UNIVERSITY OF SOUTHAMPTON

An Analytical Inspection Framework for Evaluating the Search Tactics and User Profiles Supported by Information Seeking Interfaces

by

Max L. Wilson

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the

Faculty of Engineering and Applied Science
Department of Electronics and Computer Science

November 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

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

Doctor of Philosophy

by Max L. Wilson

Searching is something we do everyday both in digital and physical environments. Whether we are searching for books in a library or information on the web, search is becoming increasingly important. For many years, however, the standard for search in software has been to provide a keyword search box that has, over time, been embellished with query suggestions, boolean operators, and interactive feedback. More recent research has focused on designing search interfaces that better support exploration and learning. Consequently, the aim of this research has been to develop a framework that can reveal to designers how well their search interfaces support different styles of searching behaviour.

The primary contribution of this research has been to develop a usability evaluation method, in the form of a lightweight analytical inspection framework, that can assess both search designs and fully implemented systems. The framework, called Sii, provides three types of analyses: 1) an analysis of the amount of support the different features of a design provide; 2) an analysis of the amount of support provided for 32 known search tactics; and 3) an analysis of the amount of support provided for 16 different searcher profiles, such as those who are finding, browsing, exploring, and learning. The design of the framework was validated by six independent judges, and the results were positively correlated against the results of empirical user studies. Further, early investigations showed that Sii has a learning curve that begins at around one and a half hours, and, when using identical analysis results, different evaluators produce similar design revisions.

For Search experts, building interfaces for their systems, Sii provides a Human-Computer Interaction evaluation method that addresses searcher needs rather than system optimisation. For Human-Computer Interaction experts, designing novel interfaces that provide search functions, Sii provides the opportunity to assess designs using the knowledge and theories generated by the Information Seeking community. While the research reported here is under controlled environments, future work is planned that will investigate the use of Sii by independent practitioners on their own projects.

Key Publications from this Work

Journal Articles

Wilson, M. L., schraefel, m. c., and White, R. W. (2009b). Evaluating advanced search interfaces using established information-seeking models. *Journal of the American Society for Information Science and Technology*, 60(7):14071422.

Wilson, M. L. and schraefel, m. c. (2009). Evaluating collaborative information seeking interfaces with a search-oriented inspection method and re-framed information seeking theory. *Information Processing & Management* (In Press).

Peer Reviewed Conference and Workshop Papers

Wilson, M. L., Andre, P., and schraefel, m. c. (2008). Backward highlighting: enhancing faceted search. In *UIST 08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 235238, New York, NY, USA. ACM.

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

Wilson, M. L. and schraefel, m. c. (2008a). Evaluating collaborative search interfaces with information seeking theory. In *1st International Collaborative Search Workshop*.

Wilson, M. L. and schraefel, m. c. (2008b). Improving exploratory search interfaces: Adding value or information overload? In *Proceedings of the Second Human Computer Interaction and Information Retrieval Workshop (HCIR08)*.

Wilson, M. L. and schraefel, m. c. (2009b). Sii: the lightweight analytical search interface inspector. In *JCDL09 Workshop on Lightweight User-Friendly Evaluation Methods for Digital Librarians*. DLib Magazine.

Contents

List of Figures

List of Tables

Acknowledgements

I'd like to thank m.c. schraefel for almost constantly pushing me in different directions. What sometimes felt like chaotic and inexplicable guidance consistently revealed itself as important lessons and generous opportunities from a confident, but modest, supervisor. This considered supervision was strengthened further by concern and advise on fitness, rest, and a work-life balance. I'd also like to thank Wendy Hall for her support, guidance, and the opportunities she created that enabled me to carry out my PhD. I'd like to thank Nigel Shadbolt for both the support and guidance provided as my internal examiner. Further, I'd like to thank Gary Marchionini for agreeing to be my external examiner. Thanks also to Ryen White, Robert Capra, Nick Belkin, and Duane Degler who have advised on aspects of the work or provided resources to expand it. Many of my studies and experiments could not have been completed without the amazing interfaces and systems built by Alistair Russel, Joe Lambert, and Daniel Smith. Many further opportunities for studying users were provided by other researchers who trusted me with aspects of their projects and experiments, including Gary Wills and Mark Weal. I'd like to attribute at least some of my progress to the regular cynicism provided by Hugh Glaser, Terry Payne, Kirk Martinez, Mark Weal, and of course Maria Karam, who also provided friendship and support along with, the Owens', Paul André (also for providing plenty of cycnicsm), Rikki Prince (for organising breakfast too), Ilaria, Reena, and Ash. Finally, and very importantly, I'd like to thank Emily for supporting throughout the process despite the difficulties it brings to a young family of three.

