UNIVERSITY OF SOUTHAMPTON
Faculty of Engineering and Applied Science
Department of Electronics and Computer Science

A mini-thesis submitted for transfer from MPhil to PhD

Supervisor: m.c. schraefel, Wendy Hall
Examiner: Nigel Shadbolt

A Transfer Report on the Development of a Framework to Evaluate Search Interfaces for their Support of Different User Types and Search Tactics

by Max L. Wilson

March 6, 2008

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

A mini-thesis submitted for transfer from MPhil to PhD

by Max L. Wilson

As the understanding of search systems, user needs and seeking strategies is developing, the design of search user interfaces is evolving to support more complicated and exploratory forms of search. With the design of new search features that enable these richer modes of exploration, comes the need to better understand the support they provide. In this report a new evaluation framework is presented that analyses search features for how they a) contribute to an overall interface, b) allow users to carry out different search tactics, and c) support different types of users and their needs. The novel contributions of the framework improve on some of the limitations of typical user studies, and allow search systems to be systematically analysed in much more detail and in much less time. The presented evaluation framework is then validated in three ways. First the validity of the models used as the building blocks of the framework are investigated through related work. Second the method of integrating these building-block models is validated and strengthened by consensus of expert opinion. Third, the overall approach is validated by comparing its analyses to the results of previously carried out user studies. The validation process has shown both the value of the framework and identified areas of future work that should be addressed for the framework to be completed. This report concludes with the set of contributions that the framework makes, and why the remaining work will be challenging, but critical to the final design.

Contents

List of Figures

List of Tables

Acknowledgements

Thanks to Ryen White, Gary Marchionini, Robert Capra, and Nick Belkin as experts in search who have advised on aspects of the work or provided resources to expand it.

Chapter 1

Introduction

1.1 Finding Information in Brief

A common notion for finding information is to perform a keyword search on the web using search engines like Google. Aside from the development of the World Wide Web, such search engines have been the product of decades of research that started in library and information sciences with the introduction of computing technology. In particular, keyword searching has been the designed to create a very simple interaction, where the user enters a few terms and the information retrieval system returns a list of the nearest matching documents. The most efficient and accurate approach to match terms to documents has received much focus, but the interaction of keyword search, however, makes some assumptions about users and the context of their search. First, keyword search assumes that users know which terms should be used to return the documents that they are trying to find and second, that to some extent the user knows that a suitable document exists. Further, keyword search also assumes that users already have a clear idea of what they want or what problem they are trying to solve. Consequently, another stream of research has focussed on both keyword search and alternative forms of searching, such as browsing and exploring, which is called Information Seeking. Information Seeking, however, is a much broader field than specifically keyword search, and is thus less-well defined and less-well understood. This report continues research into information seeking and aims to analyse search systems for their support different types of users provided by many different forms of interaction, including the well known keyword search technique.

1.2 Motivation of the Research

Finding information and building upon ones knowledge is an important activity. As well as the advantages that improved knowledge can provide in business, the activity of searching for information and learning is core to the way a child develops into an adult and throughout life. Further, with technological advances such as the web and even mobile access to it, the activities and processes of searching and learning are becoming increasingly pervasive to our lives. Given the importance placed on finding information, research into improving technologies to do so has been on many fronts, such as better data storage and more efficient algorithms. The accessibility of these developments to individuals, however, is somewhat limited and constrained by the search interfaces that provide the bridge between users and technology. Consequently, research into the design of search interfaces is key and the work presented in this report focusses on a method that can be used to support the design of search systems, so that they in turn can better support people in searching and learning.

Let us consider Google as a popular and well-known example of an interface for finding information on the web. Although Google's early prominence was based upon the significantly improved algorithms for searching the web, the more recent changes to Google have focussed on providing new ways for the algorithms to help users find information, such as searching for images, publications, and shopping products online. Outside of the Google example, many researchers have been considering ways in which search interfaces can be improved, from simple changes to the look and feel of the interfaces to investigating whole new methods of interacting with them. Examples of the latter include research into Exploratory Search (White et al., 2006), which has focussed on interface developments that allow people to freely explore and browse information, as opposed to simply searching for it with keywords. Much of my own work, for example, has been grounded in improving the interaction of an interface called mSpace (schraefel et al., 2006), which has been designed to support users in many forms of search.

Along with the research into new ways of enabling people to search and build knowledge comes the desire to clearly understand the improvements and how they help users. This desire, in the context of helping people to find information and build knowledge, provides the main motivation for the work. To enable users in their search, researchers and designers require tools that help them to better understand the design of search interfaces so that they can be improved for the benefit of their users. The thesis of this research, more specifically, is that to optimally evaluate and understand search interfaces, as opposed to general software design, evaluation methods require both the experience of evaluation from Human Computer Interaction (HCI) and detailed understanding of search from Information Seeking research. This research gap between the contributions of Information Seeking and HCI, has been accurately described by experts in the field of Information Seeking, Järvelin and Ingwersen (2004), who have noted the need for HCI involvement in their research:

> *. . . much research is needed that explores information seeking in various task and actor contexts. However, it is not sufficient to analyse just the in-between*

> *aspects of seeking activities. The need to be related to the task dimension.*

They go on to say:

> *There is a shortage of studies that relate system features to features of task and/or seeking processes.*

The framework produced by this research, and described below, begins to resolve this research gap and assesses the support provided by the features of search interfaces for different types of users and the information seeking tactics they might use. The research described in this report specifically leverages contributions from Information Seeking research to develop a framework that can be used by HCI researchers, or search system developers, to evaluate and improve their designs in a quick, cheap and repeatable way ahead of expensive, but important, user studies. User studies are important, and the framework should not replace them. Instead, therefore, the framework can be used to strengthen early designs and complement and inform the design of user studies.

1.3 Overview of the Report

The rest of the report is structured as follows:

- Chapter 2 presents related work on four fronts. First, related work on Information Seeking behaviour is presented to provide both context to the remainder of the report and specific models that are used later in the framework. Second, related work on the developments of search interfaces is presented, including some interfaces that are later assessed by the framework. Third, as the framework being produced is to be used to help design software, related work into the design and evaluation process is discussed. Finally, work relating to the development of a framework is considered.

- Chapter 3 presents the evaluation framework, which has been the main product of this research and is designed to assess search interfaces for their support for different types of users and the different tactics they may wish to employ. The framework is built using two of the models described in the previous chapter. The framework has been designed to use these models in a novel way to quickly, cheaply, and in a repeatable manner, evaluate the designs of complicated search interfaces, while still revealing detailed analyses of their individual strengths. Given these strengths of the framework, it has been designed for use in the early stages of the design process to strengthen designs and to complement and then inform the structure of user studies.

- Chapter 4 presents the process of validation, which has been applied to the evaluation framework to both strengthen its design and show that it can be used to accurately predict the strengths and weaknesses of interface designs. This validation process, and overcoming the challenges of carrying it out, have constituted the main body of work since the framework was initially designed and presented in the Progress Report (Wilson, 2007). The result of the validation process has shown that the framework can accurately analyse designs to reveal their strengths and weaknesses in ways that user studies may struggle to achieve over a larger period of time or be unable to achieve at all.
- Chapter 5 discusses the initial investigation into possible extensions of the framework that are designed to improve upon the potential limitations of the framework that have been identified through the validation process. The plan for extending and subsequently completing the evaluation framework is described in detail, including the expected methods to be used and challenges that may be faced during the final period of research.
- Finally, Chapter 6 concludes with a summary of achievements thus far, along with the expected changes and contributions of completing the research.

There are also three Appendices to the report:

- Appendix A contains a Gantt chart, and description, showing how the work packages described in Chatper 5 will be completed in the final year.
- Appendix B contains a description of my recent and planned community participation, such as reviewing for and attending events.
- Appendix C contains a list of 8 of my publications that are related to, or referenced by, the work described in the report. The actual papers, excluding the large monograph, are included after the bibliography.

1.4 Research Since the Previous Progress Report

This section is present to briefly make clear what in the report below is new research since the progress report (Wilson, 2007). The most notable development since the progress report is detailed in Section 4 and concerns the Validation of the Framework. The Progress report originally proposed the framework, and the work since has been to design and perform methods to validate and strengthen the framework. As described in the body of Section 4, the validation process was both extremely important and valuable and the results have, as hoped, improved the quality of the framework.

Further to the more defined progress seen in the validation, the period has provided opportunity to apply the framework to interfaces and develop on both the related work and the way the framework is presented. Notably, the more structured approach to related work and defining different fields and sub-fields of research, presented in Section 2 compared to the progress report, has supported the deeper understanding of the framework and its contributions. As a consequence, the improvements in the framework and the way it is described are included in Section 3. Finally, the continued work on the framework, and a developed understanding of its limitations, has provided the means to clearly define remaining work required to complete the framework, which is described in the planned extensions in Section 5.

In summary, the body of the Progress report has been strengthened and better described in Section 2 and Section 3. Further to the earlier report, the evaluation framework has been validated (Section 4) and the final requirements to complete the framework have been planned (Section 5).

Chapter 2

Related Work on Search and the Development of Information Systems

In this chapter, related work is discussed on four fronts. First, and in most detail, related theory, and subsequent studies, on the nature of search is described to provide context to the discussions later in the report. The nature of the research described in the report focusses on understanding how such related work can be used in a novel way to improve other practices such as interface design. Second, related work on the developments of search interfaces is presented, including some interfaces that are later assessed by the framework. Third, as the framework being produced is to be used to help design software, related work into the design and evaluation process is discussed. Finally, work relating to the development of a framework is considered.

2.1 Information Behaviour

Information is a pervasive to our lives. As we develop as a child, and throughout our lives, new information helps us to understand the world and how we can interact with it (Piaget, 1962). The simple notion that new information causes a transition between a former and new state of knowledge was formalised by Brookes (1980). As visualised well by Wilson (1999) in Figure 2.1, however, searching for information is only a part of general information behaviour.

In more recent work, Godbold (2006) summarises research into general information behaviour with Figure 2.2. Here she continues the concept of information causing knowledge changes (Brookes, 1980) by showing what might information behaviour could exist between the former and new states. Notably, once a knowledge gap (Dervin, 1992) has

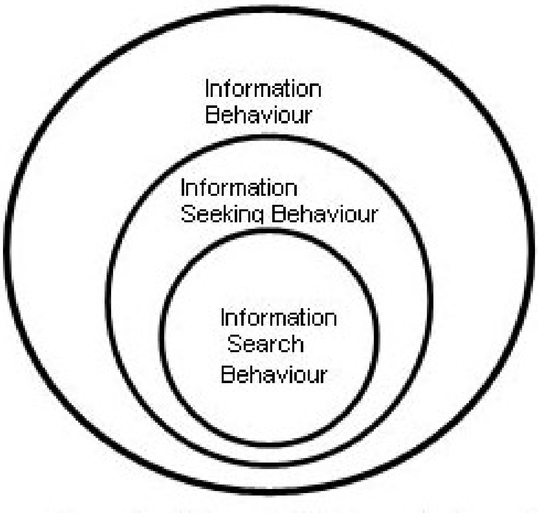

Figure 2.1: The onion model of information behaviour presented by Wilson (1999).

been identified, there are three things a user could do: bridge the gap, close the gap, or ignore the gap. From the model, we can see that information seeking, and within it information search, is only one of the actions a user might take. Information systems, however, are widespread and the need to bridge a gap by finding information provides a very great demand. This demand is what drives the research into understanding search and is only emphasised by the years of research into information science, library science and information retrieval research that has followed.

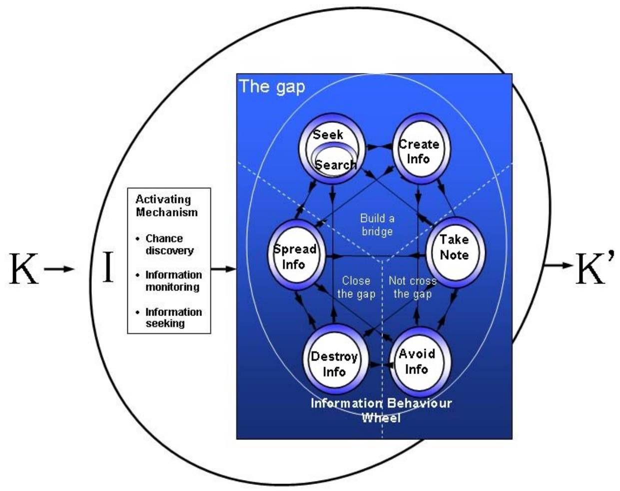

Figure 2.2: The wheel model of information behaviour presented by Godbold (2006).

2.2 Searching for Information

Driven by the desire to find information, the research into search has focussed on many different areas. Information Retrieval is a well known example that has focussed mainly on the effective indexing of documents and efficient matching of terms to indexes during search. This research was broadened slightly to consider other interactive ways of allowing users to enter terms for a keyword search. This view, however, is still focussed on a term-index-document setting, but has been broadened further by information seeking research to look more at users, tasks, goals and strategies for finding information. As part of this extended view of search, a recent investigation into what has been known as Exploratory Search (White et al., 2006), has focussed particularly on the alternatives to keyword searching based on what has been learned from Information Seeking research. Each of these are described in more detail below.

2.2.1 Information Retrieval

Information retrieval research has focussed on a very simple metaphor of search: tell me what you are looking for and I'll do my best to find it. This focus has, therefore, mainly investigated the keyword search interaction that has been made well known on the web by search engines like Google. Users are expected to enter terms that describe what they are looking for, and the search system does its best to find the most relevant documents to give to the user. Not surprisingly research in information retrieval has focussed on how best to decide which are the most relevant documents to given terms and has worked on the problem in three main fronts: matching algorithms, measures for success, and environments for testing.

2.2.1.1 Matching Algorithms

Many matching algorithms have been created, tried and tested. Early work suggested that the number of times that a search term appeared in a document, named simply *term-frequency* was a good indicator for a strong match. This was soon extended to say that *term-frequency, inverse-document-frequency* (or *tf*idf*) improved on this by suggesting that the number of documents that terms occurred in was also important, such that the best match is when a term appears many times, but in a only a few documents (Jones et al., 1972). This immediately removes less useful terms such as 'the' or 'and', which would occur in many documents many times. Such words, often referred to as stopwords, are usually ignored in search algorithms.

Other advances showed that word stemming was important, and that finding the route of each word, by removing optional suffixes for example, meant that a user could enter words like 'absorb', but match documents that use words such as absorbing, or even absorption. *tf*idf* provided a very robust algorithm for matching terms to documents, and adjustments such as word stemming provided only small optimisations to its overall performance. Further work, however, investigated the use of weightings that could be used to enhance *tf*idf*. Subsequent weighting algorithms included RSJ (Robertson et al., 1976) and BM25 (Robertson et al., 1998), which is one of the most widely-used approaches now.

2.2.1.2 Measures for Success

Over time, two specific measures of success for information retrieval have been considered: precision and recall. Raghavan et al. (1989) discusses the two and carefully states the desire that is the driver for any information retrieval system is to *'Retrieve as many relevant items as possible and as few non-relevant items as possible in response to a request'*. The first half of this statement is considered as Recall, where an algorithm aims to get as many relevant documents as possible. The second half is considered as Precision, where the system accurately determines what is and is not relevant. Typically improving one constrains the other, as finding as many documents as possible often involves including ones that are not relevant and making sure that only relevant items are included reduces the number of documents retrieved. Another predictable measure is speed. Early publications about Google showed promising speeds for millions of documents (Brin and Page, 1998), and their aims to improve speed has resulted in most current queries being answered in less than half a second.

2.2.1.3 Testing Environments

Not surprisingly, the need to test any new ideas for algorithms has led to the development of communally available test sets. These test sets have been regularly involved in the TREC Conferences (Harman, 1997) where researchers compete to provide the best results over different datasets. These test environments have allowed research to thrive, but has required the contributions of many researchers to produce both indexed document collections and human assessments of relevance for testing benchmarks.

2.2.2 Interactive Information Retrieval

The research into Information Retrieval has certainly produced efficient and important research into search, but other research has also looked outside of the box to consider what it is like for users to use keyword search and the requirements they have for results, once the speed, precision and recall have been optimised. One observation was that providing a more interactive dialogue with an information retrieval system had specific benefits for retrieval effectiveness (Koenemann and Belkin, 1996). In their paper they investigate Relevance Feedback (Salton and Buckley, 1990), which aids query reformulation by making suggestions to the user. This allows a more interactive dialogue with a search system, by extending the original request-result pattern to include many cycles of suggestions and optional acceptance of them. Koenemann and Belkin (1996) were one of the first to show that there was empirical evidence that improving interaction provided benefits for the users.

Other interactive developments have also been proposed, such as query expansion (Robertson, 1991). Query expansion simply suggests additional terms from documents that closely match a keyword query, to the user as a potentially more specific query. We can see such practices involved in online search engines such as Google, shown in Figure 2.3. The increasing popularity of more interactive forms of information retrieval has led to the development of evaluation frameworks for new interactive designs (Su, 1992; Borlund, 2003). This desire is similar to the motivation that produced the TREC conferences, which also began an interactive track.

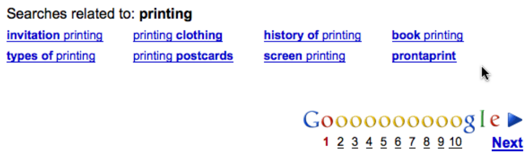

FIGURE 2.3: Query expansion in the commercial online search engine: Google.

2.2.3 Information Seeking

The research into interactive information retrieval, while proposing many advances to information retrieval, opened many more questions about users and their tasks. Saracevic (1997) produced a novel view of an information system that showed the levels of complexity of both users and computing technology, shown in Figure 2.4. From the model we can see that user interaction with a search system is driven by the users tasks, intent (or goals), and their knowledge or understanding. These each affect the kinds of query that the user produces and then enters into the interface. The design of the technology is then based around the hardware, algorithms and data that are available. The computer side of the model, which is often finite and easily understood, was later broken down into many more specific parts by Bates (2002), but it has taken much more research to begin to understand the complexity of human searchers.

Recent work by Järvelin and Ingwersen (2004) has summarised a lot of work into the context of users with the model shown in Figure 2.5. From this model, we can see that Information seeking, and the search tasks it involves, is controlled by the context of the work task the user has and their social, organisational, and cultural contexts.

Further to this model, Järvelin and Ingwersen (2004) break down Information Seeking research into nine dimensions, where datasets and algoirthms (the primary concerns of Information Retrieval) are only two. They are: Work Task, Seeking Task, Actor (or searcher), Perceived Work Task, Perceived Search Task (where perception may differ from what is actually required), Document, Algorithm, Interface, and Strategies. Referring back to the notions of Information Retrieval (IR) and Interactive Information

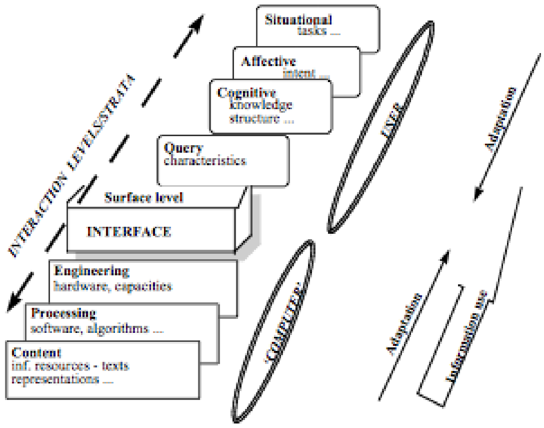

FIGURE 2.4: The stratified levels of a search system, involving both users and technology, from Saracevic (1997).

