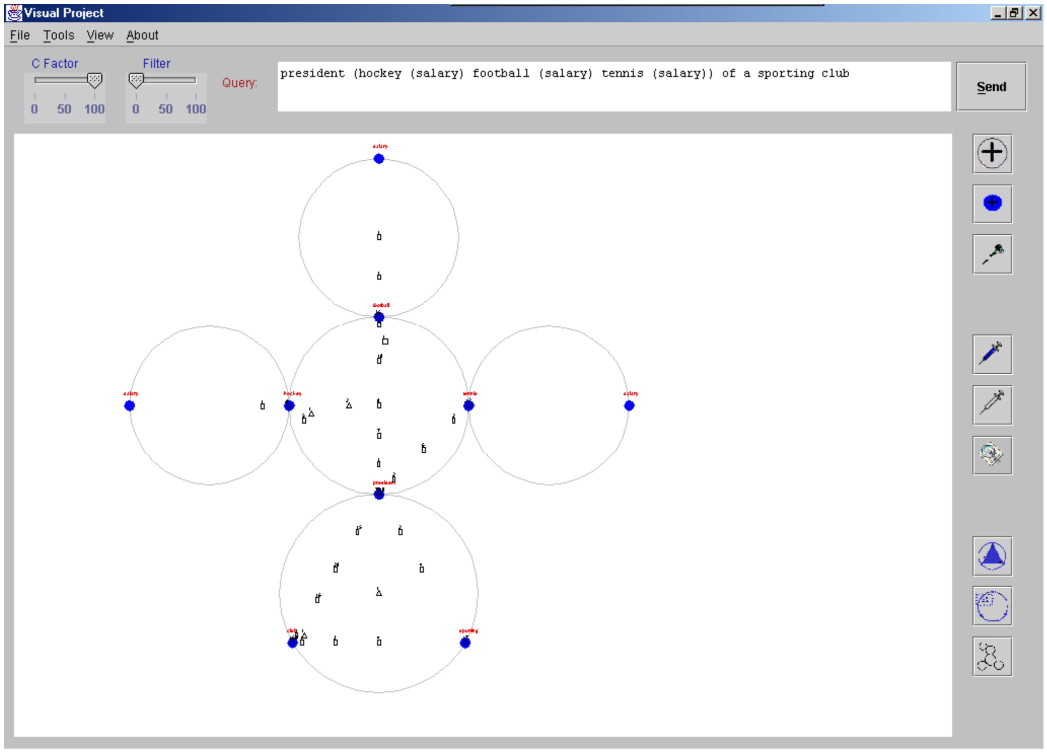


Figure 50. Example of Building Multiple Schemas

Now let me explain what we have done. All the documents from the initial query that have the word 'president' will be allowed to migrate through this node into the next bubble. That means all the documents in this bubble are guaranteed to have the word 'president'. Now all the documents in this circle must have the words 'hockey' and since they come from the circle must also have the word 'president'. Similarly documents in this circle will have the words 'president football' and 'president tennis'. Once the documents filter through each node then their relationships to the nodes within the bubble are the same as before. The only difference is that they must contain the word of a node before they can pass through it. Consider what effect this has had to our documents in the outer bubbles. For one, nearly all the documents are filtered leaving only the documents with 'president hockey' and one with 'president hockey salary'. This gives us a higher chance of having relevant document in the outer circles.

Again, if you didn't catch all of this, don't worry, you'll have a chance to experiment later.

This is about all that you will need to know for the user testing. Some of the other functions can be described to as you use this system for yourself.

You are now ready to start the user trial. At the bottom of the screen you will see a window called 'Questions'. Click on this once and a new window appears. To begin the trial, type your last name in the box provided and click the button found directly below. From that point onwards, the questions will appear sequentially to you. Please answer each question carefully before preceding to the next. The emphasis here is on accuracy however you will be timed. To answer each question you will be required to open the document that has the information in it, and write the answer and document number in the answer sheet next to you. Please note that not all questions will have an answer. If at any time you feel there is no answer to the question, check the 'NO ANSWER' box next to the question and proceed. If you feel you cannot find the answer, check the 'ANSWER NOT FOUND' check box and proceed to the next question. If unsure, assume that the answer was not found and check this box.