for Emily and Ashton, my parents (Geoff and Sheila), my sister Lois, and the Cullis family

Chapter 1

Introduction

> *On IBM's Web site, the most popular feature was the search function, "because people couldn't figure out how to navigate the site," said Carol Moore, IBM's vice president for Internet operations. The second most popular feature was the 'help' button, because the search technology was so ineffective.*
>
> **Tedeschi, New York Times, 1999**

Search is a *very* loaded term. Many narrowly, and perhaps understandably, associate search with the service provided by Google, which can be used to find webpages on the World Wide Web. schraefel (2009) describes how Google's dominance, as the primary means of search for most people, has almost blinkered our understanding of how to find information. Searching, however, is something that many people do numerous times a day, for: websites on the internet, products within certain websites, files on a computer, and even numbers in telephones. Consequently, a lot of academic research and industrial investment has focused upon improving the speed and accuracy of search systems to the point where Google returns most searches on the entire of the indexed web (recently reaching 1 trillion pages[1]) in less than a quarter of a second. Similarly, in responding to the quote at the top of this introduction, IBM spent more than $1 million on the search functionality of their website, over a few months, and with over 100 people working on it. The result, however, was that sales increased 400% and website support requests went down 80% (Tedeschi, 1999).

What Google, IBM, the many other search providers, and e-commerce vendors further demonstrate is a) that getting search right is important and in great demand, b) that getting search right costs a lot of money, and c) that getting search right makes a lot of money. If getting search right is important, then how do we check that our search interfaces support the right kinds of search? Further, how can we get search right in the first place, during the early in the phases of design where it is cheap (Pressman,

[1]http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html

1992), rather than spending millions on improving deployed services? Currently, neither the search communities nor the human-computer interaction communities can answer these questions alone. There is a gap between these expertise, which makes the task of answering these questions an expensive and complicated one to achieve. To answer these questions, therefore, this research has focussed on how the knowledge and expertise of the search communities can be united with that of Human-Computer Interaction (HCI), and utilised to create a method that evaluates the usability of search interfaces.

In fact, searching is an activity that has existed for much longer than the computer, for: books in libraries, information in books, food in shops, memories stored away in boxes, friends in crowded places, and notes made in journals. It is perhaps not surprising, therefore, that the study of Human Computer Interaction (HCI) is becoming increasingly prevalent within the broad area loosely-defined as search, to ask: is an empty text box, waiting for search keywords, always an appropriate interface for search? Many would agree that keyword search will not *always* provide the best form of interaction to searchers (White et al., 2006; White and Roth, 2009). The limited cycle of query-results-query-results can be very constraining for situations that require browsing or learning about an unfamiliar domain of information. White and Drucker (2007) would argue that keyword search is effective for at *least* 17% of the time, and potentially appropriate for another 60% of the time. Some of our own research at Southampton, however, has shown that while searching for news footage from within a single website, alternatives to keyword search were used in around 50% of searches (Wilson and schraefel, 2008c). Consequently, this research has also focussed on trying to understand the ways that we search for information, in order to assess the support provided by search interface designs, whether keyword search is right for the job, or less familiar alternative modes of interfaces.

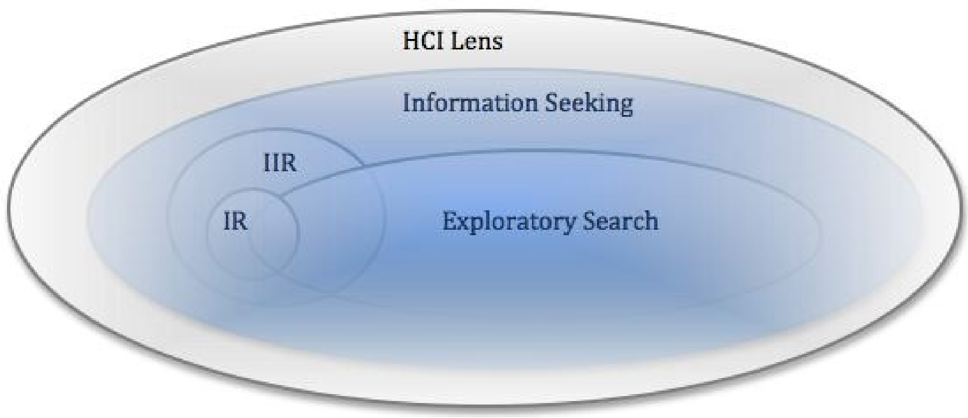


FIGURE 1.1: Diagram showing how the various research fields within information seeking are related. Familiar services like Google support IIR, but consequently only a small portion of Information Seeking activities, including alternative exploratory modes of Search.