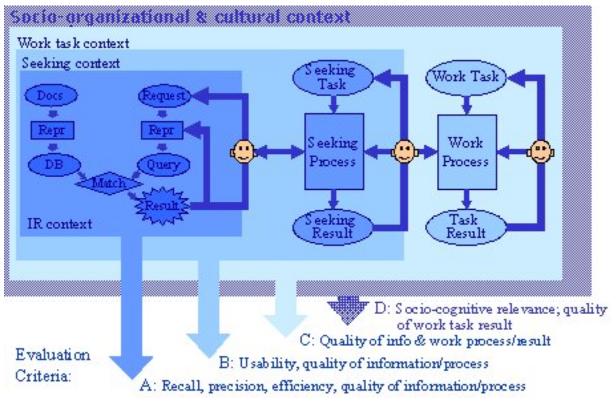

FIGURE 2.5: The model of search contexts provided by Järvelin and Ingwersen (2004)

Retrieval (IIR), Figure 2.6 shows how these fields, and Information Seeking (IS), relate to the 9 dimensions. We can see, for example, how information retrieval has focussed mainly on the search engine algorithm dimension and then how the interactive developments looked at the users, interfaces and interactions. Information Seeking research has looked more specifically at the Actors, Search Tasks and Strategies for users, and we now go over some models produced by this focus.

2.2.3.1 Actor Dimension

One of the key models of the Actor dimension was presented by Belkin et al. (1993), who broke down users into four dimensions, each with binary options, shown in Table 2.1. The dimensions are *Method*, *Goal*, *Mode* and *Resource* and in combination produce sixteen unique conditions. *Method* describes whether a user is either searching for an information object, or scanning a set of information objects. This is easily differentiated by finding a specific paper in order to get its reference details, or by searching for a possible paper, which may not exist, that can be used to support a point. *Goal*

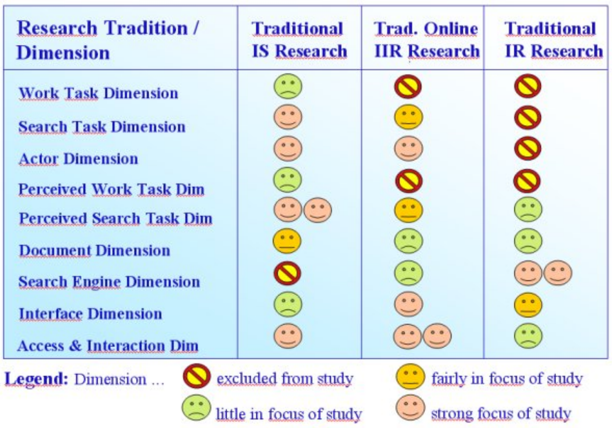

FIGURE 2.6: Research areas broken down by the dimensions of information seeking, provided by Järvelin and Ingwersen (2004)

describes whether a user is learning about something or selecting something. Using the bibliographic example differentiates these as researching a topic (Learn), or finding a reference to use (Select). *Mode* is between recognising and specifying. One might remember that there was a useful publication at CHI2005 and so is trying to identify it in the proceedings (Recognize), or may have known the author, title and year and has typed them into the ACM Portal (Specify). *Resource* is between wanting information items or meta data about an information item. Usually, with a bibliographic repository users are trying to find specific papers (Information), but it is possible that the user is trying to find out first what workshops existed in a conference so that they can better define a search query at a later point in time (Meta-Information).

ISS	Method	Goal	Mode	Resource
1	Scan	Learn	Recognize	Information
2	Scan	Learn	Recognize	Meta-Information
3	Scan	Learn	Specify	Information
4	Scan	Learn	Specify	Meta-Information
5	Scan	Select	Recognize	Information
6	Scan	Select	Recognize	Meta-Information
7	Scan	Select	Specify	Information
8	Scan	Select	Specify	Meta-Information
9	Search	Learn	Recognize	Information
10	Search	Learn	Recognize	Meta-Information
11	Search	Learn	Specify	Information
12	Search	Learn	Specify	Meta-Information
13	Search	Select	Recognize	Information
14	Search	Select	Recognize	Meta-Information
15	Search	Select	Specify	Information
16	Search	Select	Specify	Meta-Information

TABLE 2.1: Different types of users defined by four binary dimensions, from Belkin et al. (1993). These are often referred to as Information Seeking Strategy (ISS) conditions.

For example, search engines like Google poorly support users in finding meta-information (*Resource*), as a user must know which words to use in advance before she can begin to find items of information. It also provides poor support for recognising as a *Mode* because a user has to specify meta-information in the query. This means that Google primarily supports only half of the potential search conditions of users. Marchionini (2006) notes that search engines are usually concerned with precision (maximising the accuracy of the top result) than recall (maximising the number of relevant results), and so the extent of support for ISS conditions is further reduced by poor support for learning (*Goal*). This is validated by work done in 2004, which estimated that around 81% of search engine users viewed only one result page (Beitzel et al., 2004). Relevance feedback efforts, such as Google's 'Related Pages' suggestions, have tried to support the user in terms of meta-information. Yet the user would still have to fire off at least one query and then process the results before any support is provided. Google is best used, therefore, for ISS15, where the user is searching (*Method*) to select (*Goal*) by specifying (*Mode*) attributes of a specific information object (*Resource*). Subsequently it least supports users who are scanning (*Method*) to learn (*Goal*) by recognising (*Method*) some meta data about an information object (*Goal*): this is ISS2. Faceted browsing, a technique described in the next section, tries to support users by presenting all the meta-information on screen in advance and letting them choose. Conversely, this best supports ISS2, but may poorly support ISS15: useful meta-data can be embedded in long lists and it may require more effort to find them than to simply type them into a search box.

Research by Kriewel (2006) summarises research into the situational aspects of users, noting that accumulated effort, previous actions, remaining needs, technical problems, and accuracy of results are all factors that can effect the search situations of users. These search conditions were used in conjunction with the typical interaction patterns defined by Belkin et al. (1995) for each of their 16 ISS user conditions, to develop the Digital Library system: DAFFODIL, which was designed to recognise some standard situations and recommend various functions that support the different ISSs. The automatic estimation of user situations and goals is well summarised by Chi *et al.*, which then describes a method of user modelling called *information scent*. This algorithm performed well at estimating the goals of the users, and then suggested paths for further achieving them (Chi et al., 2001).

2.2.3.2 Search Task and Strategy Dimensions

Much research has investigated search strategies for information seeking (Moody, 1991; Pharo, 1999; Kriewel, 2006; Marchionini, 2006). Some more specific examples include sensemaking (Dervin, 1992), which describes a high level strategy of understanding and interpreting large amounts of information, and Information Foraging (Pirolli and Card, 1995), discussing how users follow scents of information in search. One of the more structured and informative models of search strategies, however, was presented by Bates (1990), which investigated both the levels of complexity in search strategies and search systems. First she identifies 5 levels of system automation ranging from complete user action to complete system automation. This was then combined with four levels of search activities: *Move, Tactic, Stratagem* and *Strategy*. The first of these is a single action performed by the user, either physically or mentally: mental actions may be deciding or reading. A *tactic* is a combination of *moves* and there are endless combinations of *moves* that can be used to support a *tactic*, which depends on system implementation. In earlier work, Bates (1979a,b) defines 32 specific information search *tactics*. For example, one *tactic* is CHECK, which is the ability to check decisions previously made. *Stratagems* are a larger combination of both individual *moves* and *tactics*: some examples include performing a citation search or following a footnote. *Strategies* are again higher and involve a combination of *moves, tactics* and *stratagems*: this might be finding relevant work for a paper and depends heavily on the users current overall work task.

2.2.4 Exploratory Search

More recently, the research into Exploratory Search (White et al., 2006) has focussed on the design of search interfaces for the more general information seeking strategies than the basic keyword search style of information retrieval. Referring back to the dimensions and research fields, Exploratory Search is building on the Interface dimension to support the wider knowledge of Search Tasks and Actor dimensions provided by Information Seeking. To support the design of more exploratory search interfaces, Marchionini (2006) identified a number of typical activities that are shown in Figure 2.7.

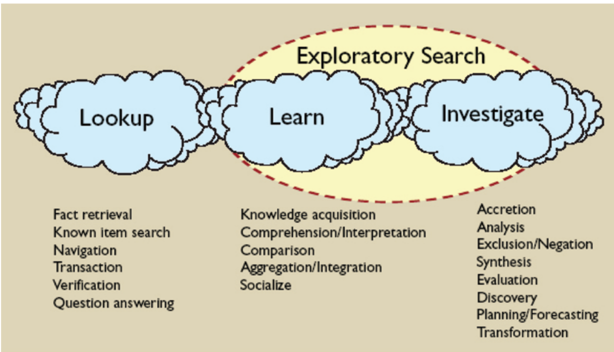

FIGURE 2.7: The activities associated with traditional and exploratory search, from Marchionini (2006).

Much research has gone into the design of exploratory systems, such as: interacting with automatically or carefully constructed classifications of a dataset (Hearst, 2006); how browsers can be used on mobile technology (Wilson et al., 2006); how audio or other multimedia can support exploration (schraefel et al., 2004); how users can learn from browser design (Wilson et al., 2007); and how users browse and search in real systems over time (Wilson and m.c. schraefel, 2008). Below we discuss some of the interfaces that have been developed to support more exploratory forms of search and general information seeking.

2.3 Search and Browsing Interfaces

2.3.1 Encouraging Exploration with Classifications

One of the more common approaches to supporting more exploratory forms of search has been to develop classification schemes to provide structure to the set of documents being searched. Popular online examples are directories, such as Yahoo Directory [1] and Google's Directory[2] shown in Figure 2.8. Such directory schemes provide a thematic hierarchical classification that can be used to find websites by their topics.

FIGURE 2.8: The Google directory provides a hierarchical classification scheme.

Hierarchical classifications, however, are usually limiting, as there are many alternatives to thematic constraints that could be useful to users. Faceted browsing (Hearst, 2000) has become an alternative approach where multiple, possibly hierarchical, classifications are applied simultaneously to a dataset, so that users may use some or all of them to help find related information. Often, providing a faceted classification involves knowing the dataset in detail and carefully constructing the facets. Hearst (2006) has shown that careful construction will always provide benefits over automatically generated classifications based on clustering. This is not always possible with datasets that are continuously developing, like the web.

[1] http://dir.yahoo.com - Yahoo! Directory
[2] http://directory.google.com/ - Google Directory

2.3.2 Traditional Faceted Browsers

In the traditional design of faceted browsing, used by browsers such as Flamenco (Yee et al., 2003), shown in Figure 2.9, and the Relation Browser (Zhang and Marchionini, 2005), shown in Figure 2.10, facets are interdependent and each affect each other. For example, applying a constraint in one facet filters all of the facets present to show meta-data relating to the selection.

The Flamenco browser supports both keyword search and faceted browsing, accounting for both those who know their target and those who don't have much knowledge about the domain. The initial display shows all the possible facets in two columns, with vertical scroll as necessary. Here the user can either make an initial selection from the facets or use the search box, which is consistently at the top left (unless viewing a target object). By entering a search query or selecting an item in one of the facets, the user is moved away from the initial view to one where all the facets are listed vertically down the left column, with the search box remaining at the top left. A "breadcrumb" (Bonnie Lida and Pilcher, 2003) is presented at the top right, which presents a visualisation of the path of selections made by a user. A search term acts as a domain filter and the search results (displayed in the remaining space at the bottom right) may still be browsed using the facets. If the search term can be matched to particular items in the facets, these are presented to the user above the breadcrumb.

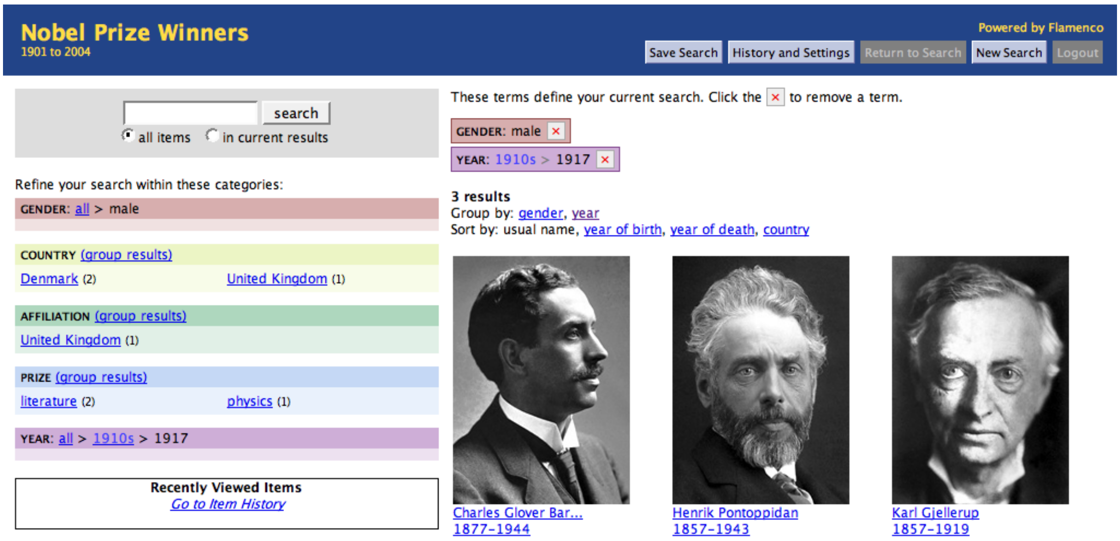

FIGURE 2.9: The Flamenco Browser, showing nobel prize winners with a faceted classification.

When a selection is made in a hierarchical facet, the sub-categories within the facet are shown and a per-facet breadcrumb displays the selection made. If there are no sub-items, the facet is effectively minimised (facet representations grow and shrink with the number of options within it). If facets are hierarchical, results are automatically clustered into the sub-categories of the latest selection. The user may optionally group the results by any other facet through a single interaction provided by the presence of a link along side of each facet name. Any potential option for selection is accompanied by numeric volume indicators (NVIs - Wilson and schraefel (2006)), to estimate the number of target objects that can be reached by its selection.

When target object selections are made, the user is moved away from the faceted browser display to one that shows a summary of the data associated with their choice. From here, the user is given options to return to the faceted browser: extra facet selections can be made to expand or further narrow their constraints and view similar objects. Users may also reset the interface by pressing the 'New Search' button.

The relation browser interface currently presents all the facets and their contents persistently: these facets are listed across the top of the UI and grow/shrink to fit on the screen. Users can reorder columns by using a drop down list that formulates as both a mechanism for changing the facet and also for acting as its label. The ordering of columns is purely The user can make facet selections in any order and the temporary hierarchy built is controlled by this selection order: this breadcrumb order is not currently visualised. NVIs are represented a single-bar bar graph behind each item in each facet. The population of the bar represents both the number of achievable target objects from making that selection and, uniquely, the number of total target objects in the dataset. The exact figure is represented as an NVI to the left of each label. Hovering over items in each facet previews the affects of the selection on each of these NVIs and is made persistent by actually clicking.

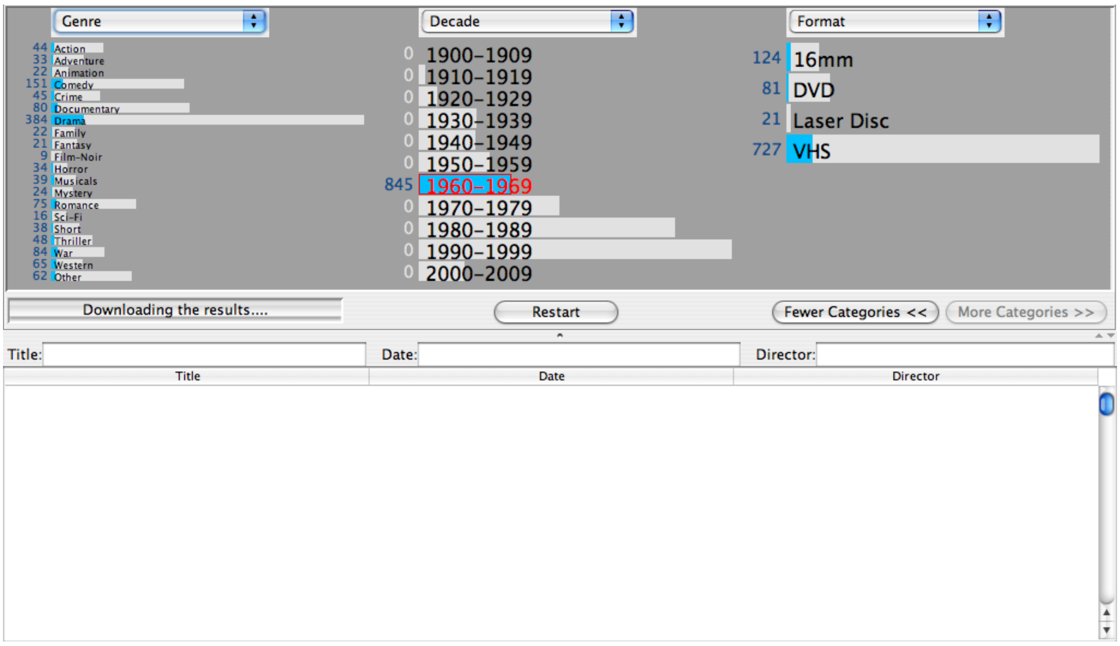

FIGURE 2.10: The Rave Browser, showing a movie archive with a faceted classification.

By pressing the *search* button, results are displayed in the lower half of the screen, where items can be filtered, sorted and individually selected. Once the search results are displayed, the previous selections above are transformed into a label representing the selections, much like a breadcrumb but without temporal order. The facet browser is also transformed to represent the subset of target objects that had been previously achieved through facet selection. Thus NVIs represent the number of target objects in the new subset. Any subsequent facet selections automatically filter the search results. Upon selection in the results, the target object is displayed in a new window.

2.3.3 Directional Column-Faceted Browsers

An alternative approach is to provide a directional column-faceted browser like iTunes[3] or mSpace (schraefel et al., 2006), shown in Figure 2.11. Here, instead of providing facets that uniformly affect each other, interfaces like iTunes present facets in a row that affect each other from left-to-right only. This direction means that users can see both: all of the artists in a selected genre and all the albums by a selected artist. Traditional forms of faceted search would only show the selected genre, artist, and albums.

Figure 2.11: The mSpace browser, showing a news video archive with a directional column-faceted classification.

[3] http://www.apple.com/itunes/ - Apple - iPod + iTunes

mSpace presents facets as columns to create a hierarchy through the meta-data from left to right across the browser; we call this a slice. When the browser loads, all facet columns are fully populated. If a user starts by clicking on something in the first column, the columns to the right are filtered to show related items that are associated with the selection. By next selecting something in the second column, the columns to the right of the second selection are filtered again, but the relationships shown between the first two columns are maintained. The user may, however, click on something in any column at any time and everything to the right of a selection is filtered. Any relationships to the left of a click, in columns that do not have a selection) are highlighted instead to help the user learn about the dataset (Wilson et al., 2007) and find the paths to the items they have selected. For example, should the user start in the third column, the related items in the first and second columns are highlighted, but not filtered, to maintain the left-to-right structure. Similarly, if the user starts by clicking on something in the first column, and then clicks in the third column, items in the second column that bridge the gap are highlighted. This combination of left-to-right and backward (or leftward) highlighting provides the user with benefits of both traditional and directed faceted browsing.

As the order of columns provides additional information to the user, mSpace provides easy interactions to let the user change the order of the slice. Users may add, remove and reorder the columns through direct manipulation. To remove a column, they can click the **x**; this matches familiar software design of most operating systems. This column then gets listed with the set of optional facet columns. Any one of these optional columns can be added to the slice by dragging it to the desired place. Any column already in the slice can be easily moved around with the same action.