At the end of the trial, you will be asked to fill out a short questionnaire about your experiences with using this interface. Please make sure your name appears at the top of this questionnaire. When you are ready to begin notify the tester. End of instructions.

Appendix B. Experiment Documentation

B.1 Experiment Questions

1. Find two documents that talk about the electric motor car.

2. What is the most popular registered boat in Italy and France?

3. A well known deep sea adventurer collaborates with a revolutionary photographer to find a vessel destroyed in WW1. What was the name of the photographer?

4. Find a document that talks about the electric bicycle.

5. There are two documents that have the words 'president kennedy assassination' but actually make no references to President Kennedy's Assassination. List these two documents.

6. Enter the query 'white house press bush' and do not change this. Now browse the documents until you find one that talks about Lady Barbara Bush suffering from persistent double vision and swelling around her eyes.

7. Is it a felony offense to donate blood if infected with the AIDS virus?

8. Who did Charles White play for and what was his position.

9. Find a document about a lady who commanded troops into battle.

10. You have been asked to do an assignment on three united states presidents. To make research easier, you will pick the three that appear to have the most documents mentioning them (and therefore the most research material). To this end, find at least five presidents and determine which three of these presidents appear to be mentioned in the most documents.

11. Are there any documents that mention all three of the presidents that you chose from the previous question? If so, list up to five documents mentioning all three presidents in the same document?

12. To finish off, you feel it would be nice to know the first name of the wife of the presidents that you have chosen. If you are able, list these.

B.2 Testing Answer Sheet

Bubble World: User Testing Answer Sheet

Test Type: T G

Name: ____________________ Date: ____________________

Question 1: ☐ NO ANSWER ☐ ANSWER NOT FOUND

____

____

Question 2: ☐ NO ANSWER ☐ ANSWER NOT FOUND

____

Question 3: ☐ NO ANSWER ☐ ANSWER NOT FOUND

____

Question 4: ☐ NO ANSWER ☐ ANSWER NOT FOUND

____

Question 5: ☐ NO ANSWER ☐ ANSWER NOT FOUND

____

____

Question 6: NO ANSWER ANSWER NOT FOUND

Question 7: ☐ NO ANSWER ☐ ANSWER NOT FOUND

Question 8: ☐ NO ANSWER ☐ ANSWER NOT FOUND

Question 9: ☐ NO ANSWER ☐ ANSWER NOT FOUND

Question 10: ☐ NO ANSWER ☐ ANSWER NOT FOUND

Tick 3 that appear the most

______________________________ ☐

______________________________ ☐

______________________________ ☐

______________________________ ☐

______________________________ ☐

Question 11: ☐ NO ANSWER ☐ ANSWER NOT FOUND

____

____

____

____

____

Question 12: ☐ NO ANSWER ☐ ANSWER NOT FOUND

____

____

____

B.3 User Testing Questionnaire

Bubble World User Testing – Questionnaire

Name: ______________________________ Date: _________________________

(please circle the appropriate response)

Question 1: How well do you feel that you were able to answer the questions using the interface provided?

Not well at all						Very Well
1	2	3	4	5	6	7

Comments: ____

____

Question 2: Do you feel that the question represented reasonable 'real world' problems that one might use a search engine to find?

Not at all						Very appropriate
1	2	3	4	5	6	7

Comments: ____

____

Question 3: What was your overall feeling in the usefulness of the interface you used?

Not impressed						Very impressed
1	2	3	4	5	6	7

Comments: ____

____

Question 4: Would you use this interface again if available as a front end to a digital library or web search engine?

Yes No

Question 5: Do you have any other comments about the interface that you used. If so, please elaborate below.