In the search-oriented communities, there has also been an escalating interest in the human searcher aspects of search systems over the past three decades (Robertson and Hancock-Beaulieu, 1992), which is explained further in Chapter 2. Almost since computers were first invented, the need to get information out of a computer has motivated the study of algorithms to do so, known as Information Retrieval (IR). Informally, as the interest in users has increased, research has branched over time into: Interactive Information Retrieval (IIR), which aims to increase the interactivity of keyword search systems like Google; Information Seeking (IS), which focuses on searchers and explores how keyword search systems are involved in the larger schemes of search; Exploratory Search (ES), which investigates alternatives to paradigms like keyword search; and most recently Human Computer Interaction and Information Retrieval (HCIR), which explores how HCI practices should be applied to each of the above. The relationships between these areas are shown in Figure 1.1, where IS consumes both ES and IIR, where IIR consumes the research into IR. HCIR, therefore, becomes an HCI lens over IS.

1.1 Problem Statement

With the growing concern over searchers and their search interactions (Robertson and Hancock-Beaulieu, 1992; White and Roth, 2009), the challenge arises as to how advances in IS interfaces can be evaluated (Marchionini, 2009). Where traditional IR evaluations have typically focussed on systems and ignored the user, new evaluation techniques are required to incorporate the unpredictability of human searching behaviour for IS. Similarly, while HCI practices may identify general usability issues with interfaces, including search interfaces, they do not assess the support for different kinds of search and the occasions they might be required. Any work in this area, from an HCI perspective cannot, therefore, ignore the academic fields of search, nor can IR researchers ignore the expertise available in HCI.

Currently, the alternative is to use empirical HCI methods with search oriented tasks, as with the $1M investment made by IBM, but there are many ways to support a support search oriented tasks, and many interface design ideas that have been proposed and studied by the IS community. Consequently, while the designs might be influenced by IS research, and the evaluation methods influenced by HCI practices, we lack an evaluation method that is influenced by both IS and HCI. Consequently, the focus of this thesis is on a search-oriented usability evaluation methodology for HCI, that is built upon the wealth of knowledge generated from Information Seeking research.

Having identified a focus on uniting the expertise of both IS and HCI communities to develop a usability evaluation method especially for search interfaces, several research questions arise:

1. Can well-established search theory be operationalised effectively to produce an HCI method?
    - What theory and models of information seeking exist?
    - Are there appropriate models and classifications that can be operationalised?
    - Can any be used in combination?
    - What would they tell us about search interfaces?
    - What type of method will it produce that can be used?
    - Are there any similar or existing methods?
2. Can this new method accurately predict search-oriented usability issues?
    - Can it find real strengths and weaknesses of search interfaces?
    - What types of strengths and weaknesses does it identify?
    - Does it produce false positives or false negatives?
    - Does it capture important issues?
    - How does it compare to other methods?
    - How does it relate to any similar methods if they exist?
3. Can this new method fit realistically within the working practices of HCI experts?
    - Can it be used by non-search experts?
    - Can it be used faster, or in a comparable period of time to other methods?
    - Would the benefits outweigh the cost of learning to use it?

In the next section, the aims for such a methodology are discussed in more detail, followed by a discussion of the research approach to answering these research questions and achieving the aims of the work.