To help users find items in a column, which could be very long when one column often shows the names of all of the documents in a dataset, they can use the in-column filter. This filter can be opened by pressing the small magnified glass on each column. As a character is typed into this box, the list is filtered to only items that contain that character. Each item in the columns has a number, currently this number shows the number of system objects (like movie in the example above) that can be found by making that selection. Each item can also have a Preview Cue icon; hovering over this icon will trigger a multimedia preview to help the user make decisions (schraefel et al., 2004).

The information panel is often a large part of an mSpace design, as it provides space for a portion of content to be displayed about a selected item. For example, if an Actor was selected, information about that actor may be displayed in the information panel. Further results relating to the actor, or based on the current set of selections and the order of the slice, are listed below to help the user jump to straight to some movies without.

The final key section of the mSpace is the collection space. This supports information triage (Marshall and Frank M. Shipman, 1997) by allowing users to keep any item in any column for later. This is similar to the work done by schraefel et al. (2002), as smaller-than-page sized 'nuggets' of information can be stored. Users can double click on any item in the columns and it will be added to the Interest space, where they can be tagged. Further social interactions are included to comment on clips and discuss them in user groups.

2.4 Producing an Evaluation Framework

As environments for evaluating information retrieval systems and for evaluating interactive information retrieval systems were built to meet the desire to test potential improvements to search, the research presented in this report is developing a framework for evaluating more exploratory information seeking interfaces. Consequently it is important to consider related work on the design and evaluation process of user interfaces, so that we can see where the framework may be used. Further, we must consider how such frameworks should be built to be reliable and used in confidence.

2.4.1 The Interface Design Process

The design of user interfaces has been researched by the HCI community so that software is built considering the needs and context of real human use and is well summarised by Shneiderman et al. (2006). Typically, it is suggested that many techniques be used in combination to design a user interface properly (Shneiderman and Plaisant, 2006). Requirements Gathering (Holtzblatt and Beyer, 1995) is usually carried out first, using techniques such as interviews, questionnaires, and ethnography (Hammersley and Atkinson, 1995). With the requirements, rapid prototyping (Rettig, 1994), heuristic evaluation (Nielsen and Molich, 1990), and cognitive walkthroughs (Wharton et al., 1994) are examples of techniques that could be used to build early designs of a system. Designs are then usually tested with users, who are given tasks, to see how well interfaces perform. Evaluations often use measures like accuracy and speed to assess an interface, but other approaches have looked at the ability to achieve an overall goal, or how much users might have learned during a set period of time (Wilson et al., 2007). The evaluation process then often produces final designs that are involved in longitudinal studies, such as the one carried out on mSpace by Wilson and m.c. schraefel (2008).

The framework described in the next chapter is designed to support the early prototype design phase for search systems, by rigourously assessing the support they provide for users. The framework could be used alongside techniques such as rapid prototyping, cognitive walkthroughs, and heuristic evaluations.

2.4.2 Developing a Framework

Finally, this research aims to produce a framework, and the last important area of related work to consider is about the method of doing so. Although there is no specific guide to producing a framework, there are examples of research that have followed similar patterns to successfully produce them. One good example is the method designed to produce a multi-point scale for questionnaires by Peterson (2000), which goes through the phases of: Theory, Development, Application, Validation, Extension and then final Evaluation. In evidence that this process has been successfully applied to producing a framework, O'Brien and Toms (2008) used a modified four-stage version of this approach that has gone through, Design, Validation, Extension, and Evaluation. Further, the development of assessments techniques like the GOMS model has followed similar processes. After proposing a design (John et al., 1985), the GOMS approach was validated with an example study (John and Newell, 1987) and extended (John, 1990). In fact the GOMS model was validated and extended many times thereafter by other authors (Gray et al., 1992; Gong and Kieras, 1994). The work in developing an evaluation framework below follows these same stages.

Chapter 3

A Proposed Framework for Measuring Support of Search Interfaces

From the search and information seeking research above there are still open questions that have been reinforced by experts in the fields (Järvelin and Ingwersen, 2004). There is a specific gap in information research that should be addressing the support provided by search interfaces for broad types of users and for the broad range of their possible search tasks. The framework proposed below aims specifically to assess, as far as possible, the full range of possible search tasks that a search interface provides. Further, it assesses these tactical options from the view of 16 user types, according to dimensions such as their existing knowledge and clarity of their perceived goals.

This chapter contains two main parts. First, the proposed framework is described in terms of its inputs, processes and outputs. Given this description of the framework, three examples of its use are presented that build up to reveal the strengths of its analyses. The first example is an analysis of keyword searching, which is a familiar method of search for many, and shows how the framework can be used to better understand a single design. The second example focusses on the comparison of two designs for a single design idea. The third, and most complex example, shows the full extent of the framework's capabilities by comparing three well developed advanced search interfaces.

3.1 Proposed Framework

The proposed framework, also described by Wilson and schraefel (2007); Wilson et al. (2008), combines two established models from Information Seeking. First, we use the two lowest and well-defined levels of Bates' model of seeking strategies (Bates, 1979a,b) to assess both the range of *tactics* that can be employed in an interface and then how they can be achieved, by number of *moves*. Second, we use the model of different user types, defined by Belkin et al. (1993), to simultaneously assess an interface from many perspectives. The two models are combined with a novel mapping between the needs of different user types and the *tactics* they might choose to employ.

Another way to imagine the framework is that it looks at user interfaces through two sets of filters, shown in Figure 3.1. At any one time, a user is viewing the user interface (UI) from one of 16 user conditions, and sees it in terms of the tactics she can employ. The interface can be seen by each tactic in a different way, in terms of how easy it is to employ that tactic across its interactive features. Bates moves are used as the metric between the layers. So each tactic has a total score of how many moves they can make on the UI. In turn, when a user looks at the potential tactics through one of the 16 user conditions, they see how many moves they can make with each tactic. The aim for any advanced IR system should be that a user can see many possible moves with each tactic, where each tactic has many possible moves to play on the interface.The exact procedure involved and type of analyses produced are described below.

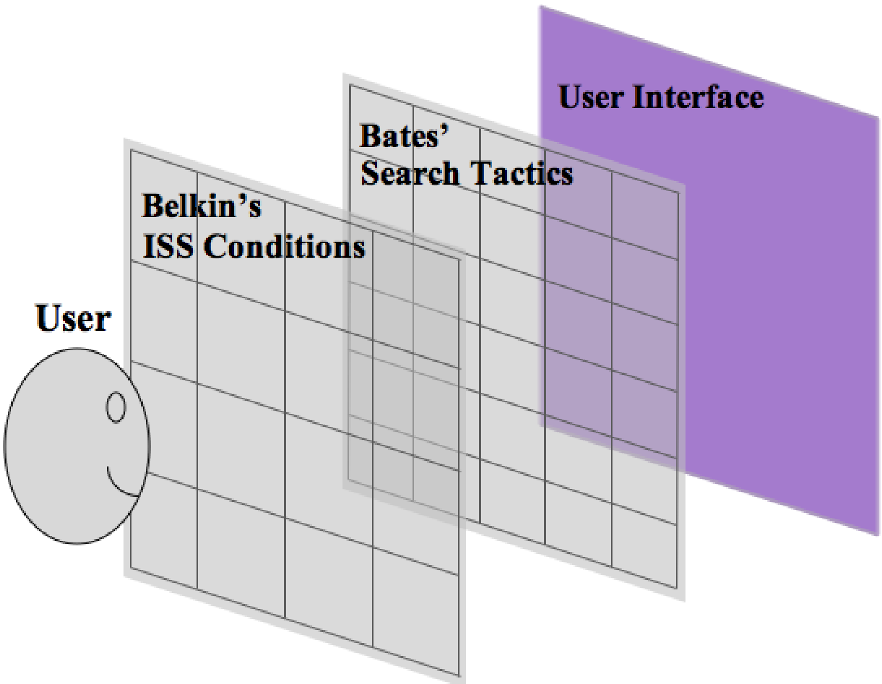

FIGURE 3.1: The interaction of the models within our evaluation framework. The parts of each layer act as a viewfinder onto the next layer, from Wilson et al. (2008).

3.1.1 Procedure of Using the Framework

The process of using the framework involves one very small but key step that is repeated within three encapsulating loops, and is shown in Figure 3.2. The most outer loop of the process, Loop L1, is to repeat the main step for each of the interfaces that is involved in the assessment. In the example shown in Section 3.2.3, three faceted browsers are compared, and so, for L1, the main step is repeated for each of the three browsers. The second loop, Loop L2, is to perform the step for every feature of the interface, such as the keyword search box, the results list, and the facets. For example, given that we are comparing three browsers, the main step is repeated for every feature within each browser. The third and final loop, Loop L3, is to repeat the main step for every possible *tactic* (Bates, 1979a,b), that users may want to carry out. The labels L1-L3 will be used in reference in the examples in the rest of the paper. Combined together, we are assessing the ability to carry out each possible *tactic* with every feature of each interface being analysed. The main step, therefore, is to calculate the support provided for each possible *tactic* for each of the features of each of the interfaces.

```
L1. For each interface
    L2. For each feature in the interface
        L3. For each tactic defined by Bates
            Count how many moves the user must
            make to achieve the current tactic with the
            current feature.
        L3. Next tactic
    L2. Next feature
L1. Next interface
```

FIGURE 3.2: The step of measuring support, by counting *moves*, is encapsulated by three loops: each *tactic*, each interface feature, and each interface.

To calculate the support for each *tactic*, by each feature, of each interface, we use the notion of a *Move* from Bates' model. As mentioned above, a *move* can be either mental or physical. When entering a keyword search, for example, a user might choose a search term (*move 1 - mental*), enter the search term (*move 2 - physical*), and press the search button (*move 3 - physical*). More about the keyword example is described in Section 3.2.1. Instead of simply measuring how a feature is used, we measure how many *moves* it takes to achieve each of the *tactics* described by Bates. It may be that to achieve certain *tactics*, the user may use the feature as normal. With some *tactics*, however, such as FIX and REARRANGE, the action may involve changing a previous query, which may involve additional *moves* like choosing the term that should be changed (mental). Ideally, users can achieve their *tactics* in as few moves as possible, and so early indicators of good designs will be to enter low numbers.

To record the number of *moves* it takes to achieve each *tactic* with each feature of each interface, the evaluator is provided with a set of tables, one for each interface. An example table is shown in Table 3.1.1. Each table has the list of features found in the interface down the left hand side and the list of *tactics* across the top. The user simply enters the number of *moves*, which may be 0, that it takes to achieve the *tactic* with the feature in the cell where the two intersect and in the appropriate table for the current interface.

Interface Name	**List of *Tactics***						**Totals for Features**
List of Features	2	0	3	0	1	1	7
	0	1	0	1	0	0	2
	1	2	1	4	0	1	9
Totals for *Tactics*	3	3	4	5	1	2	

TABLE 3.1: Example input table for one browser, where the number of *moves* is entered in the intersection of the features, listed down the left, and the *tactics*, listed across the top.

There are three caveats to this input process that must be noted. First, repeat *moves* are ignored in counting. Altering two or more terms in a keyword query, for example, is counted as if the user was changing only one, as we cannot assume any specific number of terms and cannot count an infinite number of changes. Similarly, the second caveat is to ignore optional *moves*. We cannot predict, for example, whether the desired item that a user might be searching for will be in a short list of results, at the top of a long list, or at the bottom of a long list. The model is designed to be general to any datasets and instead measures the interface. Consequently, we ignore optional *moves* such as scrolling. The final caveat is that after inserting all of the numbers, the figures are automatically inverted to be a value up to a maximum of 1. Without inversion, optimal designs support users in one or two *moves*, but being unable to achieve a *tactic* would be represented by 0. By inverting the figures, one *move* becomes the largest number, and any extra *moves* decrease towards 0, which represents no support. These inverted figures can then used to produce the analyses of the framework presented below.

3.1.2 Internal Mapping Used by the Framework

As part of the analysis process, a novel mapping was produced (also described by Wilson et al. (2008)) to decide on which *tactics* are required by each type of user. Producing this mapping is by no means trivial, as the relationships are usually many-to-many. Each *tactic* cannot be, as well, obviously or clearly connected with any specific value of Belkins dimensions. Further, as well as being difficult to state that a *tactic* x is associated with Dimension value A, we cannot easily calculate the amount that Dimension A is supported. For example, we cannot tell if a *tactic* is key to the users needs, or secondary to other *tactics*.

The only source available to generate this mapping is the definitions given in the research surrounding the two models. Consequently, the first mapping produced (Wilson et al., 2008), shown in Table 3.2, was built through careful research, literature review and interpretation. This careful work, and the produced mapping, was one of the main tasks and contributions of the first period of the research, reported by Wilson (2007). The other main contributions from this period were the novel method of integrating two models to produce a new metric, and the analysis of three faceted browsers, which is described as an example of use in Section 3.2.3. The recent work, in preparation to upgrade to a full doctoral program, has focussed on validating the framework, including this novel mapping, and is described in Section 4.

ISS	Tactics
1	CHECK, WEIGH, RECORD, SURVEY, EXHAUST, PARALLEL, SUPER, RELATE, NEIGHBOUR, RESCUE, BREACH
2	CHECK, WEIGH, RECORD, SURVEY, STRETCH, EXHAUST, PARALLEL, SUPER, RELATE, NEIGHBOUR, RESCUE, BREACH
3	CHECK, CORRECT, RECORD, CUT, SPECIFY, CLEAVE, EXHAUST, PARALLEL, BLOCK, SUPER, RELATE, NEIGHBOUR, REARRANGE, CONTRARY, RESPELL, RESPACE, RESCUE, BREACH
4	CHECK, CORRECT, RECORD, CUT, STRETCH, SPECIFY, CLEAVE, EXHAUST, PARALLEL, BLOCK, SUPER, RELATE, NEIGHBOUR, REARRANGE, CONTRARY, RESPELL, RESPACE, RESCUE, BREACH
5	WEIGH, RECORD, SELECT, SURVEY, SCAFFOLD, EXHASUT, PARALLEL, SUPER, RESCUE, BREACH
6	WEIGH, RECORD, SELECT, SURVEY, STRETCH, SCAFFOLD, EXHAUST, PARALLEL, SUPER, RESCUE, BREACH
7	CORRECT, RECORD, SELECT, CUT, SCAFFOLD, SPECIFY, CLEAVE, EXHAUST, PARALLEL, BLOCK, SUPER, REARRANGE, CONTRARY, RESPELL, RESPACE, RESCUE, BREACH
8	CORRECT, RECORD, SELECT, CUT, STRETCH, SCAFFOLD, SPECIFY, CLEAVE, EXHAUST, PARALLEL, BLOCK, SUPER, REARRANGE, CONTRARY, RESPELL, RESPACE, RESCUE, BREACH
9	CHECK, WEIGH, PATTERN, BIBBLE, SURVEY, REDUCE, PINPOINT, SUB, RELATE, NEIGHBOUR, TRACE, VARY, FIX, FOCUS
10	CHECK, WEIGH, PATTERN, BIBBLE, SURVEY, STRETCH, REDUCE, PINPOINT, SUB, RELATE, NEIGHBOUR, TRACE, VARY, FIX, FOCUS
11	CHECK, PATTERN, CORRECT, BIBBLE, CUT, SPECIFY, CLEAVE, REDUCE, PINPOINT, BLOCK, SUB, RELATE, NEIGHBOUR, TRACE, VARY, FIX, REARRANGE, CONTRARY, RESPELL, RESPACE, FOCUS
12	CHECK, PATTERN, CORRECT, BIBBLE, CUT, STRETCH, SPECIFY, CLEAVE, REDUCE, PINPOINT, BLOCK, SUB, RELATE, NEIGHBOUR, TRACE, VARY, FIX, REARRANGE, CONTRARY, RESPELL, RESPACE, FOCUS
13	WEIGH, PATTERN, BIBBLE, SELECT, SURVEY, SCAFFOLD, REDUCE, PINPOINT, SUB, TRACE, VARY, FIX, FOCUS
14	WEIGH, PATTERN, BIBBLE, SELECT, SURVEY, STRETCH, SCAFFOLD, REDUCE, PINPOINT, SUB, TRACE, VARY, FIX, FOCUS
15	PATTERN, CORRECT, BIBBLE, SELECT, CUT, SCAFFOLD, SPECIFY, CLEAVE, REDUCE, PINPOINT, BLOCK, SUB, TRACE, VARY, FIX, REARRANGE, CONTRARY, RESPELL, RESPACE, FOCUS
16	PATTERN, CORRECT, BIBBLE, SELECT, CUT, STRETCH, SCAFFOLD, SPECIFY, CLEAVE, REDUCE, PINPOINT, BLOCK, SUB, TRACE, VARY, FIX, REARRANGE, CONTRARY, RESPELL, RESPACE, FOCUS

Table 3.2: Table showing Bates' *tactics* for each of Belkin's ISS conditions according to the original mapping

3.1.3 Analyses Produced by the Framework

Three types of graph are produced by summarising the inverted figures inputted during the procedure described above: G1) an analysis of the support contributed by each feature of the interface; G2) an analysis of the support provided for each type of seeking *tactic*; and G3) an analysis of how each user type is supported, according to the novel mapping described above. The labels G1-G3 are used to refer to the three types of graph, rather than to specific instances, for the remainder of the report. Instances of each graph are shown in the examples of use below, but first we describe their purpose and the information they reveal about designs.

Graph G1 is designed to analyse the different features within each browser, such as keyword search, facets, query suggestions, spelling corrections, etc. By summarising the support each feature contributes to an interface, we can analyse the design in three ways. First, the structured programmatic approach simply, but thoroughly, identifies features that are included in one design but not in another. Second, the same approach means that we can identify novel features of a design and the strength of support it contributes to the overall design, where strength is a term for how simply it can be used to achieve *tactics*. Third, as we are summarising a metric of support, we can compare multiple approaches to providing the same feature to see which is stronger and, therefore, usually the better choice for implementation.

Graph G2 takes the opposite approach and summarises the support for each *tactic* across all the features of the designs. Consequently, the graph shows how easily each design supports a user in being able to check what they've done, for example, or change their earlier decisions. Graph G2 also provides three types of analysis. First, the graph shows which, if any, of the tactics are not supported by a design. This can be used to consider how a design could be altered so that it does support any missing tactics. Second, if a design uniquely supports a certain *tactic*, then the contributing features can be used to inform the rest of the design process. Third, if multiple designs provide different levels of support for each tactics, then the summarised metrics can reveal which version provides the most desirable range of support.

Graph G3 is the product of the novel mapping described above, which is used to summarise the recorded metrics for each type of user. For example, it could tell us that people who are confident with the interface and know what they are looking for are well supported, but users who are arriving for the first time, or are exploring an unfamiliar topic, may be less well supported. Further, when used to compare multiple designs, the affect of different approaches to feature design can be seen for each user type. Individually, the support for each user type can be seen. As the 16 user types created by Belkin et al., however, are created by combinations of dimensions with two values, the graph is interpreted in patterns of halfs, quarters, eigths and pairs. An increased height in the first half of the graph, for example, shows that it is easier for people to Scan (as their Method) than to Search. If the first and third quarters are higher than their counterparts then there is an increased ability to Learn (as their Goal) rather than to Select. To support interpreting the graph, analysis should be performed in reference to Table 2.1, where each user type is defined. The interpretation of this Graph G3 should become clearer with the examples below.

3.2 Examples of Use

To better understand how the framework and its analyses can be used, three examples are discussed below. First, an analysis of keyword searching is presented, which is a familiar interface feature for many information seekers, especially on the web with services like Google. This simple and familiar keyword example should help to understand how the basic application of the framework works. Following this, two versions of a single new feature to support faceted browsers, called Backward Highlighting, is presented to show how the framework can be used to analyse new ideas and multiple potential designs. Finally, three faceted browsers are compared to show how we can analyse larger and more complicated interfaces.