Comments: ______________________________

Appendix C. Experimental Results

Bubble World User Testing Results													
Textual Interface Results													
	Test Cases												
Question Number	1	2	3	4	5	6	7	8	9	10	SubTot	Totals		
1	359	**370**	**428**	400	**142**	**236**	**353**	223	**248**	**378**	308	324		Type 1
2	**113**	**310**	**1442**	**351**	189	**150**	**188**	136	**151**	**628**	417	391		852
3	**129**	561	**580**	**548**	271	523	330	234	242	1217	419	489		
4	**122**	456	643	215	**162**	**202**	**181**	141	**254**	**252**	196	276		Type 2
5	194	**149**	**491**	235	**363**	448	**444**	203	**258**	260	341	316		716
6	**189**	**327**	**550**	**206**	**179**	**276**	179	**196**	**319**	**116**	270	260		
7	**168**	**113**	**112**	**94**	**156**	**206**	**197**	**78**	**101**	**81**	136	136		Type 3
8	**140**	**170**	**296**	**154**	**211**	**140**	**250**	**155**	**99**	**152**	179	179		217
9	**100**	**397**	**414**	**157**	**167**	**85**	**233**	**63**	**217**	**181**		217		805
10	**243**	**1115**	**436**	**457**	498	**430**	**635**	**427**	406	**302**	517	502		Scenerio2
11	242	**466**	467	**156**	113	**159**	196	44	282	**147**	232	248		750
12			**760**	**339**	**361**	**369**	**535**	**289**	370	**399**				
												Average	Variance	
SubTime(1-8)	1414	2456	4542	2203	1673	2181	2122	1366	1672	3084	3015	2372	905850.9	
Total Time	1999	4434	5859	2973	2451	2855	3186	1900	2577	3714		3122	1405807	
Accuracy	73%	82%	82%	73%	64%	82%	73%	45%	73%	82%		76%	0.004132	
TimexAccuracy						1210			1172	1607		1330		
Number of Queries			18	29	38	33	27		19	59		32		
									67101					
Questionnair														
1	5	6	5	5	6	7	3	6	4	7		5.4		
2	3	6	5	5	5	4	4	4	6	5		4.7		
3	5	7	6	3	5	4	4	7	7	5		5.3		
4	4	3	4	4	4	1	3	3	3	4		3.3		
5	1	0	1	0	1	0	1	0	0	1		0.5		
Graphical Interface Results														
	Test Cases													
Question Number	1	2	3	4	5	6	7	8	9	10				
1	**114**	**632**	**140**	**366**	**173**	**374**	**541**	**229**	192	**310**	281	281		Type 1
2	**202**	563	**140**	**290**	**230**	**424**	**266**	**157**	172	**554**	283	283		731
3	325	807	532	297	**384**	**412**	155	179	441	**361**	386	331		
4	**66**	**99**	**123**	**233**	**355**	**405**	**60**	**55**	**88**	**229**	191	191		Type 2
5	**115**	949	**178**	**241**	**172**	**269**	**206**	**211**	**297**	**106**	187	187		766
6	**498**	**435**	**142**	**284**	**281**	**249**	**534**	**314**	**179**	**259**	320	320		
7	**139**	**140**	**129**	**224**	**176**	**205**	**108**	**172**	**155**	**187**	168	168		Type 3
8	**108**	**941**	**195**	**542**	**81**	**152**	**154**	**385**	461	**423**	255	255		129
9	**71**	**102**	**74**	**145**	**137**	**316**	**56**	**121**	**138**	**115**		129		518
10	**617**	645	**1022**	589	790	**317**	**462**	**1144**	886	**447**	668	674		Scenerio2
11	**377**	294	**202**	427	**622**	**209**	131	195	**168**	**163**	315	291		964.25
12			**683**	**643**	**748**	**209**	**855**	**295**	916	**976**		551		
												Average	Variance	
SubTime(1-8)	1567	4566	1579	2477	1852	2490	2024	1702	1985	2429		2015	160840.6	
Total Time	2632	5607	2877	3638	3401	3332	2673	3162	3177	3154		2979	127302.3	
Accuracy	91%	55%	91%	73%	91%	100%	82%	82%	55%	100%		89%	0.008855	
TimexAccuracy	1127		869	1939	1296	1809	1663	1312		1962		1497		
Number of Queries			55	47	59	38	41	26		55		46		
											3053			
Questionnair														
1	6	7	4	6	5	6	5	6	4	7		5.6		
2	7	5	5	5	5	6	4	6	5	6		5.4		
3	6	7	7	6	5	5	6	7	7	6		6.2		
4	6	7	7	6	5	7	6	7	7	6		6.4		
5	1	1	1	1	1	1	1	1	1	1		1		