1.2 Motivation, Aims, and Approach

The aim of this doctoral work, as stated above, has been to operationalise IS research, so that it might be used by those designing, building, or evaluating search interfaces. Originally, the work was motivated by research into a faceted browser named mSpace (schraefel et al., 2006), which had been designed, built, and evaluated by a team of HCI and Semantic Web academics and students. While user studies demonstrated the efficiency of certain capabilities of the browser (Wilson and schraefel, 2008c), particularly for complex and exploratory search tasks, no research methods would explain whether the interface would fit in with models of searching behaviour or searcher needs. The principles of most HCI techniques apply generically to interfaces and users. Instead, this work has focussed on producing an HCI tool that considers search interfaces and searchers by taking advantages of the wealth of expertise produced by that domain of academia. For HCI practitioners working on search interfaces, therefore, this work provides a specialised tool for the domain of searching. While the skills and experience of HCI practitioners may not be directly in information systems, this work has aimed to provide clear insight into how to improve a search interface. For IR experts, however, who are building and testing new search systems, this work provides a practical interface evaluation tool to help check and/or improve the interactivity of their software. Such a tool would be important if their core skills are in algorithmic developments, rather than in designing user interfaces.

Several aims were identified for building an operationalised tool that captures and builds upon the expertise of information seeking theory:

- Capture the interaction model of a search interface
- Understand when, or under what contexts, these interactions help Information Seeking
- Be simple enough so that it can be applied quickly and easily by anyone choosing to evaluate a search interface
- Be generic enough to assess many types of search interface
- Gather at least enough insight to confidently make improvements to a search interface design

As this work aimed to produce a new search-oriented usability evaluation method, based around established theories and models, the research approach taken resembles that of the development of other, now established methods. Although there is no specific guide on how to build such a method, there are several examples that can be followed, which include phases of: theory identification, development, application, validation, extension, further validation, and so on, with the extension and further validation often repeating numerous times. Peterson (2000) for example, went through these stages in the development of the multi-point scale for questionnaires. Similarly, O'Brien and Toms (2008) report on the progression through similar stages while developing a framework to evaluate user engagement with software. Further, the development of the GOMS approach, discussed in Chapter 2, was designed (John et al., 1985) around a theoretical model of human information processing (Card et al., 1983), and was initially validated with an example study 2 years later (John and Newell, 1987), and then extended a further 3 years later (John, 1990). In fact, the GOMS model was then extended and revalidated many times thereafter by other authors (Gray et al., 1992; Gong and Kieras, 1994). Similarly, initial validation of the Cognitive Walkthrough method was reported by (Lewis et al., 1990), and modifications were proposed in 1992 (Rowley and Rhoades, 1992) that weren't truly realised until 8 years later (Spencer, 2000). Notably Blandford et al. (2008) presented a 10 year plan for designing and validating a method called PUM. In the three years of this doctoral work, several key stages were achieved: including Theory, Application, Validation, Extension, and Re-Validation. In the next section the structure of this thesis, describing the advances through these key states, is presented. Chapter 6, however, discusses the continuing and future work for understanding and improving the new method in the longer term.

1.3 Overview of Thesis

The doctoral research carried out to meet the aims listed above is described across the following sections:

- Chapter 2 presents *Related Work* in five main topic areas. As the aim of this work is to produce a search-oriented usability method, grounded in information seeking theory, for evaluating search interaces, the five areas are: 1) IS theory, 2) IS interfaces, 3) the interface design process, 4) usability evaluation methods (UEMs), and 5) the evaluation of UEMs.
- Chapter 3 presents *The Sii 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 three of the models described in the previous chapter. The framework has been designed to use these models in a novel way to evaluate the designs of complicated search interfaces, in a quick, cheap, and repeatable manner 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 a *Validation* of the framework, which has been applied to a) strengthen its design, b) show that it can be used to accurately predict the strengths and weaknesses of interface designs, and c) answer the research questions listed above: regarding the structure, accuracy, and suitability of Sii for real usage scenarios. 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 presents two *Extensions* to the framework that have been investigated for use in certain situations. First an extension modelling the increasing complexity of a design is introduced as a potential counter measure for the framework, which otherwise pushes to provide unbounded amounts of functionality. Second, an extension for collaborative search has been produced to show that, with additional modelling, teams and groups can be considered as well as individuals.

- Chapter 6 gathers the *Conclusions* of the research. First, the research phases are summarised, and the contributions of each discussed. The chapter continues by discussing how research will continue, including future work in: applying, validating, and refining the framework with different and challenging scenarios. While the framework has been used regularly to critically assess search interfaces, the future should focus on opportunities to use Sii when actually developing new information seeking interfaces. Finally, Sii's progressive use by other researchers and designers within the HCI and search communities will be encouraged, by collaborating with early adopters of the technique. The chapter concludes by summarising the aims, motivations, and contributions of the research.