3.2.1 Keyword Searching

Existing interfaces or their individual features can be analysed by the framework so that we can better understand when, and for whom, they work well. In this example, we analyse keyword search, which is usually included in most search interfaces, especially online search engines. The value of loop L1 in this case is 1, as we are only evaluating the single design of Google. The value of loop L2 is also 1, as we are evaluating the keyword search feature only. Finally, the value of loop L3 is 32, which is fixed by the number of Bates' *tactics*. Thus, we will be assessing the support provided by the keyword search box of Google for each of the 32 tactics (Graph G2), and subsequently the user types (Graph G3). Graph G1 is not presented as it would only contain one bar representing the keyword search box of Google.

According to Graph G2 (Figure 3.3), which reveals the support for different search *tactics*, we can see that the particularly well-supported *tactics* are CHECK, VARY and RESPELL. CHECK is easily supported as Google shows the user what they have just searched for in a search box at the top of the results list. VARY and RESPELL are well supported, in a single move, by the query expansions and the spelling suggestions that are suggested by Google along with search results. Overall, we can see a consistent amount of support for the Search Formulation and Term *tactics*, which are core to the design of keyword search and defining search terms. The good support for monitoring *tactics* such as CHECK and PATTERN, however, is due to the careful design which makes sure users are clear about the consequences of their actions.

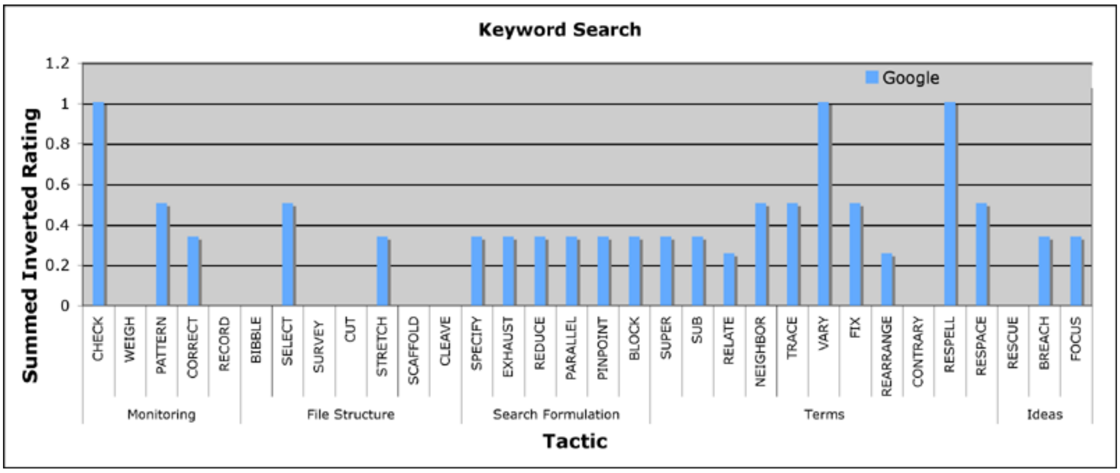

FIGURE 3.3: Graph G2 showing the support provided for each *tactic* by keyword search, where taller bars represent stronger support for a *tactic*.

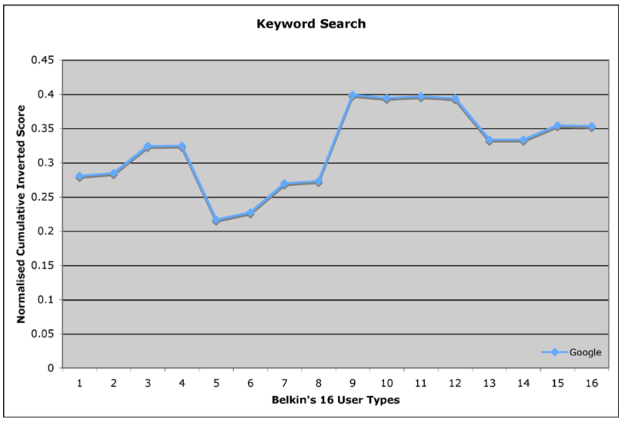

FIGURE 3.4: Graph G3 showing the support provided for each of Belkin's user types by keyword search, where peaks represent stronger support for a user type.

From reading Graph G3 (Figure 3.4) in patterns (as explained above), we can immediately see, as expected, that the latter half of the graph, which represents users who are searching and know what they are looking for, is much higher than the first half. We can also see, especially in the first half of the graph, that the even eighths are higher than the odd eighths, which represents increased support for specifying and decreased support for recognising.

3.2.2 Measuring the Backward Highlighting Technique

Wilson et al. (2007) report on a study that examined a new feature of mSpace called Backward Highlighting. In brief, the new feature was designed to enhance facets in directional column-faceted browsers such as iTunes and mSpace. As such browsers only filter from left-to-right, in order to provide additional data and options to the user, certain relationships are not conveyed from right-to-left that would be by traditional faceted browsers, such as Flamenco and RB++. More detail on this problem is included by

Wilson et al. (2007). Backward Highlighting, therefore, was designed to improve directional column-faceted browsers by highlighting the right-to-left relationships (backwards against the flow of filtering) so that the user receives a faceted experience that includes the best of both styles.

In designing the best implementation of Backward Highlighting, we considered two designs, where the second is referred to as Bucket Highlighting. The design of Bucket Highlighting is different in that it groups the highlighted, and thus related, items together, but at the cost of reducing the screen space for the column. In this study, the value of L1 (number of interfaces) is 2, as there are two designs of the new feature. The value of L2 (number of features) is 1, which is the Backward Highlighting technique alone. The move data (main step) for each tactic (L3) was entered for the one feature in both designs.

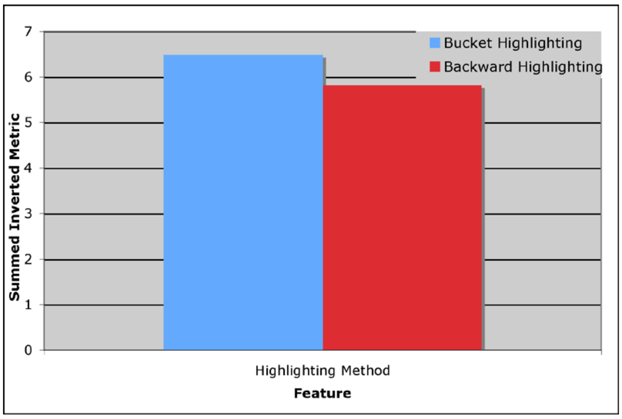

FIGURE 3.5: Graph G1 showing the support provided by the two different designs of Backward Highlighting, where taller bars represent stronger support provided by a feature.

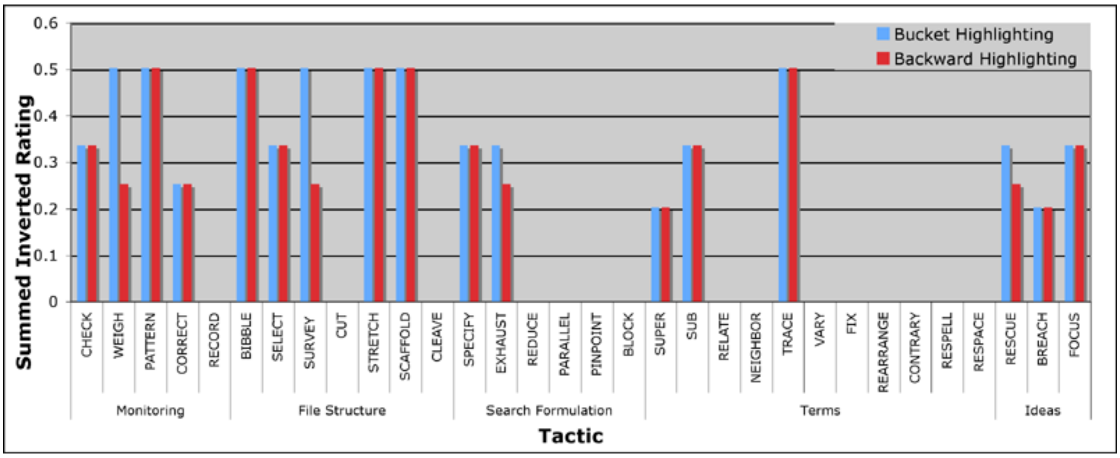

FIGURE 3.6: Graph G2 showing the support for each *tactic* provided by the two designs of Backward Highlighting, where taller bars represent stronger support for a *tactic*.

As there is only one feature, but two designs, Graph G1 (Figure 3.5) only shows one feature (the new technique) and suggests that Bucket Highlighting design is slightly stronger. Graph G2, shown in Figure 3.6, shows us in which tactics the Bucket Highlighting design is stronger. In particular, it supports tactics such as WEIGH and SURVEY, as all of the highlighted items are together and so the user can assess the set of related meta-data more easily.

Graph G3 tells us that it least supports user type 3 and most supports user type 15. Referring to Table 2.1, user type 3 is in the position where they are trying to learn about specific information by scanning for it. User type 15 is one who is searching to select specific information. By assessing the patterns of the graph, the second half of every pair is higher, so Backward Highlighting is better for users interested in the meta-information, i.e. the content of the facets, which are being highlighted and presented to the user. The odd eighths are also higher than the even eighths, showing that the tool makes it easier for users to recognise things than to specify them. The even quarters and the latter half of the graph are slightly higher indicating that the tool helps users to search for and select things, than to scan and recognise them. Finally, in comparing the two implementations, Bucket Highlighting, which groups the highlighted items, more specifically supports the user types that are recognising meta-data, shown by the difference in the odd eighths.

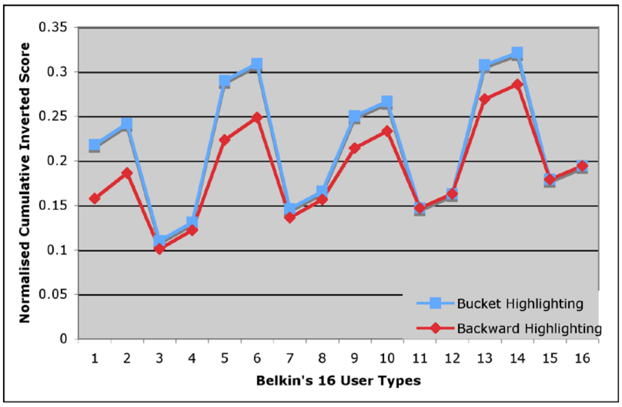

FIGURE 3.7: Graph G3 showing the support for each type of Belkin's user by the two designs, using the original mapping, where peaks represent stronger support for a user type.

3.2.3 3 Faceted Browser Comparison

In previous work (Wilson and schraefel, 2007; Wilson et al., 2008; Wilson, 2007), the framework was applied to three faceted browsers: mSpace (schraefel et al., 2006), RB++ (Zhang and Marchionini, 2005), and Flamenco (Yee et al., 2003). When applying the framework to the three browsers, the value of L1 is three and involves repeating the enclosed steps for mSpace, then for RB++, and then for Flamenco. The value of Loop L2 is 12, which is the unified list of the all features of the three browsers, and involves stepping through each of the them in turn. Then for each feature, to ask how many moves it takes to achieve each of the *tactics* (L3 - the value of which is always 32, according to the list of *tactics* defined by Bates).

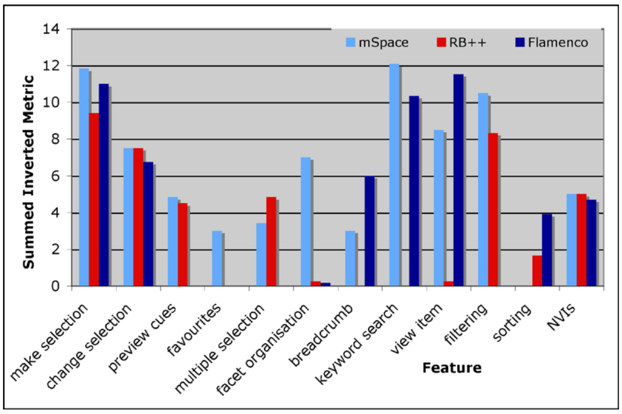

FIGURE 3.8: Graph G1 showing the support provided by each feature of each faceted browser, where taller bars represent stronger support provided by a feature.

Graph G1 (Figure 3.8), shows the significant contribution of different interface features. First, the slight drop in the Flamenco bar for changing a selection reflects the four steps required compared to the 2 and 3 steps required by mSpace and RB++. It may also be noted that Flamenco has no preview cue, and thus the appropriate bar is absent from the graph. The ease of multiple selections in RB++ is also clearly shown. One feature to compare is 'View Item'. RB++ has a significant drop in support here, as the implementation has a significant separation between Target Objects and Browser. Target Object pages may be simply launched in a separate window, but there are no ways in which the user can interact with the original browser when viewing them. The only option is to close the window and return to the browser. In Flamenco and mSpace, users can make further selections from the Target Object page that cause automatic interactions with the facets. An example is selecting an item of related metadata, which is then applied as an additional constraint to the search. This is most obvious in mSpace where the facets are always present, even when viewing a Target Object page.

mSpace has no sorting function, which is shown clearly on the graph, but is well supported by RB++ and Flamenco. In Flamenco, a user is able to group the results by any of the facets in the system and provides the strongest implementation of a sorting method. However, Flamenco does not support filtering. In mSpace, user can filter long lists of items in facets to jump quickly to selections. RB++ also provides the filtering of Target Objects by reusing the facets as filters: this support is only for Target Objects and is thus a weaker implementation. The in-browser collection space in the mSpace interface clearly provides support for the interface and is also unique to mSpace.

Graph G2 (Figure 3.9) shows the support provided by each interface for each of the 32 known tactics. A number of observations can be drawn from Figure 3.9. First, each interface has a high bar for SURVEY. This is expected when evaluating faceted interfaces because the user is presented with optional selections at each stage. Such a high bar would not be so visible in keyword only interfaces, like Graph G2 (Figure 3.3).

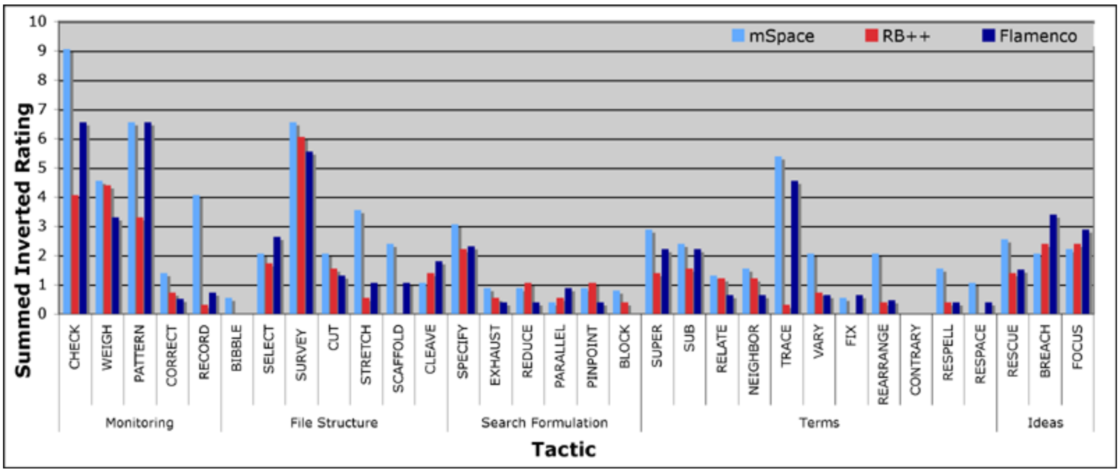

FIGURE 3.9: Graph G2 showing the total support for each *tactic* provided by each faceted browser, where taller bars represent stronger support for a *tacitc*.

The first tactic, CHECK, has different levels of support in all three interfaces: this tactic is to see what actions have been made to corroborate them with the current aims. In RB++, although previous selections are highlighted in the interface, no representation of order is given and so a lower support for checking ones actions is provided. In Flamenco, this feedback is given in a breadcrumb, and is visible when navigating through the facets. To view a Target Object in Flamenco, the user is moved to a new page with a summary of that object. Thus, before the user can view the breadcrumb, they must first return to search: this requires two moves. In mSpace, breadcrumbs are embedded into the ordered facets. As mSpace is a focus+context browser, the user can view the facets and their previous actions at all times, including when viewing a Target Object. This leads to a taller bar for mSpace and then Flamenco in Graph G2 (Figure 3.9).

The large difference in the score assigned to the support for the RECORD tactic suggests that the interactions for saving information in mSpace are much simpler than those in Flamenco and RB++. The mSpace interface includes a within-browser collection space that can store any object in the facets. Although any state reached in Flamenco and mSpace can be saved using the parent application, and pages displaying Target Objects in all three interfaces can be saved in this way, a single double-click move can store facet items in the Interest panel of the mSpace browser at any point: even when viewing a Target Object it can be saved with by double-clicking or dragging the item into the box.

There is also a significant peak over the STRETCH and SCAFFOLD tactics for the mSpace browser. STRETCH, reusing objects in unintended ways, is highly supported because of the explicit ordering of facets. The reordering of facets allows users to see the effects of meta-data on other meta-data: this reordering involves a single dragging action. SCAFFOLD, finding quick paths to Target Objects, is highly supported, because selecting preview cue objects will bring up not only information about its Target Object, but can also be used to see its position in the facets. Users may recover a path used to find items in the Interest panel by dragging it onto the columns or double-clicking the item, displaying a quick jump to a previous path.

It may be noted that mSpace is specifically higher over all of the Term Tactics (SUPER to RESPACE). Also, with the exception of SPECIFY, none of the interfaces support the Search Formulation Tactics (SPECIFY to BLOCK) very well. It may also be noted that no interface supported CONTRARY, which is finding an antonym of a selection. After investigation, these higher ratings in mSpace are supported mainly by a combination of features. While it is easy in Flamenco to use the SUPER tactic, by simply removing an item from the breadcrumb, users of mSpace have two options: they may simply identify and click on a different item, or they may reorder the columns so that a selection is placed higher up the temporary hierarchy (which we call the slice). The former of these two is not achievable in Flamenco, as alternatives of a selection are hidden and the exact selection is only displayed in the breadcrumb. The RELATE and NEIGHBOR tactics are also poorly supported in Flamenco due to the aforementioned four step process to change a selection. REARRANGE is also well supported by mSpace due to the ease in reordering facets. Finally, tactics like RESPELL are well supported by mSpace because changes to misspellings and unrecognised words in the keyword search are suggested and can be applied by a single click.

Finally, SCAFFOLD and TRACE are both less-well supported by RB++ as the facet columns are used for two purposes: making facet selections and, once Target Objects have been listed, filtering Target Objects. The selections made before Target Objects are listed are hidden. It is a unique feature that this separation exists, as making facet selections are by nature filtering the Target Object list and most browsers merge these conditions.

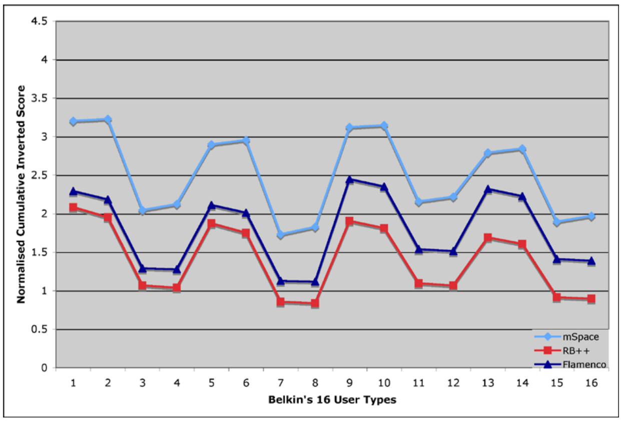

FIGURE 3.10: Graph G3 showing the support provided for each type of Belkin's users by each faceted browser, where peaks represent stronger support for a user type.