Table 7. Spread Sheet of Experimental Results

Bibliography

1 Baeza-Yates, R and B. Ribeiro-Neto “Modern Information Retrieval”, Addison-Wesley, 1999.

2 Ellis, D. “A behavioural model for information retrieval system design”, Journal of Information Science, 15:237-247, 1989.

3 Borgman, C.. “Why are online catalogs still hard to use?”, Journal of the American Society for Information Science, 47(7): 493-503, 1996

4 O’Day, V. and R. Jeffries “Orienteering in an information landscape: how information seekers get from here to there”, In Proc. of the INTERCHI ’93, Amsterdam, Netherlands, IOS Press, April 1993.

5 “Evaluation of Text, Numeric and Graphical Presentations for Information Retrieval Interfaces: User Preference and Task Performance Measures”, University of Pittsburgh.

6 Koshman, S. “VIBE Usability: an Investigation into Visualization Techniques for Information Retrieval”, Dissertation, University of Pittsburgh, 1996.

7 “How Information Retrieval Started”, American Society of Indexers, www.asindexing.org/site/history.shtml

8 Hiemstra, D. “Using language models for information retrieval”, Dissertation, Center for Telematics and Information Technology, Netherlands, 2001.

9 Salton, G. and M.J. McGill “Introduction to Modern Information Retrieval”, McGraw-Hill, 1983.

10 Luhn, H.P. “A statistical approach to mechanised encoding and searching of literary information”, IBM Journal of Research and Development 1 (4), 309–317, 1957.

11 Galton, G. and C. Buckley. “Term-weighting approaches in automatic retrieval”, Information Processing and Management, 24(5):513-523, 1988.

12 Maron, M.E. and J. L. Kuhns “On relevance, probabilistic indexing and information retrieval.”, Journal of the Association for Computing Machinery 7, 216–244. 1960.

13 Robertson, S.E. “The probability ranking principle in IR.” Journal of Documentation 33 (4), 294–304, 1977.

14 Shaw, W.M. Jr., R. Burgin, and P. Howell. “Performance standards and evaluations in IR test collections: Cluster-based retrieval models.” Information Processing & Management, 33(1):1-14, 1997.

15 van Rijsbergen, C.J. “Information Retrieval.” Butterworths, 1979.

16 Ellis, D.. “A behavioural model for information retrieval system design”, Journal of Information Science, 15:237-247, 1989.

17 Borgman, C.L. “Why are online catalogs still hard to use?”, Journal of the American Society for Information Science, 47(7): 493-503, 1996.

18 Russell, D.M., M. Stefik, P. Pirolli and S. Card. "The cost structure of sensemaking", In Proc. of ACM Conference on Human Factors in Computing Systems, pages 269-276, 1993.

19 Card, S., Mackinlay, Shneiderman "Readings in Information Visualization, Using Vision to Think" p10.

20 Hearst, M. "TileBars: Visualization of Term Distribution Information in Full Text Information Access", *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*(CHI), pp. 59-66, Denver, CO, May 1995.