There are also three Appendices to the thesis:

- Appendix A presents the full definitions of the 32 tactics provided by Bates (1979b,a).

- Appendix B provides a series of additional graphs, which are not required directly during the discussion of the case studies in Section 4.3. Although also available online, they are included in this appendix, so that readers can investigate further while reading if they want to.

- Appendix C presents practitioners guide to using the Sii website, describing the exact process of applying the framework in more detail. Further, it describes, by example, how to analyse and interpret the results produced by the framework, and then how the framework can be used to analyse the strengths of proposed redesigns.

Chapter 2

Related Work on Information Seeking, Development of Information Systems, and Evaluation Methods

> *If we consider that unlike art IR is not there for its own sake, that is, IR systems are researched and built to be used, then IR is far, far more than a branch of computer science, concerned primarily with issues of algorithms, computers, and computing.*
>
> **Tefko Saracevic, Professor II at Rutgers University**

This chapter discusses related work for the research presented in the remainder of the thesis. As this doctoral research is on the development of a Human-Computer Interaction (HCI) usability evaluation method for Information Seeking (IS) interfaces, both HCI and IS agendas are covered through a series of sections. More specifically, these sections cover: IS Theory, IS interfaces, the interface design process, usability evaluation methods (UEMs), and the evaluation of UEMs. Each of these are introduced briefly below.

The Information Seeking theory section first provides a top-down overview of the the different research agendas surrounding how people search. After this overview, a more detailed review of information seeking models is presented, including: process models, stratified system models, and searcher situation models. Three of these models are used in the construction of the Sii analytical inspection framework described in Chapter 3.

With the IS theory described, a series of IS interfaces are presented, including familiar systems like Google, and several others that are involved in analyses throughout the thesis. Search interfaces pervade our online experiences, whether it is searching for webpages, or finding products on e-commerce sites. While there are many example interfaces that can be presented, the focus in this section is to cover the particular designs involved later in the thesis.

Before discussing the available methods for evaluating search interfaces, an overview of the typical interface design process is provided. This doctoral work is not researching the design process in particular, but it is presented here to establish common terminology for the following sections. As part of this section, the consistent trade-off between increasing functionality and maintaining intuitive simple interaction is briefly introduced. This constant challenge during the design process provides context for what the Sii framework, and one of the extensions presented in Chapter 5, try to support.

In order to situate the analytical inspection framework presented in this work, Section 2.4 presents related research on different usability evaluation methods (UEMs) available for use at different stages of the design process. First an overview of the varied types of UEMs, such as user studies and expert methods, are presented. Then a categorisation scheme is described providing context for the aims of different UEMs, which are each briefly described. This categorisation and classification of UEMs is later used to explain how Sii relates to other methods in Chapter 3.

Finally, one of the key areas of related research for this work revolves around how UEMs, like Sii, are tested, validated, and evaluated. The work from Chapter 4 onwards focus on validation, example analyses, and how future work will continue to shed light on the Sii framework. Section 2.5 of this chapter, therefore, provides context over how the work achieved so far, and the planned future work, fit in with the methods used to develop UEMs like Sii.

2.1 Information Seeking Theory

Information is pervasive throughout our lives and, as visualised simply by Tom Wilson (1999) in Figure 2.1, searching is only part of how we interact with it. Systems like Google, for example, would represent only the inner circle of this onion model. Instead, there is much more to the seeking and use of information beyond simply searching. As we develop as a child, and throughout our lives, new information helps us to understand the world and how we can interact with our surroundings (Piaget, 1962). The simple notion that new information causes a transition between a former and new state of knowledge was formalised by Brookes (1980), simply suggesting that *k+i=k'* where *k* is the previous knowledge, *i* is the new information, and *k'* represents the new state of knowledge. Further, Brookes noted that new information, *i*, might arrive a) by chance, b) through monitoring an information source, or c) while seeking other information.