Predictably, as was shown in Graph G2 (Figure 3.9), there are also three distinct lines in Graph G3 (Figure 3.10), showing that mSpace provides the widest support for search. Quite clearly the graphs rise and fall in alternating pairs. This represents the alternation between recognise and specify (Mode) and is perhaps a predictable outcome for faceted browsers. By including more lessons learned from the information seeking work on keyword search, such as relevance feedback, we might see a balance between these two conditions. Within each of these alternating pairs, the mSpace bars marginally increases where the others fall. This indicates an increased support for meta-information (Resource).

Considering individual browser patterns, while RB++ and Flamenco follow a similar pattern for the first 8 ISS conditions, Flamenco notably improves on this gap in the final 8 conditions. These two halves are made unique by the Method dimension and indicates that Flamenco provides better support for search, which is defined by having a known Target Object to exist: this might be knowing that an academic paper exists and just trying to find it. This significant increase, also sharper than mSpace, may be present due to the better support for making further selections and the lower support for changing selections.

The final pattern we draw from Graph G3 (Figure 3.10) is shown every four conditions and is controlled by Belkin's Goal dimension. The Learn aspect of this dimension is shown by height differences between ISS1-4 and ISS5-8, and again between ISS9-12 and ISS13-16. This is characterised by the ability to see options in faceted browsers. The persistence of these options shown throughout to the user of mSpace is highlighted by the exaggerated difference in the first and third troughs compared to the second and fourth.

3.3 Expected Use in the Design Process

The framework presented above is designed to assess interfaces in a quick but structured way to identify both strengths and weaknesses in designs. The above three examples show how this process has value in three different expected areas of use. The first keyword example shows us how we can individually assess the ways in which a particular idea might support users. This can help us to better understand existing features, like keyword search, or analyse a new idea in terms of how it might help users.

The second Backward Highlighting example represents the way that we expect this tool will be mostly used: in understanding different designs for a new interface feature. The framework could be effectively used to identify issues and understand prototypes early in the design phase before human and financial resources are given to expensive studies with users. Later in this report, we show an example of how the framework helps to explain why a very large user study, by Capra et al. (2007), was unable to accept their hypotheses. If the framework had been applied beforehand, the study could have been designed differently and perhaps produced some evidence to support their hypotheses.

Finally, the third way that we expect the framework to be used is in situations where a user study may be difficult to carry out, such as the the faceted browser comparison above. Such occasions are described in the following section on the strengths and weaknesses of the framework.

In which ever way the framework is used, it may be foolish to rely on the proposed framework alone when designing software, as the assessment is based in theoretic models. Instead, and in line with research that suggests that multiple evaluation techniques should be used together over time (Shneiderman and Plaisant, 2006), we recommend that the framework be used in to inform better design and then to support the involvement of other techniques such as user studies.

3.4 Additional Strengths and Current Limitations

Aside from the analyses the framework provides, one of the main strengths of the framework, is that it can easily be applied to interfaces stored on machines that are not controlled by the investigator and stored anywhere in the world. In the faceted browser example above, the interfaces are stored at Southampton, UK, Berkeley and Chapel Hill, NC. To compare these interfaces in a fair user study, like the one carried out by Capra et al. (2007), each interface would have to present the same, or similar and equally structured content so that the same types of tasks could be given to users, and any instrumentation would have to be applied to each design. To arrange these resources for the study, the source code would have be provided to the investigators by each institution or each institution would have to collaborate and coordinate to conform to the same specifications. Either approach involves a lot of hard work and by many people. In the example above, however, three faceted browsers showing three distinct datasets, are compared with no changes or efforts required by any researchers other than the investigators, who simply carried out the analysis. This strength, as noted by Wilson et al. (2008), could provide the means to revive the interactive streams of the TREC conferences (Harman, 1997) that have stopped because of such barriers.

Another strength of the framework is that it can identify the aspects of designs that make one interface better than another. Where user studies may show that one interface might be better than another interface, unless only one feature is different between the two designs, the results usually struggle to show where or explain exactly why there is a significant difference. One approach, again taken by Capra et al. (2007) is to use qualitative discussions to investigate the cause of results. With the metrics and summaries produced by the framework, it is easy to identify where each browser is particularly strong and which features are causing any differences.

Finally, one of the most significant advantages over this form of analytical evaluation, provided by the framework, is that it can be done in very little time. Aside from the not requiring the preparation and organisation of multiple institutions, the study does not require any users. Our own experience with user studies, confirmed by communications with Capra et al. (2007), indicates that comparisons of complex interfaces, like the ones discussed above, can take around 40 hours with participants, not including preparation and analysis of results. The application of the framework to three interfaces takes around two days, in total, to both apply the framework and analyse the results. Analysing the single Backward Highlighting feature takes only a few hours.

The framework has two areas that could pose as limitations or cause inaccuracies in its analyses. First of all the accuracy and validity of the framework is somewhat influenced by the accuracy and validity of the two models that make up the two core building blocks. Second, Graph G3 depends heavily on the novel mapping created between the two models. This mapping has been created from careful research and educated interpretation, but as it was not produced from any empirical evidence, such as user studies, it needs to be validated. Both of these issues are addressed in the following Validation chapter, and are backed up with a validation of the whole approach against previously carried out user studies.

Chapter 4

Validating the Proposed Framework

Although the framework has previously shown promising results (Wilson and schraefel, 2007; Wilson et al., 2008), they must be validated before the approach can be used as a basis for decision or for influencing changes in design. Above we identified two key areas of the framework that need validating. First, the validity of the models used by the framework is addressed by considering related work in Section 4.1. The second area to validate is the novel integration of the chosen models, which, given the lack of empirical evidence, is validated by consensus of expert and novice opinion in Section 4.2. Finally, to provide empirical evidence to the accuracy of the overall framework, it is validated against user studies in Section 4.3.

4.1 Validation of the Original Models

One of the first and most important steps in validating the framework is to be confident in the models chosen to produce the analysis. As both the Bates and the Belkin models are the building blocks of the framework, the accuracy of the framework will be somewhat influenced by the accuracy of the models.

Aside from reusing her own model of search moves and tactics for many years (Bates, 1990), many other studies have shown the accuracy and thoroughness of Bates identified *tactics* by analyzing the actions of searchers. Before using Bates model in his own research (Hsieh-Yee, 1998), Hsieh-Yee identified a further 6 studies that used Bates' *tactics* and *moves* to explain the search behavior of participants (Hsieh-Yee, 1993; McClure and Hernon, 1983; Moody, 1991; Shute and Smith, 1993; Wildemuth et al., 1991, 1992) and 2 occasions where the model has been used to design a new search system (Buckland, 1992; Smith et al., 1989). Here, Bates *tactics* and *moves* are used together in a novel way to create a metric within the framework. Her work has clearly made a significant and strong contribution to Information Seeking research and we accept the model as having received strong validation.

Nicholas Belkin has been cited as one of the more prominent researchers in the field (White and McCain, 1998) and the model has been cited by many research papers. There has been very little direct validation, however, of the model and its accuracy. In the same paper that proposed the four dimensions (Belkin et al., 1993) and in a follow-up paper (Belkin et al., 1995), Belkin and his colleagues used the model to build two separate systems that support various types of users, but no empirical studies were performed. In response to research by Pharo (1999) that suggested that the model may be insufficiently exhaustive for some conditions, however, Cool and Belkin (2002) produced an extended version that goes into much more detail. This extension was then validated by Huvila and Widen-Wulff (2006) by applying the extended model to multiple scenarios.

Although the original model being used in the framework has not been directly validated, it provided the core understanding for an elaborated version of the model that was later given a strong validation. Further, the extended model reuses the dimensions of the first version, but goes into more detail than is necessary for the framework below. Personal communications with Nicholas Belkin have also suggested that not all of the dimensions will be applicable to the design of the framework and thus the appropriate use of the model would have to be carefully researched. Consequently, we accept that the core value of the research is strong and that the indirect validation is sufficient for using the earlier model at this stage of development. The extended model should, however, be considered for future work, which forms part of the work packages described in Chapter 5. Including the extended version, however, will not be trivial as it produces over 100 types of user in comparison with the current 16 less-finite types of user. Further the relevant aspects of the extended model will have to be carefully selected and the new application will have to be re-validated to check that it produces more accurate results.

Based on this assessment of the core building block models of the framework, which are often cited as key research in reviews of Information Seeking research, we can be confident in the contribution to the work. The following section validates the novel mapping that integrates the two models, which represents the main contribution and novel half of the framework development.

4.2 Validation of Novel Mapping between Models

The novel integration of Bates and Belkins models described in Section 3.1.2 is constructed of a non-trivial mapping, as each *tactic* (Bates) cannot be obviously or clearly connected with any specific value of Belkins dimensions. Further, as well as being difficult to state that a *tactic* x is associated with Dimension A, we cannot easily calculate the amount that Dimension A is supported. Consequently, it is important that the chosen mapping be validated so that the margin for error in the non-trivial integration is reduced. As there is no fixed process or metric to produce the mapping, it can only be discussed with and supported by other researchers. For this validation process, 3 search experts and 3 un-schooled researchers, with little or no knowledge of information seeking, have so far been involved in assessing the existing mapping. The method, results, and discussion of this validation are discussed below.

4.2.1 Method

To formalize the mapping assessment, rather than simply performing structured discussions, an analysis method was designed to: clearly present the models, collect mapping suggestions, and automatically produce a refined mapping. The process of formalizing a correlated mapping between the two models is one of the contributions of the second period of research, as it can be reused for the remaining period of research as more experts are able to volunteer approximately two hours of their valuable time.

To collect mapping suggestions from participants, an online form, shown in Figure 4.1, was generated that clearly presented each of Bates *tactics*, one at a time, along with a clear description from the original publications. Below the *tactics* was a persistently available description of each of Belkins dimension values. For every *tactic* (Bates), the participant was asked to select a dimension value (Belkin) that it most, second-most, and third-most supported. The decisions for each *tactic* were stored in a database and added to a spreadsheet when the participant had completed the full set of *tactics*.

To process the assessments provided by the participants, the number of times each dimension value was selected for each *tactic* was counted and the most popular choices were highlighted in a spreadsheet. This spreadsheet analysis provides three types of information. First, it identifies parts of the mapping that are unanimous across all participants, including the original mapping and novice opinion. Any such decisions can be quickly accepted so as to reduce the discussion time required by the already generous participants. Second, the process identifies parts of the mapping that were in close competition, so that they could be discussed. Preference, in this second case, was given to the opinion of experts, especially if they were in agreement. Third, the process identified parts of the mapping that varied widely and required further investigation. The results of this validation are discussed below.

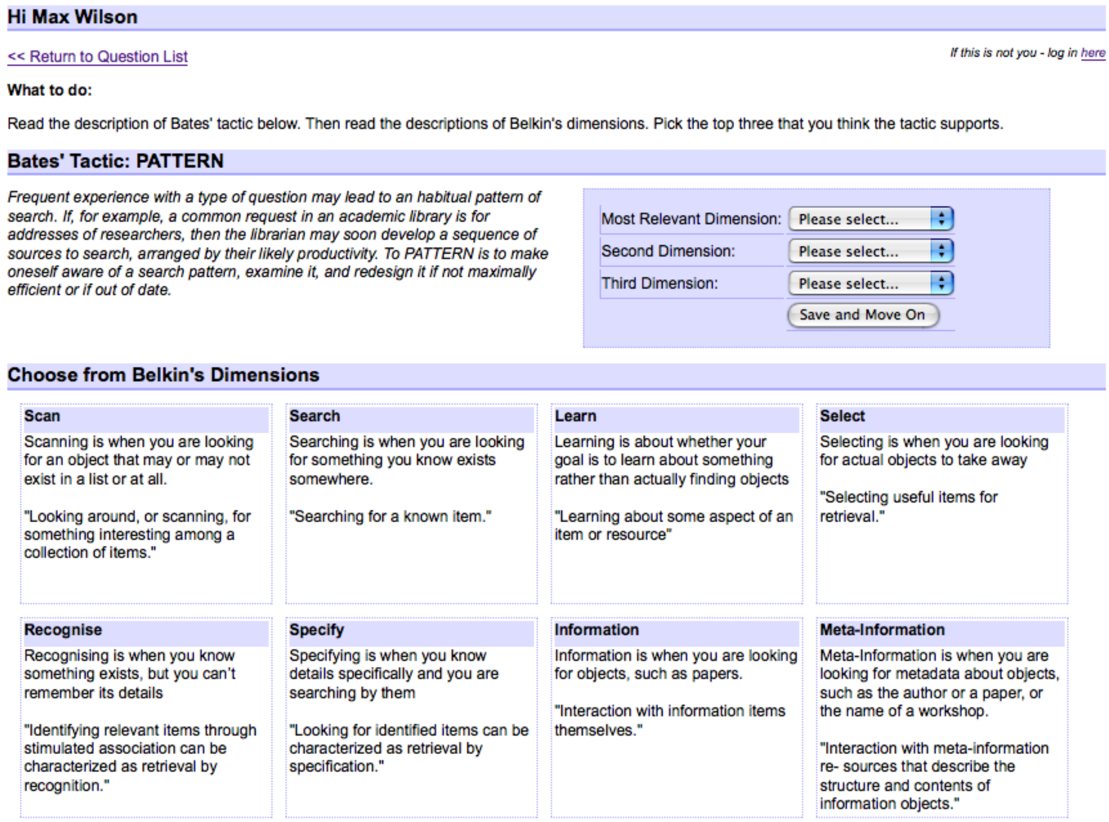

FIGURE 4.1: Online form used to collect expert and novice opinion about mappings between the Bates and Belkin models. The *tactics* are displayed one at a time, along with a definition, and the range of Belkin's dimension values are persistently available for reference. The participant can then select their top three choices of dimension value for each *tactic*.

4.2.2 Results

The process of producing a revised validated mapping was successful, but in evidence that producing a mapping between the two models is non-trivial, only 34% of the tactics could be immediately agreed upon without need for discussion or investigation. The rest of the tactics, as planned, were investigated by either assessing the difference in expert and novice opinion, or by revisiting literature to inform discussion. The distribution of agreement between participants is shown in Table 4.2.2. 38% of the *tactics* received a high correlation, and the decision was taken on the side of the experts in all but one case, where the second highest correlation of the experts, for the *tactic* matched the original mapping and the highest correlated *tactic* of the novices. Almost a third of the tactics had to be carefully researched and discussed. In the worst case, the suggested mapping of one *tactic* was different for almost every participant.

The correlation between the old and new mappings (the new mapping is shown in Table 4.2), is only around 60%, showing that the validation process was extremely important for the validity of the overall framework. The difference that this mapping makes on the previous analysis of comparing mSpace, RB++, and Flamenco, is discussed

Unanimous Decision	High Correlation	Split Decision	Varied Opinion
34.38%	37.50%	18.75%	9.38%

Table 4.1: Table showing the range of agreement and disagreement for a refined mapping between the Bates and Belkin models

below. The validation of the whole framework, using this refined mapping, is then shown to be more accurate when analysing the results of a user study of Backward Highlighting. The new mapping is then used to show how the results of the study of faceted browsers by Capra et al. (2007) could have been predicted.

4.2.3 Discussion

The process of validating the integration of the two models has provided opportunity to produce and refine new mappings. In particular, these mappings affect the information conveyed by Graph G3, as it controls the way that the information from Graph G2 is summarized for each user type. In Figure 4.2 we see a revised version of Graph G3 on the information reported by Wilson et al. (Wilson and schraefel, 2007; Wilson et al., 2008). As mentioned above, Graph G3 is read in patterns. In comparison with the previous graph (Figure 3.10) we can see three specific improvements in what Graph G3 tells us.

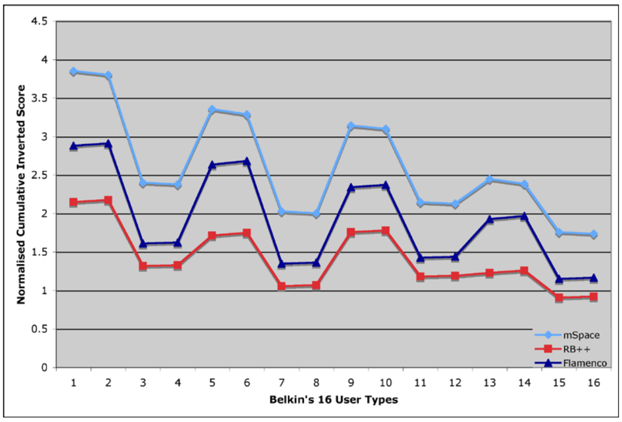

Figure 4.2: Graph G3 showing the support provided for each of Belkin's users by each faceted browser, using the refined mapping, where peaks represent stronger support for a user type.

First, instead of suggesting that Flamenco has enhanced support for Searching for known items over Scanning for items that may or may not exist, we see that the emphasis has moved to support users who will need to Recognize their results over being able to Specify. This pattern appears because the presence of facets allows users to recognize search terms rather than having to know them in advance to specify queries in a keyword search. Further, this notable improvement for Flamenco is inline with its facet optimization, where used facets are minimized to give more space to unused facets. This

ISS	Tactics
1	CHECK, WEIGH, RECORD, SURVEY, PARALLEL, SUPER, NEIGHBOUR, TRACE, BREACH, RESCUE
2	CHECK, WEIGH, SURVEY, STRETCH, PARALLEL, SUPER, NEIGHBOUR, TRACE, BREACH, RESCUE
3	CHECK, WEIGH, CORRECT, RECORD, BIBBLE, SURVEY, CUT, SCAFFOLD, SPECIFY, EXHAUST, REDUCE, PARALLEL, PINPOINT, BLOCK, SUPER, NEIGHBOUR, TRACE, REARRANGE, CONTRARY, RESPELL, RESPACE, BREACH, RESCUE, FOCUS
4	CHECK, WEIGH, CORRECT, BIBBLE, SURVEY, CUT, STRETCH, SCAFFOLD, SPECIFY, EXHAUST, REDUCE, PARALLEL, PINPOINT, BLOCK, SUPER, NEIGHBOUR, TRACE, REARRANGE, CONTRARY, RESPELL, RESPACE, BREACH, RESCUE, FOCUS
5	RECORD, SURVEY, PARALLEL, SUPER, TRACE, BREACH, RESCUE
6	SURVEY, STRETCH, PARALLEL, SUPER, TRACE, BREACH, RESCUE
7	CORRECT, RECORD, BIBBLE, SURVEY, CUT, SCAFFOLD, SPECIFY, EXHAUST, REDUCE, PARALLEL, PINPOINT, BLOCK, SUPER, TRACE, REARRANGE, CONTRARY, RESPELL, RESPACE, BREACH, RESCUE, FOCUS
8	CORRECT, BIBBLE, SURVEY, CUT, STRETCH, SCAFFOLD, SPECIFY, EXHAUST, REDUCE, PARALLEL, PINPOINT, BLOCK, SUPER, TRACE, REARRANGE, CONTRARY, RESPELL, RESPACE, BREACH, RESCUE, FOCUS
9	CHECK, WEIGH, PATTERN, RECORD, SELECT, CLEAVE, SUB, RELATE, NEIGHBOUR, VARY, FIX
10	CHECK, WEIGH, PATTERN, SELECT, STRETCH, CLEAVE, SUB, RELATE, NEIGHBOUR, VARY, FIX
11	CHECK, WEIGH, PATTERN, CORRECT, RECORD, BIBBLE, SELECT, CUT, SCAFFOLD, CLEAVE, SPECIFY, EXHAUST, REDUCE, PINPOINT, BLOCK, SUB, RELATE, NEIGHBOUR, VARY, FIX, REARRANGE, CONTRARY, RESPELL, RESPACE, FOCUS
12	CHECK, WEIGH, PATTERN, CORRECT, BIBBLE, SELECT, CUT, STRETCH, SCAFFOLD, CLEAVE, SPECIFY, EXHAUST, REDUCE, PINPOINT, BLOCK, SUB, RELATE, NEIGHBOUR, VARY, FIX, REARRANGE, CONTRARY, RESPELL, RESPACE, FOCUS
13	PATTERN, RECORD, SELECT, CLEAVE, SUB, RELATE, VARY, FIX
14	PATTERN, SELECT, STRETCH, CLEAVE, SUB, RELATE, VARY, FIX
15	PATTERN, CORRECT, RECORD, BIBBLE, SELECT, CUT, SCAFFOLD, CLEAVE, SPECIFY, EXHAUST, REDUCE, PINPOINT, BLOCK, SUB, RELATE, VARY, FIX, REARRANGE, CONTRARY, RESPELL, RESPACE, FOCUS
16	PATTERN, CORRECT, BIBBLE, SELECT, CUT, STRETCH, SCAFFOLD, CLEAVE, SPECIFY, EXHAUST, REDUCE, PINPOINT, BLOCK, SUB, RELATE, VARY, FIX, REARRANGE, CONTRARY, RESPELL, RESPACE, FOCUS

Table 4.2: Table showing Bates' *tactics* for each of Belkin's ISS conditions according to the revised mapping produced by the validation

reorganization means that more meta-data can be recognized. One of the reasons that mSpace is notably higher in the Specifying conditions (even eighths) is that it offers both Boolean keyword searches and interactive spelling suggestions, which were not present in the other browsers at the time of evaluation.