21 Eick, S.G., J. L. Steffen, and E. E. S. Jr. "SeeSoft---A Tool for Visualizing Line Oriented Software Statistics", IEEE Transactions on Software Engineering, 18(11):957--968, November 1992.

22 Keim, D. and H. Kriegel "VisDB: Database Exploration using Multidimensional Visualization", University of Munich, www.dbs.informatik.uni-muenchen.de/dbs/projekt/visdb/visdb.html

23 Korfhage, R.R. "To see or not to see – is that the query?", In Proc. Of the 14th Annual Int. ACM SIGIR Conference, pages 134-141, Chicago, USA, 1991.

24 Hemmje, M., C. Kunkel, and A. Willett. "LyberWorld – a visualization user interface supporting fulltext retrieval", In Proc. Of the 17th Annual International ACM SIGIR Conference, pages 249-259, Dublin, Ireland, July 1994.

25 Spoerri, A. "InfoCrystal: A visual tool for information retrieval & management", In Proc. of Information Knowledge and Management '93, pages 11-20, Washington, DC, Nov 1993.

26 "WebVIBE", University of Pittsburgh, http://www2.sis.pitt.edu/~webvibe/WebVibe/

27 Hearst, M.A. and J.O. Pedersen "Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results.", In Frei, H.P. et al., (Eds.) Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '96) (pp 76-84), 1996.

28 Hearst, M. "TileBars: Visualization of Term Distribution Information in Full Text Information Access" Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems(CHI), pp. 59-66, Denver, CO, May 1995.

29 Veerasamy, A. and R. Heikes, "Effectiveness of a graphical display of retrieval results.", In Belkin, N.J. et al., (Eds.) Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '97) (pp 236-245). New York : ACM, 1997.

30 Leouski, A. and J. Allan, " Visual Interactions with a Multidimensional Ranked List." In W.B. Croft, et al. (Eds.), Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (pp 353-354), 1998.

31 Hendley, R.J. et. al., "Narcissus: Visualizing Information," in Proceedings Of Information Visualization Symposium 1995, N. Gershon and S.G.Eick, ed., IEEE CS Press, Los Alamitos, Calif., 1995, pp. 90-96. 1995.

32 Hearst, M.A. and C. Karadi, "Cat-A-Cone: An Interactive Interface for Specifying Searches and Viewing Retrieval Results using a Large Category Hierarchy." In Belkin, N.J. et al., (Eds.) Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '97) (pp 236-245). New York : ACM, 1997.

33 Nowell, L.T. "Visualizing Search Results: Some Alternatives to Query-Document Similarity.", In Frei, H.P. et al., (Eds.) Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '96) (pp 65-75). New York : ACM, 1996.

34 Leuski, A. and J. Allan. "Evaluating a visual navigation system for a digital library." International Journal on Digital Libraries, 2000.

35 Scaife, M. and Y. Rogers "External Cognition: how do Graphical Representations work?", School of Cognitive and Computing Sciences University of Sussex , Brighton BN1 9QH, U.K.

36 Molitor, S., S. Ballstaedt, and H. Mandl, "Problems in Knowledge acquisition from text and pictures." In: H. Mandl & J.R. Levin, Eds. *Knowledge Acquisition from Text and Pictures.* Amsterdam: North-Holland/Elsevier Science Publishers. pp. 3-30, 1989.

37 Winn, W. "An account of how readers search for information in diagrams.", *Contemporary Educational Psychology,* **1**8, (2), 162-185, 1993.

38 Swan, R., J. Allan, and D. Byrd "Evaluating a Visual Information Retrieval Interface: AspInquery", at TREC-6 Center for Intelligent Information Retrieval Computer Science Department University of Massachusetts Amherst, MA 01003.

39 Norman, D. "The Psychology of Everyday Things". New York: Basic Books, 1988.

40 Norman, D. "Cognition in the head and in the world." *Cognitive Science,* **17** (1), 1-6, 1993.

41 Vera, A.H. and H.A. Simon, "Situated Action: A Symbolic Interpretation.", *Cognitive Science,* **17** (1), 7-48, 1993.

42 Zhang, J. and D.A. Norman "Representations in distributed cognitive tasks." *Cognitive Science,* **1**8, 87-122, 1994.

43 Stenning, K. and R. Tobin "Assigning information to modalities: comparing graphical treatments of the syllogism.", In: E. Ejerhed & S. Lindstrom, Eds. *Action, Language and Cognition.* Proceedings of International Conference on Dynamic Semantics, Umea: Sweden. September 1993. Kluwer. (in press), 1995.

44 Stenning, K. and J. Oberlander "A cognitive theory of graphical and linguistic reasoning: logic and implementation.", *Cognitive Scienc*e, **1**9, 97-140, 1995.

45 Sugiyama, K. and K. Misue "Visualization of structural information: Automatic drawing of compound digraphs.", *IEEE Transactions on Systems, Man, and Cybernetic*s, **2**1, (4), 876-892, 1991.

46 Jansen, B.J., A. Spink, J. Bateman, and T. Saracevic "Real life information retrieval: A study of user queries on the web." Sigir Forum, 32:1, 5-17, 1998.

47 Jansen, B.J., A. Spink, J. Bateman, and T. Saracevic "Failure analysis in query construction: data and analysis from a large sample of web queries.", In: Proceeding of Digital Libraries '98, Pittsburgh, PA, USA, June 24-27, 1998. ACM Press, New York, pp289-290, 1998.

48 Spink, A., J. Bateman, and B.J. Jansen "Searching heterogeneous collections on the web: behaviour of Excite users.", Information Research on Electronic Journal, 4,:2, 1998.

49 Cunningham, J.S.S., R. McNab, and S. Boddie "A transaction log analysis of a digital library.", International Journal on Digital Libraries, v 3, no 2, p 152-169, 2000.

50 Croft, W.B., R. Cook, and D. Wilder "Providing government information on the internet: Experience with THOMAS.", In Proceedings of Digital Libraries. '95, pages 19-24, Austin, TX, June 1995.

51 La, X.A. and R.B. Keefer "Query expansion/reduction and its impact on retrieval effectiveness.", In Donna Harman, editor, Proceedings of the Third Text Retrieval Conference TREC-3 pages 231-239. National Institute of Standards and Technology Special Publications 500-225, 1995.

52 Harman, D. (editor) Proceedings of the Third Text Retrieval Conference TREC-3 National Institute of Standards and Technology Special Publication 500-225, 1995.

53 Salton, G. and C. Buckley "Improving retrieval performance by relevance feedback." JASIS, 41(4): 288-297, 1990.

54 Jing, Y. and W.B. Croft "An association thesaurus for information retrieval." In Proceedings of RIAO, pages 146-160, Rockefeller University, New York, 1994.

55 Hearst, M. "Improving Full-Text Precision on Short Queries using Simple Constraints", Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval (SDAIR), Las Vegas, NV, April 1996.

56 Lenat, D and R. Guha "Building Large Knowledge Bases", Addison-Wesley, 1990.

57 Gaines, B. (editor) "Using Explicit Ontologies in Knowledge-based System", Development: a special issue, International Journal of Human-Computer Studies, vol 46, no 2/3 1997.

58 Furnas, G., I. Landauer, L. Gomez, and S. Dumais "The Vocabulary Problem in Human-System Communication", Communication of the ACM, 30 (11):964-971, 1987.

59 TREC, Text Retrieval Conference Home page, www.trec.nist.gov/

60 Newby, G.B. "An investigation of the role of navigation for information retrieval.", Proceedings of ASIS '92, 20-25, 1992.

61 Spoerri, A. "InfoCrystal: A visual tool for information retrieval." Visualization '93 Proceedings (pp. 150-157). Los Alamitos: IEEE Computer Society Press, 1993.