The second notable refinement is the missing rise in the RB++ browser for user types 13 and 14, who are Searching to Select by Recognizing. This difference is most likely to be because the other two browsers progressively filter results with each selection. RB++, however, requires users to explicitly ask for results after any number of selections. Consequently, users cannot recognize that their selections have found the right results as easily. The third notable refinement is that in mSpace, there is slightly better support for Information over Meta-Information, which can be attributed to the fact that, although each browser presents facets, only mSpace has a facet specifically for Information items.

Combined with the more expressive results in Graph G3, we can be confident in the refined mapping that has been produced in collaboration through consensus and discussion.

4.3 Validation of Whole Framework by Examples

To be confident that the refined framework ultimately produces accurate results, they must be formally checked against results that have been statistically proven, or not, by user studies. Two such studies are discussed below. The method used for both user study examples is the same. The designs or systems used in the studies are reviewed and entered into the framework. The three output graphs of the framework are used to show that the user study results could have been predicted. Further, we show how the study results can be explained and evaluated in more detail than the user studies. Finally, we discuss how the latter study could have been designed differently in a way that could have revealed the results that the initial hypotheses were seeking.

4.3.1 Backward Highlighting

In Section 3.2.2, the design of a tool to support directional column-faceted browsing was analysed by the framework, using the original mapping. Below, Graph G3 is revisited using the new refined mapping and shows that the new analysis matches the results of a user study performed by the designers, reported by Wilson et al. (2007). The new version of Graph G3, using the refined mapping, is shown in Figure 4.3.

First, the new mapping (for both design options) puts more emphasis on meta-information, which is important because the tool specifically highlights backwards up the facets that show meta-information. Further, this meta-information rise is sharper for times when the user is recognizing (users 2, 6, 10 and 14), which is important as a user who is knowledgeable enough to specify the items to select does not necessarily need the new technique other than to guide her eye. The original mapping (Figure 3.7) incorrectly indicated that backwards highlighting is in general better for users searching for a known item (right half of the graph) where as it actually well supports users who do not already know the relationships (scanning) even to recognize them. Further, the new mapping puts a more even balance on the odd and even quarters (learning and selecting), which

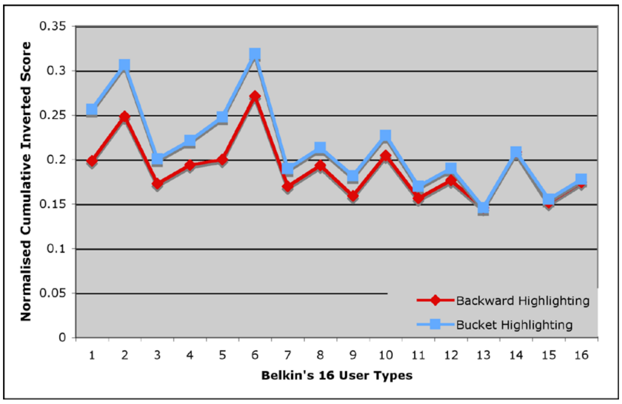

FIGURE 4.3: Graph G3 showing the support for each type of Belkins users by the two designs, using the new mapping, where peaks represent stronger support for a user type.

is better than the old mapping, as it acknowledges that users can simply learn about the data from the tool and also use the highlights to make selections. Arguably a user can more easily learn from the highlights, as it does not involve any further actions, and so the slight downward slope, from left to right, in the new mapping (Figure 4.3) may also be more accurate.

This revised analysis says that user type 6 is, in fact, the best supported, rather than type 14. User type 6 is one who is scanning to recognise and select meta-information. This almost exactly matches the definition of the tool, which is designed to highlight related meta-data in the columns of a directional faceted browser. It provides the least support, however, for user type 13, who is searching to recognize and select information, which makes sense as Backward Highlighting has little to do with the core information objects, which in turn makes them hard to recognise.

In terms of the two designs, the results of the user study indicated that there was very little difference between the two designs. We see that the two lines in Figure 4.3 do follow a very similar pattern. Statistical evidence was provided to show that slightly more about the meta-information could be learned with the Bucket Highlighting condition. This is shown in Figre 4.3 by the most significant gaps being on the left of the graph, where users are scanning and learning more often. The reduced gaps in the second and fourth quarters, compared to their first and third counterparts, supports the study results that participants did not necessarily make more selections in either condition. Finally, the overall increased support described by the graphs could have predicted that the users, overall, would have preferred the Bucket Highlighting technique; this preference was another finding in the user study.

4.3.2 3 Faceted Browser Study

Similar to the analysis of faceted browsers by Wilson et al. (2007; 2008) described above in Section 3.2.3, Capra et al. (2007) report on a user study of two faceted browsers and the original website of the source data. The original website was the Bureau of Labor Statistics, which presents a hierarchical classification on its homepage that categorizes US government reports. The website is compared to both the RB++ browser and an unconfigured or vanilla version of Endeca. Both browsers included faceted classifications over the same goal object: government reports on labor.

Two studies were carried out: one between participants and one within participants. The first study was designed to provide empirical results and the second to provide a qualitative to gain further insights. In both studies participants were asked to carry out three types of task: 1) a simple look up task where the answer could be found using just one facet; 2) a complex lookup task that required the use of multiple facets in conjunction; and 3) an exploratory task where participants were asked to learn about a given topic and report on the most interesting or important facts.

To provide context to the results of the studies, we first analyse the tasks given to the users to see what user types they become when carrying out the tasks. The types of task used in the study break up into two types of user according to Belkins dimensions. The two lookup tasks are both user type 13, where the user knows the answer lies within the systems (Searching) and their Goal is to select the answer to show they have completed to the task. As they do not know all the facts about the report, they cannot specify which report they want but can recognize reports that may contain the answer. Finally, the user is looking for an answer in the reports, rather than in the classification schemes, and so they are looking for Information.

User types 1 and 2 represent the exploratory task, as the facts that they find could either be produced from the meta-information in the classifications or the information in the reports. As there is no specific answer to the question, the participant will be scanning in order to learn more about the topic. Again they will only be able to recognize interesting reports as they see them. We now compare the results of these user studies with the analysis provided by the framework, presented in Graphs G1-G3 (Figure 4.4, Figure 4.5, and Figure 4.6).

The results of the study were not as expected, as no browser performed particularly well compared to the others. Even the original website performed equally well if not slightly better than the faceted browsers in the results. By applying the framework to the same three interfaces, we can see from Graph G3 (Figure 4.6) that the point where the three browsers perform most evenly is at user 13 - the user type that represents the simple and complex lookup tasks. Further, we can see that for the exploratory tasks (user types 1 or 2), the website even outperforms the RB++ browser. These findings could have

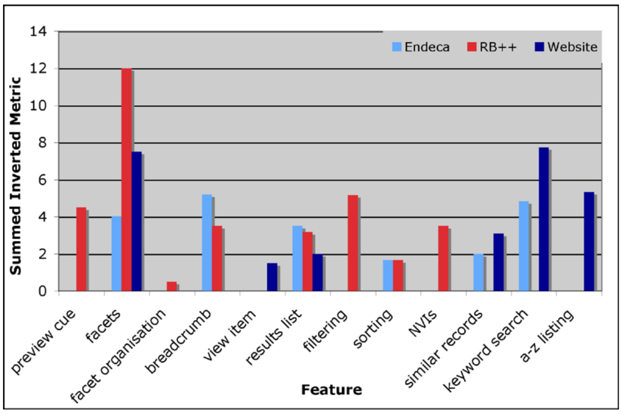

FIGURE 4.4: Graph G1 showing the support provided by each feature of the three interfaces in Capra et al. (2007), where taller bars represent stronger support provided by a feature.

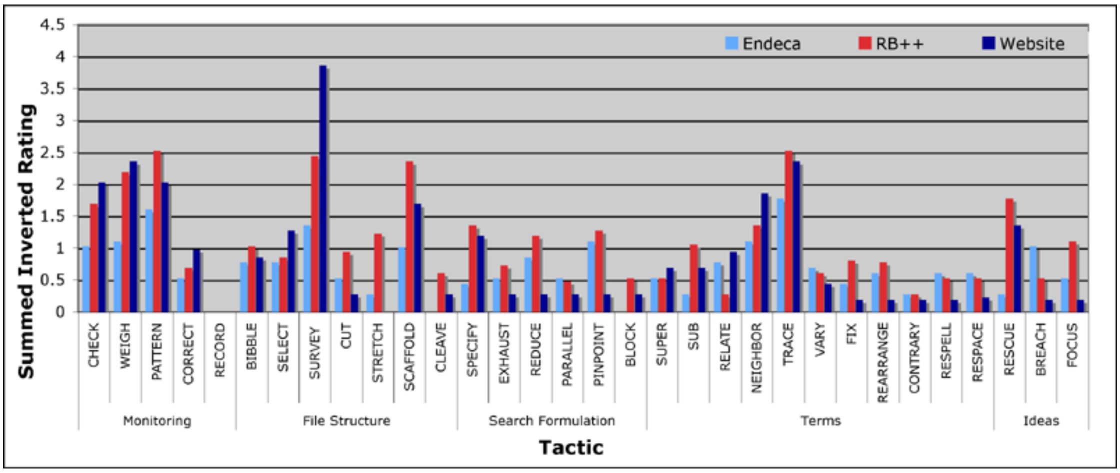

FIGURE 4.5: Graph G2 showing the support for each of Bates tactics provided by each interface in Capra et al. (2007), where taller bars represent stronger support for a *tactic*.

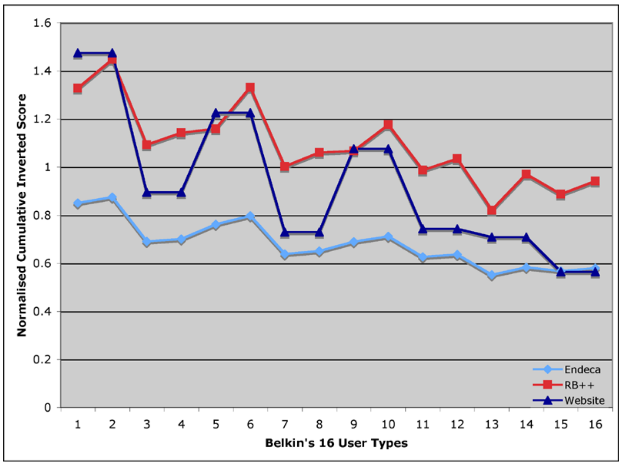

FIGURE 4.6: Graph G3 showing the support for each of Belkins user types provided by each interface in Capra et al. (2007), using the new mapping, where peaks represent stronger support for a user type.

predicted that the differences between the browsers was going to be marginal. Instead, the benefits of the RB++ may have been better shown if the users had been given tasks to find specific reports and had been given all the meta-data about the report to help. Such tasks would have represented user type 15. In general, the RB++ browser would appear to perform best for tasks that involved being able to specify the information they needed to find, as the even eighths of Graph G3 consistently show it provides stronger support.

As part of the qualitative analysis from the second study, participants were asked to label their most and least favorite aspects of the three browsers; summarized in Table 4.3.2. By referencing Graph G1 (Figure 4.4) we can see that the analysis by feature could have predicted these results too. According the results of the framework, the original website provided the strongest keyword search function (1); the RB++ browser does not provide keyword search at all (11). According to the graph, the second strongest feature of the website was the clearly presented facets (2), although it also shows that the facets in RB++ are more powerful (9). Of the three browsers, the website provided the least strong search results (3). The website was also the only browser not to provide some means of filtering or sorting the results (4). Although providing both facets and keyword search in Endeca, neither implementation was as strong as the other browsers (7). The RB++ Browser was the only browser to provide numbers to indicate how many documents were to be found given certain selections (10). RB++ provided numeric indicators in two forms, specific values (NVIs in Figure 4.4) and previews of affect before clicking (Preview cues in Figure 4.4). More about the value of numeric indicators is discussed by Wilson and schraefel (2006).

There are some results that cannot be so clearly revealed by Graph G1 (Figure 4.4). For example, our analysis shows that the results listings in Endeca were quite strong, which is in contrary to (8). One explanation could be that the feelings towards Endeca were quite neutral. A rating of how favorable the features were perceived was not reported, and so we cannot tell if this feature was specifically disliked. Another comment that was not predicted was that the participants did not like the structure of the facets in RB++. In the paper, Capra et al. suggest that the number of facets in the items were uneven. There is not a metric for this sort of aspect in the framework, but Hearst (2006) reports that the careful construction of facets is important in the design of faceted browsers. Consequently, this result is outside of the aims and design of the framework, but could have been predicted according to other related work.

A second unexplained result is the dislike of the RB++ facets, which could be in comparison to the clear representation on the original websites front page. We can see from Graph G2 (Figure 4.5) that the original website was particularly strong for tactics such as SURVEY, WEIGH, and CHECK. In particular, the clear layout of the classification on the front page of the website supports the ability to SURVEY a wide range of options. This less clear representation in RB++ faceted browser could explain its mention in the

Interface	P/C	Feature	From Feature Analysis (Figure 4.4)
Original Website	Pros	1. Keyword Search	The BLS website provides a very strong keyword search function, and RB++ does not at all.
		2. Clear Facets	The facets in RB++ are the most powerful in terms of functionality, but the clear layout of the facets in the BLS website make it stronger than the plain Endeca browser. The analysis by task type shows that the BLS website allows users to survey their options more clearly.
	Cons	3. Poor Search Results	Of the three interfaces, the BLS provided the least powerful search results listings.
		4. Manipulating Data	The website is the only browser not to provide sorting or filtering of results.
Plain Endeca	Pros	5. Useful Facets	The number of facets in the Endeca interface is more than the BLS website. This is not explicitly shown in Figure 4.4 though.
		6. Narrow Results	The increased number of facets makes it easier to narrow results.
	Cons	7. Limited Search	Although providing some aspect of both facets and keyword search, they both provide significantly less support to the user.
		8. Poor Search Results	
Relation Browser	Pros	9. Powerful Facets	The facets in RB++ provide the most powerful support for the users
		10. Numeric Values	The RB++ browser is the only browser to provide numbers that indicate the size of categories, and provide them in a preview form too.
	Cons	11. Limited Search	There is no keyword search in the RB++ browser
		12. Poorly built facets	

Table 4.3: List of identified pros and cons of the interfaces that could have been predicted by the 'by feature' analysis shown in Graph G1 (Figure 4.4)

least favorite features (12). Regardless of the potential explanations for this unpredicted result, we perceive it as a remaining challenge for the development of the model, which is discussed further in Chapter 5 in terms of assessing the simplicity or complexity of each interface.

Finally, another overall comment, that was not included in the most and least favorite features, was that, once selected, the participant had to leave the two faceted browsers to view an individual report. Users of the original website, however, can simply view the reports on the website. This disconnect is also shown in Graph G1 (Figure 4.4), where the support when viewing an item is only significant for the website.

4.4 A Refined Framework

The validation process above has played a vital part in the development and testing of the framework. First, the process has questioned and approved the strength of the models being used in the framework. Second, the novel mapping between the two models has been questioned and adjusted by multiple researchers to the extent that a revised mapping has been proposed based upon the consensus of a group. Third, this new mapping has been shown to more accurately correlate to the results of a user study that investigated a new faceted search feature called Backward Highlighting. Finally, comparing the revised and refined framework to a large user study of multiple browsers has shown just how accurate the framework's analyses can be.

Although this refined framework, using the mapping in Table 4.2, has shown to be accurate and insightful for analysing search interfaces, there are two areas that still require more research. First, we have identified that there is perhaps a revised version of Belkin's model, by Cool and Belkin (2002) that could be included. Second, the analysis of the study by Capra et al. (2007) shows that although the framework could have predicted most of the results, it does not capture an important aspect: simplicity versus complexity. Neither using the refined version of Belkin's model or including a measure of complexity will be a trivial tasks. The non-trivial nature and potential directions for achieving both in the final doctoral year are presented as extensions to the framework in Chapter 5 below.

Chapter 5

Future Extensions to the Proposed Framework

5.1 Summary of Open Questions from the Framework Validation

As well as informing changes in the framework, the validation process in Chapter 4 highlighted two areas that require further investigation: a measure of simplicity is required to complement the current measure of functionality and the extended model of users by Cool and Belkin (2002) needs investigating. Neither will be trivial to achieve and both are described in more detail below. The chapter concludes with a detailed plan of work packages to complete the framework, which is supported by the Gantt chart shown in Appendix A.

5.2 Interface Complexity

In the analysis of the study by Capra et al. (2007), the result that was not explained by the framework, but is within the scope of its aims, was that there was value for the participants in having a clear simple representation of facets. In opposition to the opinion of users, the framework suggests the less interactive version of facets used by the website are inferior to the powerful RB++ facets. Although the power of the RB++ facets was recognised by both the study and the framework, the clarity of the website facets was only valued by the study participants.

This highlights an assumption of the framework that more concurrent functionality is always best. Increased functionality, however, comes at a cost which is often labelled complexity. Tischler (2005) reports on the importance of simplicity in industrial design, citing examples like the Google front page, which, unlike its competitors, is add free and only promotes a small fraction of the real power behind its service. Similarly, the success of the iPod, for example, has been reported to lie in the trade off between functionality and simple design (Rojas, 2004).

The notion of simplicity is also present in academia. The Media Lab, at the Massachusetts Institute of Technology, has a Simplicity Consortium that is dedicated to the simplicity of design. Further, one of Jacob Nielsen's 10 design heurstics is to strive for a minimalist aesthetic and part of the Heuristic Evaluation created by Nielsen and Molich (1990).

As the current framework takes the assumption that more functionality is better, it would be beneficial to have a measure that is pulling in the opposite direction to reduce complexity. With both measures, an interface design can optimally include a good trade off of functionality and simplicity. Although this measure is desirable and simplicity is an important part of interface design, there is no simple clear metric available that can. Instead, three possible approaches to including a measure of simplicity into the framework are proposed below.