62 Spoerri, A. "InfoCrystal: a visual tool for information retrieval." Unpublished doctoral dissertation, Massachusetts Institute of Technology, MIT, 1995.

63 Koshman, S.L. “User Testing of a Prototype Visualization-Based Information Retrieval System”, Dissertation, University of Pittsburgh, 1996.

64 Koshman, S.L. “VIBE User Study Technical Report”, Technical Report #LS062/IS97001, University of Pittsburgh, 1997.

65 Kim, H. and R.R. Korfhage, “BIRD, a browsing interface for the retrieval of documents.”, In Proceedings of the 1994 IEEE Symposium on Visual Languages. Los Alamitos: IEEE Computer Society Press, 1994.

66 Kim, H. “BIRD: Browsing Interface for Retrieving Documents”, Dissertation, University of Pittsburgh, 1997.

67 Mendenhall, Wackerly and Scheaffer “Mathematical Statistics with Applications”, Fourth Edition, Duxbury Press, USA, 1994.

Vita

Christopher Laurent Van Berendonck was born in Denain, France in July 1971, and moved to Australia at the age of 3. He graduated from Bowen State High School, Bowen Australia in December 1987 and enlisted in the Royal Australian Air Force the following month. He earned an Associate Diploma in Electronic Engineering from Footscray College of TAFE and a Radio Technician (Ground) course from the Air Force RADS in December 1990. Posted to Darwin, Northern Territory as a Communication Technician he began part time study towards a Bachelors degree at the University of Northern Territory. He was commissioned in 1993, and attended the University of Northern Territory full time until graduating in December 1996 with a Bachelor of Engineering (Electrical/Electronics) and a Bachelor of Science (Mathematics and Theoretical Physics). Posted to the Aircraft Research and Development Unit Adelaide Australia in January 1997, he worked as a design engineer and release authority for prototype designs to many RAAF platforms.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
09-03-2002	**Master's Thesis**	Jun 2001 – Mar 2002

4. TITLE AND SUBTITLE

BUBBLE WORLD – A NOVEL VISUAL INFORMATION RETRIEVAL TECHNIQUE

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Christopher L. Van Berendonck, Flight Lieutenant, RAAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 P Street, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GCS/ENG/02M-09

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Capt David Marsh
Air Force Research Laboratories, Information Directorate
Information Technology Division
(315) 330-2885 david.marsh@rl.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/IFTB

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

With the tremendous growth of published electronic information sources in the last decade and the unprecedented reliance on this information to succeed in day-to-day operations, comes the expectation of finding the right information at the right time. Sentential interfaces are currently the only viable solution for searching through large infospheres of unstructured information, however, the simplistic nature of their interaction model and lack of cognitive amplification they can provide severely limit the performance of the interface. Visual information retrieval systems are emerging as possible candidate replacements for the more traditional interfaces, but many lack the cognitive framework to support the knowledge crystallization process found to be essential in information retrieval. This work introduces a novel visual information retrieval technique crafted from two distinct design genres: (1) the cognitive strategies of the human mind to solve problems and (2) observed interaction patterns with existing information retrieval systems. Based on the cognitive and interaction framework developed in this research, a functional prototype information retrieval system, called Bubble World, has been created to demonstrate that significant performance gains can be achieved using this technique when compared to more traditional text-based interfaces. Bubble World does this by successfully transforming the internal mental representation of the information retrieval problem to an efficient external view, and then through visual cues, provides cognitive amplification at key stages of the information retrieval process. Additionally, Bubble World provides the interaction model and the mechanisms to incorporate complex search schemas into the retrieval process either manually or automatically through the use of predefined ontological models.

15. SUBJECT TERMS

Visual Information Retrieval, Cognitive Framework, Interaction Framework, Search Schemas, Cognitive Amplification, Ontology Models

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					LtCol Timothy M Jacobs
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU	174	(937) 255-6565, ext 4279; e-mail: timothy.jocobs@afit.edu