5.2.1 Involving Flow

In using the model of users, the subsequent work by Belkin et al. (1995) suggests that the flow of interaction with an interface is important. In their work, Belkin et al. create scripts of typical interaction patterns for each of the 16 user types. Further, as users change from novice to experienced, the typical patterns indicate how a user might flow from one user type to another. Belkin et al are not the only researchers to consider flow in information seeking. Kuhlthau (1991), for example, defines six stages that users typically follow when completing an information seeking task.

Instead of the framework suggesting that a user interface simultaneously support many types of users and search tasks, it may be possible to evaluate how a user is supported at stages of interaction. With knowledge of how users typically progress in their seeking tasks, the framework could assess the features that are needed at different stages and show when a feature could be hidden to make the interface less complicated. While lots of guidelines for designing interfaces suggest that consistency is important in design (Shneiderman and Plaisant, 2005; Rubinstein and Hersh, 1986), Grudin (1989) show that there are times when consistency may be at the detriment to good user interface design.

5.2.2 Formalising Engagement

Another stream of research is trying to identify and measure what causes users to engage with an interface and then what causes them to disengage (O'Brien and Toms, 2005, 2007). Personal communications with the authors suggests that in their forthcoming article, O'Brien and Toms (2008) present a version of their framework that includes 31 questions that together produce a metric for measuring engagement with an interface.

For the framework, such a measure could be used to discover when a strong combination of features has an opposite to desired effect and begins to cause a user to disengage with the interface. A good design, therefore, will provide strong features in a way that does not cause disengagement. This may provide a strong opportunity to produce opposing measures in the framework that allows designers to produce the best tradeoff of complexity and support. Plans have already been made to discuss potential collaborations.

5.2.3 Cognitive Load

Cognitive Load Theory, or CLT, is a stream of research that focusses on the user process of using new information in combination with information in memory. In basic terms, users have a limited amount of short term memory for achieving their goals, and when the problem or the interface complexity increases, it requires the user to use some long term memory to continue. When the load of short term memory and the cost of regularly accessing long term memory increases, the user reaches cognitive overload and the task becomes harder to achieve. A busy or complicated interface might use a lot of short term memory to process what has to be done to achieve their *tactics* with the interface, pushing the aspects of the task to long term memory. Consequently, reducing the effect of an interface on short term memory may improve the interface.

Cognitive Load Theory may also provide a possible opportunity to discover a measure of complexity for interfaces. Unlike formalising engagement, this area does not have an immediately obvious metric, but it may be easy to map the number of *moves* required to achieve a task and the number of features concurrently available in an interface to some level of cognitive load. More investigation into this psychological area of research is required to understand the full opportunity of its use.

Each of the three areas, and any others that may be discovered in the process, will be investigated as part of the future work into producing a measure of complexity to complement the current bias of the framework.

5.3 Finer Grained User Types

One of the main issues identified during the validation of the models being used in the framework is that the model of different user types by Belkin et al. (1993) was extended by Cool and Belkin (2002) after Pharo (1999) suggested that it was not exhaustive enough. The new version received significant validation by Huvila and Widen-Wulff (2006) where the original model did not. However communications with Nicholas Belkin have suggested that all of the extensions may not be applicable to the design of the framework, as they are factors of the interface rather than of the users.

It will be important to properly consider a transfer from the old model to the revised model, and this alone will take some considerable work. First, the aspects of the new design will have to each be considered for involvement in the design. When this is complete, the number of different user types will be known. Aside from the new model being much bigger, identifying many more types of users, and not being entirely applicable, including the new model will require producing and validating a new mapping. So far producing the current mapping has been part of two years work, some of which, however, can be taken as learning and experience.

5.4 Further Validation

Naturally, any changes or additions to the model, in terms of measuring complexity or changing one of the core models in the framework, will have to be re-validated to make sure the new version of the framework produces the same or better results. One of the contributions of the work so far, however, has been to produce reusable processes and measures than can be used to quickly evaluate the differences caused by making changes to the model. In simple terms, the process can be applied to the same user studies to see how the changes increase or decrease the accuracy of the framework.

5.5 Detailed Plan

The two key areas of further work have been identified by the research carried out so far. First, and possibly most importantly, there is clearly a need to discover when simplicity in a design provides benefits. Work packages 1, 2, and 3 will look at achieving this goal that may have a substantial affect on the design of the evaluation framework. The second key area of work will be to upgrade the existing model of user types by Belkin et al. (1993) to the more recent version by Cool and Belkin (2002). The exact extend and cost of this activity is not fully known, but will be researched carefully in Work packages 4, 5, and 6. Both of these activities should provide opportunities for publications, and it is planned that the outcome of both of these activities will produce a second article for the Journal for the American Society for Information Science and Technology, that will cover the validation and extension phases of finishing the framework, to match the first paper that describes the framework in great detail (Wilson et al., 2008). These publications will add to existing set of published work described in Appendix C and included at the end of the report.

To support the work packages, and the expected time it will take to complete them, a Gantt chart has been produced and included in Appendix A. More detail about the Gantt chart is included in the appendix. The chart is designed, however, to show more explicitly how the packages will be organised so that dependencies and parallel activities are supported.

5.5.1 WP1: Identify Options for Measuring Complexity

The main challenge of this work package will be to assess each of the known research areas that might be able to produce a measure of complexity, for their potential in contributing to the model. Like when originally designing the framework described above, the process may well require models to be novelly repurposed to provide a suitable metric. Further, and even less-trivially, the new metric will have to be mapped to the existing framework so that the two measures can be correlated in a simple but effective way. This may be very challenging as they will be inherently measuring different aspects of the same browsers.

Method (12 weeks total)

Research each possibility in detail (3 weeks each), such as the concepts of engagement and cognitive load theory, and design a way that they can be repurposed and included into the framework (1 week each).

5.5.2 WP2: Validation of each Option

Once the options for adding a metric for complexity to the interface have been created, each will have to be tried and tested to see how it improves or potentially impedes the accuracy of the framework. This process, as we have seen previously in the validation of the novel mapping, is very important. Although this process could be seen as simply and quickly applying the ideas, using the reusable methods produced during the validation phase of the research, each measure will be integrated differently and so separate integration methods will have to be created for each one.

Method (3 weeks total)

Design a method for each potential measure to be included in the reusable processes and apply each measure to a the two user studies discussed above to see their affect. The affect must then be assessed for its contribution or improvement before a recommended option can be selected.

5.5.3 WP3: Produce Refined Framework

Once the design of the new metric has been selected, the remaining challenge is to produce a refined framework that can be easily applied. In particular, this process involves creating clear instructions for use, simple input forms for the spreadsheet, and automatically generated analyses, all of which can be used in general situations like the current form of the framework. One of the most challenging parts of the process will be to select the most appropriate graphical representations for the newly integrated feature to clearly reveal where the functionality of a design is at the cost of simplicity or visa versa, so that the most optimal design can be easily selected.

Method (2 weeks total)

Make adjustments to the framework spreadsheet template file, create new input tables, produce refined instructions for applying the framework, and construct the most appropriate graphical representations for the results.

Intermediate Outcome

The product of this investigation will be a secondary and complementary measure for the framework that will permit the evaluation of search interfaces in a new way. This product provides an opportunity for a paper to be submitted to the International Conference on Human Factors in Computing Systems (CHI) in 2009. CHI is a high-level ACM conference that is a well respected and hard to get into, and will make a respectable venue for the research. The submission date will be around September in 2008, which matches nicely with the estimated time for carrying out the work, as shown by the Gantt chart in Appendix A.

5.5.4 WP4: More Detailed Model of User Types

As mentioned above, the integration of the revised model of user conditions, produced by Cool and Belkin (2002), will not be a simple switch of models. First, the new model will have to be assessed to identifiy which dimensions are applicable. Second, the structure of the new model involves variable length hierarchies within a dimension rather than simply 2 values. The identification of the new set of users will therefore be less straight forward. Third, when the appropriate parts are chosen, the currently validated mapping to Bates model will heave to be reformed to match the new parts of the framework. Finally, like before, this process will be best confirmed by the opinion of other experts.

Method (8 weeks total)

Tasks

- Research the design and validation of the extended model (2 weeks)
- Choose the appropriate parts to include in the framework (1 week)
- Produce a new mapping between the model by Bates (1979b) and by Cool and Belkin (2002) (3 weeks)
- Gather and analyse validation of this revised mapping (the more the better - at least 2 weeks)

As the last item requires the generous time of other researchers, the more time available for such an activity the better, subsequently, I plan to carry out Work Package 4 as soon as possible, so that there is plenty of time for others to participate.

5.5.5 WP5: Validate Model Change

The revised and validated mapping to the model by Cool and Belkin (2002) will need to be validated against the user studies used before. As before, any changes have to show a marked improvement on the accuracy of the results before they can be accepted as part of the framework.

Method (on-going - minimum 1 week)

This process can be carried out as input arrives from volunteering researchers. However there is a minimum time needed to produce a final analysis which must be carried out before the final framework is constructed.

5.5.6 WP6: Produce Final Framework

As with work package 3, there must be a process of converting the changes into a form that can be easily used and applied by other researchers and in real use situations.

Method (2 weeks total)

Make adjustments to the framework spreadsheet template file, create new input tables, produce refined instructions for applying the framework, and construct the most appropriate graphical representations for the results.

Intermediate Outcome As based in the non-trivial integration of theoretical Information Retrieval models, the International Conference on Research and Development on Information Retrieval (SIGIR) in 2009. This paper will provide a strong follow-up paper to the work that has been submitted to SIGIR2008 on the Validation in Section 4. Like CHI, SIGIR is a high-level ACM conference and will make a respectable venue for the work. The deadline for SIGIR2009 will be around February 2009, and well matches the estimated completion time according to the work packages and the Gantt chart in Appendix A.

5.5.7 WP7: Write Thesis

Time must be allotted to writing the final thesis report, so that other researchers can understand, apply, and potentially recreate the research and be published within the field of Information Seeking. This is a key activity that must be involved within the planning of the final years work.

Method (10 weeks total)

The method of writing and completing the thesis will be streamlined to an agreed schedule, so that chapter drafts can be commented upon individually and ahead of the final submission deadline. Some of the written text will be supported by the content of this report, however, which will have already received comments. Consequently, this scheduled and staggered approach to writing the thesis will mainly, but not solely, focus on the content of the final year's work and the revisions necessary for the introducing and concluding chapters.

Tasks

- On going reading must continue to make sure the latest research in Information Seeking is included in the thesis (4 weeks spead over period)
- Writing the thesis (6 weeks)

Additional Outcome The product of the Validation above and the planned extensions to the work in the final year will make a strong follow-up submission to the Journal of the American Society for Information Science and Technology (JASIST). The work already accepted (with revisions) into JASIST, and included in Appendix C, is on the design of the framework. The follow-up article, which matches the reviewers comments requesting validation of the work in the first journal article, will cover the validation and completion of the framework. Like the CHI and SIGIR conferences, JASIST is very well respected. With no specific submission deadlines, the journal article will be constructed very easily from the content written for the overall thesis.

The time involved in planning, constructing, editing, and potentially restructuring the report to make sure the research is conveyed most effectively is important. Plenty of time should be assigned specifically to this task, but the work will be on-going throughout the final year.

Chapter 6

Conclusions

Above, this report has described, validated and proposed extensions to an evaluation framework that can be applied to search interfaces to assess how well they support different types of user, where different user types are defined by aspects such as their previous knowledge and goals. By analysing the different features of each interface and how they can be used to achieve goals, we can begin to see how well an overall interface supports users and where it could be altered to broaden the support provided. The early work, originally reported in Wilson (2007), was validated to show that the framework can be used to predict the results of user studies, and further the reasons for the results. We then identify future work in this area to complement the current measure of functionality by assessing the complexity of a design. Each of these are discussed in more detail below.

6.1 Proposed Framework

The framework described above has been designed to analyse search interfaces in three ways. First it analyses the features of search interfaces, such as keyword searches or browsing facets, to identify how they contribute to the overall support provided by an interface. This support is then broken down into how users are supported in achieving different types of search tactics. Finally, these tactics are mapped to the needs to different types of users, so that the interface can be assess according to each of their needs.

There are five contributions to the study of search interfaces made by the framework and the analyses it provides. First, the framework can assess interfaces independently of the datasets they present and without the need to have access to the code or user logs. Second, where comparative studies of complete interfaces may tell us which performs best for a given task, the three analyses of the framework allow researchers to identify the features of the designs that have provided any benefits. Third, where a user study might have a limited set of tasks and interfaces, the framework can assess any number of interfaces simultaneously from the points of view of many different user types. This third benefit can provide a much more complete view of an interface rather than a select view controlled by the user study conditions. Fourth, the framework can be applied in relatively little time compared to a user study. User studies, reported in the report above, have spent around 40 hours with users during a user study, with many more required for preparation and analysis. The application of the framework can take only a few hours per interface to apply and analyse the results. Finally, given the identification of the strengths of the interface and the support for different user types, the framework can be used to inform the design of user studies so that they are best designed to meet hypotheses.

The framework also provides two contributions to the nature of academic research into information seeking. First, the framework reuses two existing information seeking models in order to produce a metric that can be used in an evaluation. The contribution, therefore, is the approach used to produce a new measure for information seeking by building upon previous models. The second contribution is the novel mapping used to map features of the two models, this mapping has been made available as a single entity so that it can be investigated, revised or extended by other researchers.

There are many strengths provided by the evaluation framework, but as it is based in theory, which at best only partially models the continuum of human behaviour, the framework cannot be solely relied upon to understand systems. Instead, we have first suggested that it be used early in the design process to inform both the designs being tested and the way that they are evaluated. Second, the framework has been validated, summarised below and in detail in Section 4, so that it can be used in confidence.

6.2 Validation Results

The validation of the framework has been on three fronts: the models being combined, the method of combining them, and the overall results of the combination. First, in depth literature review has reviewed the validity of the models used as the building blocks of the framework. The model of tactics and strategies (Bates, 1979b,a) has received strong validation and has been used successfully many times. The model of user types (Belkin et al., 1993) has been used by the authors, but some limitations were found. An extended version has been developed and validated, but is not wholly applicable for this framework. Further work, more of which is described below, is required to finally decide on the inclusion of the extended version, but for now the original model provides a strong basis for the framework.

The second phase of the validation was to assess the novel mapping used to combine the two models used. The mapping was not produced by any metric that can be checked, but by careful analysis of related literature. Consequently, the mapping required validation by achieving consensus between multiple interpretation of the literature. A combination of expert and novice participants reviewed a pre-prepared set of related literature and provided their own mappings between the two models and the collective consensus was used as a revised mapping. The revised mapping matched only 60% of the original mapping, but only a third of the tactics were easily agreed upon by the participants and the rest required further discussion and investigation. Both provide evidence the process of producing the mapping is not trivial and that the validation was extremely important.

The final validation was to show that the framework can produce the same results as user studies, and further supported the new mapping, which was more accurate than the original mapping. Two example user studies were reported and their results compared to the analyses provided by the framework. Both showed that the majority of the results could have been identified by the framework and in much less time. The user studies also identified some areas of extension, which are described below.

6.3 Identified Future Work

Now that a framework has been designed, tested, and validated, the next step is to revise and extend the framework to resolve some of the limitations identified by the process so far. The future work is planned in two main areas. The first main task is to assess the integration of the extended model of user types provided by Cool and Belkin (2002). This process is not trivial, as if the research suggests that it should be integrated, a new mapping will have to be generated and validated by both consensus and user studies. This involves repeating many of the steps that have contributed to the work so far.

The second main task proposed for future work is to identify a model of complexity that can be used to complement the framework that currently aims to increase the functionality of search interfaces. A complementary model of complexity will allow designers to identify the best combination of functionality and simplicity. Three areas of research have been identified so far that could provide a measure of complexity: research into causes of disengagement with software, research into cognitive load theory, and research into the typical patterns of interaction with search software. Each will be investigated before a model is integrated into the framework and re-validated.

6.4 Final Remarks

The evaluation framework described in this report has been designed to provide a new type of analysis for search software to investigate how they might support different types of users and which features of the interface are providing the support. The analyses can be used, then, to consider ways that the interface can be altered to improve support for user types who were previously limited in their search. The framework takes a holistic view of a system, by systematically and thoroughly analysing the whole interface and viewing it simultaneously from many perspectives. The framework also breaks many barriers that may limit user studies, such as access or control over interfaces and datasets and analysing what has caused the differences in study results.

The examples of application have shown that the framework can produce insightful results about user interfaces and the validation process has shown that it is both accurate and quick to perform. Due to the theoretical nature of the framework, it should be used early in the design process to improve the designs and inform the subsequent user studies applied to it. The positive results seen so far show that there is value in pursuing this focus of research to strengthen and finalise the contributions it can make to the design of interactive search systems.

Appendix A

Gantt Chart for Completing the Doctoral Research

Figure A shows how the Work Packages detailed in Section 5.5 will be carried out over the final year. There are four things to note about the Gantt Chart. First that, although unconfirmed at this time, an internship at Microsoft Research in Cambridge, UK, is expected to begin in May 2008 for three months and consequently the Gantt Chart, and studentship, continues until May 2009. Second, as the validation in Work Package 5 requires external participation, Work Package 4 will be completed before the internship, so that the maximum amount of time is available to collect results. Third, there are two conference submission deadlines marked with superscript numbers matching the expected publications described in Section 5.5. Four, that both the validation (WP5) and the continued reading (part of WP 7) will continue during the internship while most other tasks, such as WP1 and WP2 may be put on hold. This is conveyed by the difference in colour saturation during the internship of the relevant work packages.

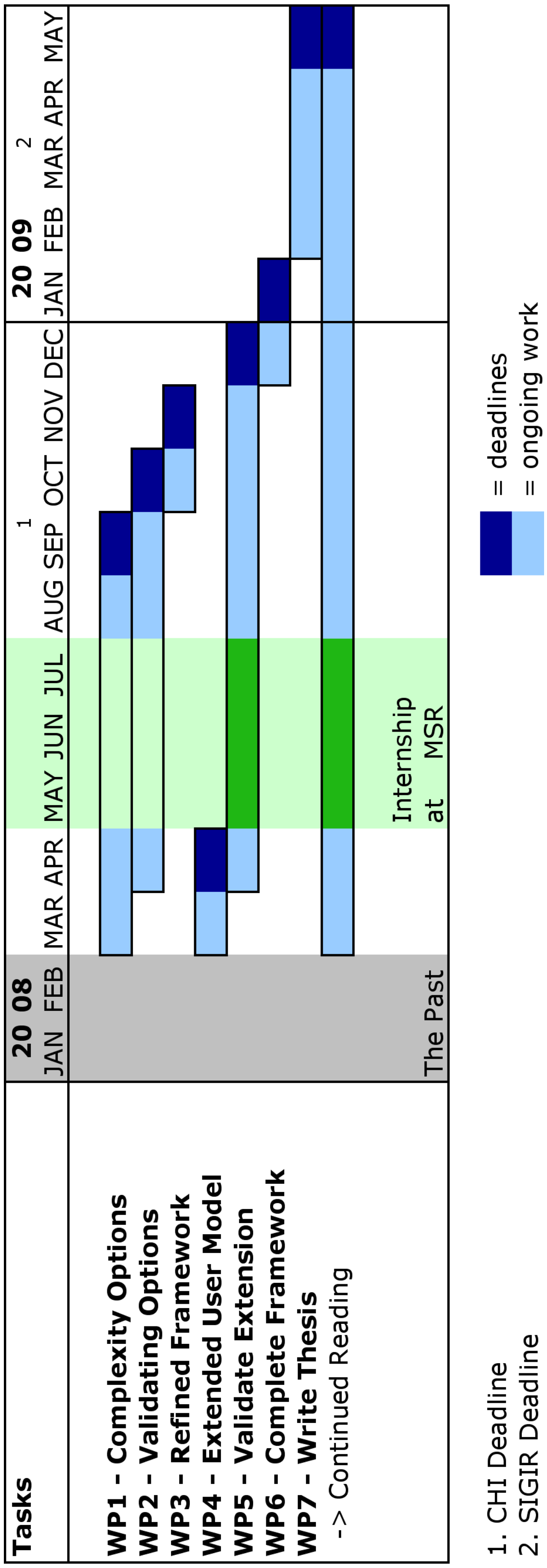

FIGURE A.1: Gantt chart showing how the work packages required to complete the framework are arranged in the final year, considering both dependencies and parallel activities.

Appendix B

Community Participation

To support my experience and knowledge development during the studentship, and with the support of my supervisor, I have been as active as possible in the different fields relating to my research, mainly: Human Computer Interaction and Information Retrieval. As part of this community involvement, I have, over the last year, been to the following conferences: ISWC2007, SIGIR2007, ESSIR07 (an information retrieval summer school), and UIST2007 (a top-tier conference on novel and early user interface designs). The latter of these, and with the financial support of the Royal Academy of Engineering, has also enabled visits to academic institutions, such as MIT, and industry partners such as Endeca and Nokia. The work described in this report was presented to each of the institutions visited. The visits have also spurred collaborative efforts.

Part of my research has led me to become closely involved with the Semantic Web User Interaction interest group (SWUI[1]), and as part of this participation I manage the Wiki, as well as attend events and review submissions. I have also reviewed for many of the conferences I have submitted to and attended, including the CHI, UIST, and the British Computer Society's HCI conference.

One of the more tangible returns for this contribution has been the interest shown in my work by Microsoft Research. Aside from the interest in my work shown by Ryen White, my participation at UIST2007 initiated discussions with other Microsoft Research employees and has, although unconfirmed at this time, is likely to lead to an internship. Notably, I was invited to apply for an internship after UIST2007.

In the final year, I have similar plans for community involvement. As well as continuing to review for conferences and support the SWUI group, I intend to become further involved with the University of Southampton's interest in Web Science, and attend conferences such as CHI2008 and JCDL2008. I have also submitted to SIGIR2008, but

[1]http://swui.webscience.org

given the distant location, attendance will be based largely on the acceptance of the paper.

Appendix C

Relevant Publications to the Research

Out of my full list of publications [1], currently totalling 28, with 2 articles, 12 conference and workshop papers, 11 technical reports, and 3 conference posters, I have listed 8 below, where I have been at the core of the work, and are cited in the report above. They have been ordered with the most relevant and significant first and each has a small description. The full texts can be found after the Bibliography.

1) **Wilson, M. L.**, schraefel, m. c. and White, R. W. (to appear) *Evaluating Advanced Search Interfaces using Established Information-Seeking Models.* In submission to the Journal of the American Society for Information Science and Technology . (Accepted with revisions due)

This article, accepted with revisions into JASIST, was the produce of the first year's work in designing and first applying the framework to advanced search interfaces. The paper was co-authored with Ryen White, from Microsoft Research, who has been closely involved with the Exploratory Search workshops and was keen to be involved in publishing the work. It is important to note that the work was purely produced from my research, but the publication was authored with the advice, comments, and edits provided by Ryen White.

2) **Wilson, M. L.** and schraefel, m. c. (2007) *Bridging the Gap: Using IR Models for Evaluating Exploratory Search Interfaces.* In: SIGCHI 2007 Workshop on Exploratory Search and HCI, 28th April 2007, San Jose, CA, USA.

This workshop paper was presented at the ACM CHI conference workshop on Exploratory Search to many experts in the Exploratory Search field, including Ryen White who subsequently co-authored the paper above.

[1] http://www.ecs.soton.ac.uk/people/mlw05r/

3) **Wilson, M. L.** and schraefel, m. c. (2008) *A Validated Framework for Measuring Interface Support for Interactive Information Seeking.* Submitted to the ACM Conference on Research and Design of Information Retrieval, Singapore.

Currently in submission to SIGIR2008 is a paper on the validation of the framework presented in Section 4. Notifications are in April 2008.

4) schraefel, m. c., **Wilson, M. L.**, Russell, A. and Smith, D. A. (2006) *mSpace: improving information access to multimedia domains with multimodal exploratory search.* Communication of the ACM, 49 (4). pp. 47-49.

At the start of my studentship research, I contributed (second-most to the lead author) to an article that featured in a special edition of the Communications of the ACM on Exploratory Search. The article focussed on the design of mSpace, as an exploratory search browser. mSpace has since provided the means to design, test and evaluate new search ideas and has subsequently led to publications, some of which are listed below.

5) Kules, W., **Wilson, M. L.**, schraefel, m. c. and Shneiderman, B. (to appear) *From Keyword Search to Exploration: How Result Visualization Aids Discovery on the Web.*

Although not included below, due to its size, my research and growing expertise has led to my involvement in a monograph this is still in development. It has been a privilege to work with experts, such as William Kules, Ben Shneiderman, and m.c. schraefel. The monograph is due to be included in the new Foundations and Trends in Web Science Journal later this year.

6) **Wilson, M. L.** and schraefel, m. c. (2008) *A Longitudinal Study of Exploratory and Keyword Search.* In: ACM/IEEE-CS Joint Conference on Digital Libaries, June 16-20, 2008, Pittsburgh, PA, USA.

This recent investigation into the traditional keyword and more recent exploratory forms of searching and knowledge building was accepted on the 5th of March 2008 for the JCDL08 conference later this year. This publication represents a contribution to another field at a jointly sponsored ACM and IEEE conference on digital libraries, as opposed to information retrieval and HCI design.

7) **Wilson, M. L.**, Smith, D. A., Russell, A. and schraefel, m. c. (2006) *mSpace Mobile: a UI Gestalt to Support On-the-Go Info-Interaction.* In: the SIGCHI Conference on Human Factors in Computing Systems (CHI06) extended abstracts, 22-27 April, 2006, Montreal, Canada.

This was one of my earlier publications that was on more ubiquitous contexts to searching for information and gaining knowledge on the move. The work is tertiary to the content of my studentship research, but the value of the slightly different approach contributes to the overall knowledge development during the studentship, and is important for fleshing out the boundaries to the main focus.

8) **Wilson, M. L.** and schraefel, m. c. (2006) *mSpace: What do Numbers and Totals Mean in a Flexible Semantic Browser.* In: The 3rd International Semantic Web User Interaction Workshop at ISWC2006, November, 2006, Athens, GA.

This paper, accepted into a user interaction workshop of yet another field of research, investigated the new opportunities for more exploratory forms of search provided by technological advances in other fields. As mentioned in my community participation (Appendix B), I continue to actively partake in this new field to be aware of the opportunities that are being revealed by developments in the Semantic Web.

Bibliography

M. J. Bates. Idea tactics. *Journal of the American Society for Information Science*, 30 (5):280–289, 1979a.

Marcia J. Bates. Information search tactics. *JASIS*, 30:205–4214, 1979b.

Marcia J. Bates. Where should the person stop and the information search interface start? *Inf. Process. Manage.*, 26(5):575–591, 1990. ISSN 0306-4573.

Marcia J. Bates. The cascade of interactions in the digital library interface. *Inf. Process. Manage.*, 38(3):381–400, 2002. ISSN 0306-4573.

Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, David Grossman, and Ophir Frieder. Hourly analysis of a very large topically categorized web query log. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 321–328, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-881-4.

N. J. Belkin, P. G. Marchetti, and C. Cool. Braque: design of an interface to support user interaction in information retrieval. *Inf. Process. Manage.*, 29(3):325–344, 1993. ISSN 0306-4573.

N.J. Belkin, C. Cool, A. Stein, and U. Thiel. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert Systems with Applications*, 9(3):379–395, 1995.

Spring Hull Bonnie Lida and Katie Pilcher. Breadcrumb navigation: An exploratory study of usage. *Usability News*, 5(1), 2003.

P. Borlund. The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3):8–3, 2003.

S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

Bertram C. Brookes. The foundations of information science: Part I. Philosophical aspects. *Journal of Information Science*, 2(3-4):125–133, 1980.

M. K. Buckland. Design for an adaptive library catalog. *Proceedings of the ASIS Mid-Year Meeting*, 165:71, 1992.

R. Capra, G. Marchionini, J.S. Oh, F. Stutzman, and Y. Zhang. Effects of structure and interaction style on distinct search tasks. In *Proceedings of the 2007 conference on Digital libraries*, pages 442–451. ACM Press New York, NY, USA, 2007.

Ed H. Chi, Peter Pirolli, Kim Chen, and James Pitkow. Using information scent to model user information needs and actions and the web. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 490–497, New York, NY, USA, 2001. ACM Press. ISBN 1-58113-327-8.

Colleen Cool and Nicholas J. Belkin. A classification of interactions with information. In *CoLIS 4: Proceedings of the 4th international conference on Conceptions of Library and Information Science*, pages 1–15, 2002. ISBN 1-59158-016-1.

B. Dervin. From the mind's eye of the user: The sense-making qualitative-quantitative methodology. *Qualitative research in information management*, pages 61–84, 1992.

Natalya Godbold. Beyond information seeking: towards a general model of information behaviour. *Information Research*, 11(4):paper 126, 2006.

Richard Gong and David Kieras. A validation of the goms model methodology in the development of a specialized, commercial software application. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 351–357, New York, NY, USA, 1994. ACM Press. ISBN 0-89791-650-6.

Wayne D. Gray, Bonnie E. John, and Michael E. Atwood. The precis of project ernestine or an overview of a validation of goms. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 307–312, New York, NY, USA, 1992. ACM Press. ISBN 0-89791-513-5.

Jonathan Grudin. The case against user interface consistency. *Communications of the ACM*, 32(10):1164–1173, 1989. ISSN 0001-0782.

M. Hammersley and P. Atkinson. *Ethnography: Principles in Practice*. Routledge, 1995.

D. K. Harman. The trec conferences. *Morgan Kaufmann Multimedia Information And Systems Series*, pages 247–256, 1997.

Marti A. Hearst. Next generation web search: Setting our sites. *IEEE Data Engineering Bulletin: Special Issue on Next Generation Web Search*, 2000.

Marti A. Hearst. Clustering versus faceted categories for information exploration. *Commun. ACM*, 49(4):59–61, 2006. ISSN 0001-0782.

K. Holtzblatt and H.R. Beyer. Requirements gathering: the human factor. *Communications of the ACM*, 38(5):31–32, 1995.

I. Hsieh-Yee. Search tactics of web users in searching for texts, graphics, known items and subjects: Asearch simulation study. *Electronic Resources: Use and User Behavior*, 1998.

Ingrid Hsieh-Yee. Effects of search experience and subject knowledge on online search behavior: Measuring the search tactics of novice and experienced searchers. *Journal of the American Society for Information Science*, 44(3):161–174, 1993.

I. Huvila and G. Widen-Wulff. Perspectives to the classification of information interactions: the cool and belkin faceted classification scheme under scrutiny. *Proceedings of the 1st international conference on Interaction in context*, pages 144–152, 2006.

Kalervo Järvelin and Peter Ingwersen. Information seeking research needs extension towards tasks and technology. *Information Research*, 10(1), 2004.

B.E. John. Extensions of GOMS analyses to expert performance requiring perception of dynamic visual and auditory information. *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, pages 107–116, 1990.

Bonnie E. John and Allen Newell. Predicting the time to recall computer command abbreviations. *SIGCHI Bull.*, 17(SI):33–40, 1987. ISSN 0736-6906.

Bonnie E. John, Paul S. Rosenbloom, and Allen Newell. A theory of stimulus-response compatibility applied to human-computer interaction. *SIGCHI Bull.*, 16(4):213–219, 1985. ISSN 0736-6906.

K.S. Jones et al. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

J. Koenemann and Nicholas J. Belkin. A case for interaction: a study of interactive information retrieval behavior and effectiveness. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 205–212, New York, NY, USA, 1996. ACM Press. ISBN 0-89791-777-4.

Sascha Kriewel. Finding and using strategies for search situations in digital libraries. *Bulletin of the IEEE Technical Committee on Digital Libraries (TCDL)*, 2(2), 2006.

Carol C. Kuhlthau. Inside the search process: Information seeking from the user's perspective. *Journal of the American Society for Information Science*, 42(5):361–371, 1991.

Gary Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46, 2006. ISSN 0001-0782.

Catherine C. Marshall and III Frank M. Shipman. Spatial hypertext and the practice of information triage. In *HYPERTEXT '97: Proceedings of the eighth ACM conference on Hypertext*, pages 124–133, New York, NY, USA, 1997. ACM Press. ISBN 0-89791-866-5.

C. R. McClure and P. Hernon. *Improving the quality of reference service for government publications*. American Library Association Chicago, 1983.

M. K. Moody. Documents search strategies and general reference search strategies: an analysis and comparison. *The Reference librarian*, (32):57–69, 1991.

Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, New York, NY, USA, 1990. ACM. ISBN 0-201-50932-6.

Heather L. O'Brien and Elaine G. Toms. What is user engagement? a conceptual framework for defining user engagement with technology. *Journal of the American Society for Information Science and Technology*, 59(6):1–18, April 2008.

H.L. O'Brien and E.G. Toms. Engagement as Process in Human-Computer Interactions. In *Annual Conference of of American Society for Information Science and Technology*, 2005.

H.L. O'Brien and E.G. Toms. Evaluating engagement in interactive search. In *SIGIR2007 Workshop on Web Information Seeking and Interaction*, 2007.

R.A. Peterson. *Constructing Effective Questionnaires*. Sage Publications, 2000.

N. Pharo. Web information search strategies: A model for classifying web interaction. In *Proceedings of the 3rd International Conference on the Conceptions of the Library and Information Science*, pages 207–218, 1999.

J. Piaget. The stages of the intellectual development of the child. *Bull Menninger Clin*, 26:120–8, 1962.

Peter Pirolli and Stuart Card. Information foraging in information access environments. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 51–58, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co. ISBN 0-201-84705-1.

V. Raghavan, P. Bollmann, and G.S. Jung. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Information Systems (TOIS)*, 7(3):205–229, 1989.

M. Rettig. Prototyping for tiny fingers. *Communications of the ACM*, 37(4):21–27, 1994.

SE Robertson. On term selection for query expansion. *Journal of Documentation*, 46 (4):359–364, 1991.

SE Robertson, S. Jones, et al. Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, 27(3):129–46, 1976.

SE Robertson, S. Walker, MM Beaulieu, N. Square, E.C.V. London, and M. Gatford. Okapi at TREC-4. DIANE Publishing, 1998.

Peter Rojas. Entelligence: Understanding the success of the ipod and ipod mini. Online, 2004.

R. Rubinstein and H. Hersh. *The Human Factor: Designing Computer System for People*. Digital Press, Burlington, Mass, 1986.

G. Salton and C. Buckley. Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, 41(4):288–97, 1990.

Tefko Saracevic. The stratified mode of information retrieval interactino: Extension and applications. In *Proceedings of the American Society for Information Science*, volume 34, pages 313–327, 1997.

m.c. schraefel, Max L. Wilson, and Maria Karam. Preview cues: Enhancing access to multimedia content. Technical report, School of Electronics and Computer Science, University of Southampton, August 2004. http://eprints.ecs.soton.ac.uk/9253/.

m.c. schraefel, Max L. Wilson, Alistair Russell, and Daniel A. Smith. mspace: improving information access to multimedia domains with multimodal exploratory search. *Communications of the ACM*, 49(4):47–49, 2006. ISSN 0001-0782.

m.c. schraefel, Yuxiang Zhu, David Modjeska, Daniel Wigdor, and Shengdong Zhao. Hunter gatherer: interaction support for the creation and management of within-web-page collections. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 172–181, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-449-5.

Ben Shneiderman, Benjamin B. Bederson, and Steven M. Drucker. Find that photo!: interface strategies to annotate, browse, and share. *Commun. ACM*, 49(4):69–71, 2006. ISSN 0001-0782.

Ben Shneiderman and Catherine Plaisant. *Designing the user interrface: Strategies for effective human-computer interaction (4th Ed.)*. Addison-Wesley, 2005.

Ben Shneiderman and Catherine Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors*, pages 1–7, New York, NY, USA, 2006. ACM. ISBN 1-59593-562-2.

S. J. Shute and P. J. Smith. Knowledge-based search tactics. *Information Processing and Management*, 29(1):29–45, 1993.

P. J. Smith, S. J. Shute, B. Galdes, and M. H. Chignell. Knowledge-based search tactics for an intelligent intermediary system. *ACM Transactions on Information Systems (TOIS)*, 7(3):246–270, 1989.

L.T. Su. Evaluation Measures for Interactive Information Retrieval. *Information Processing and Management*, 28(4):503–16, 1992.

Linda Tischler. The beauty of simplicity. *Fast Company*, (100):Online, November 2005.

C. Wharton, J. Rieman, C. Lewis, and P. Polson. The cognitive walkthrough method: a practitioner's guide. *Usability inspection methods table of contents*, pages 105–140, 1994.

H. D. White and K. W. McCain. Visualizing a discipline: An author co-citation analysis of information science, 1972-1995. *Journal of the American Society for Information Science*, 49(4):327–355, 1998.

Ryen W. White, Bill Kules, Steven M. Drucker, and m.c. schraefel. Introduction. *Commun. ACM*, 49(4):36–39, 2006. ISSN 0001-0782.

B. M. Wildemuth, R. de Bliek, S. He, and C. P. Friedman. Search moves made by novice end users. *Proceedings of the 55th annual meeting on Celebrating change: information management on the move: information management on the move table of contents*, pages 154–161, 1992.

B. M. Wildemuth, Jacob, Fullington, R. de Bliek, and C. P. Friedman. A detailed analysis of end-user search behaviors. *Proceedings of the ASIS Annual Meeting*, 28: 302–12, 1991.

Max L. Wilson. A nine month report on progress towards a framework for evaluating advanced search interfaces considering information retrieval and human computer interaction. Technical report, School of Electronics and Computer Science, University of Southampton, 2007.

Max L. Wilson, Paul Andre, Daniel A. Smith, and m.c. schraefel. Spatial consistency and contextual cues for incidental learning in browser design. Technical report, School of Electronics and Computer Science, University of Southampton, September 2007.

Max L. Wilson and m.c. schraefel. A longitudinal study of exploratory and keyword search. In *ACM/IEEE-CS Joint Conference on Digital Libraries*. IEEE Computer Society, June 2008.

Max L Wilson, m.c. schraefel, and Ryen W. White. Evaluating advanced search interfaces using established information-seeking models. *In submission to the Journal of the American Society for Information Science and Technology*, 2008.

Max L. Wilson, Alistair Russell, Daniel Smith, and m.c. schraefel. mspace mobile: Exploring support for mobile tasks. In *British Conference on Human-Computer Interaction 2006*, 2006.

Max L. Wilson and m.c. schraefel. mspace: What do numbers and totals mean in a flexible semantic browser. In *The 3rd International Semantic Web User Interaction Workshop*, 2006.

Max L. Wilson and m.c. schraefel. Bridging the gap: Using ir models for evaluating exploratory search interfaces. In *First Workshop on Exploratory Search and HCI at SIGCHI07*, 2007.

T.D. Wilson. Models of information behaviour research. *Journal of Documentation*, 55 (3):249–270, 1999.

Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-630-7.

Junliang Zhang and Gary Marchionini. Evaluation and evolution of a browse and search interface: relation browser. In *dg.o2005: Proceedings of the 2005 national conference on Digital government research*, pages 179–188. Digital Government Research Center, 2005.