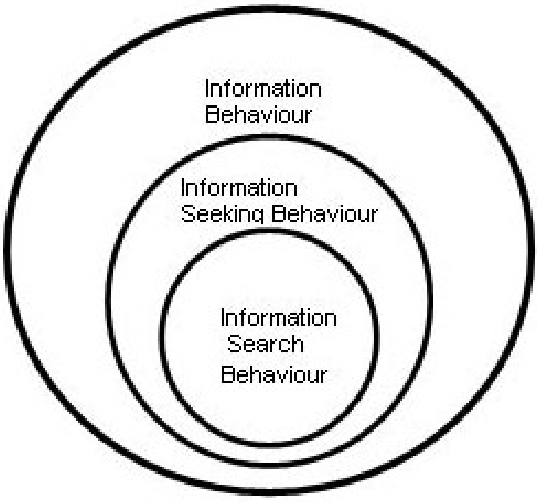


Figure 2.1: The onion model of information behaviour presented by Tom Wilson (1999).

In more recent work, Godbold (2006) summarises research into general information behaviour with Figure 2.2. Here she continues the notion that information causes a transition between states of knowledge (Brookes, 1980) by showing what information behaviours could exist between the former and newer states. Notably, once a knowledge gap (Dervin, 1992), or anomalous state of knowledge (Belkin et al., 1997), has been identified, there are three things a user could do: bridge the gap, close the gap, or ignore the gap. From Godbold's model, we can see that information seeking, and within it information search, is only one of the actions a user might take to bridge the gap. Searchers may also choose to construct or create additional information if unable to find any. Created information may be rationally deducted or irrationally generated. To close the gap, people may choose to spread or dispute information, or destroy/ignore information. Finally, people may choose not to cross the gap, by making a mental note for another time, or avoiding information. Goldbold evaluated each of these in more detail as well as reviewing the various sources of theory that contribute to the model. Further she points out that users may switch between these different strategies over time (choosing to eventually cross the gap), or overlap them (creating some information and disputing other parts). 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 researchers to analyse search and why businesses invest heavily in providing effective search support for customers accessing their products. Tedeschi (1999) reports on the improvements made by IBM to the search features of their website. At a cost of over $1m, the improvements increased sales by 400%. Further, the demand is only emphasised by the decades of documented research into information science, library science and information retrieval research.

Driven by the desire to find information, the research into search has focused on many

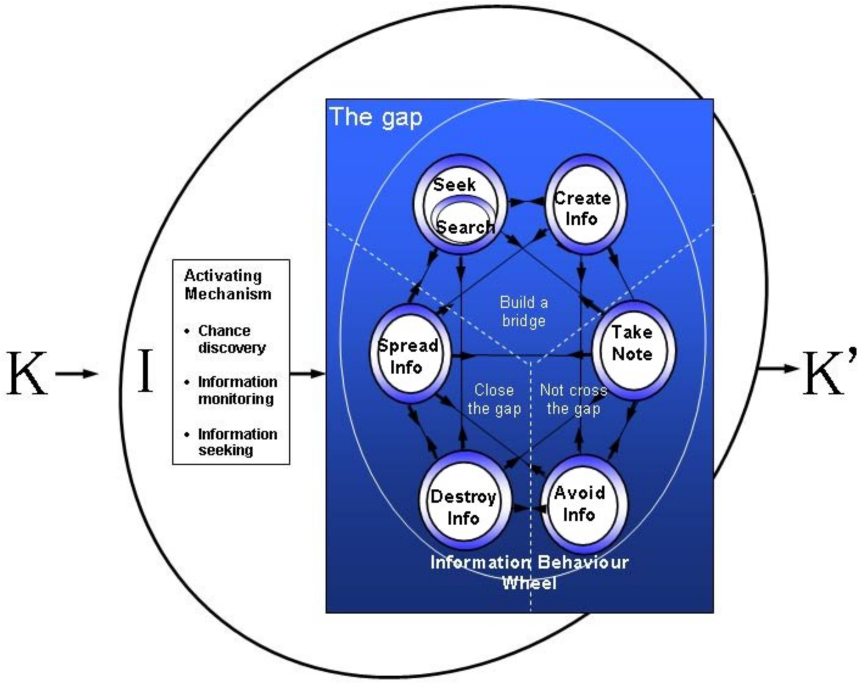


FIGURE 2.2: The wheel model of information behaviour presented by Godbold (2006).

different areas. Information Retrieval is a well known example that has focused 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 interactive view, however, is still focused 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; White and Roth, 2009), has focused particularly on the alternatives to keyword searching based on what has been learned from Information Seeking research. Each of these is described in more detail below.

2.1.1 Information Retrieval

Information retrieval research has focused on a very simple metaphor of search: tell me what you are looking for and I'll do my best to find it (White and Roth, 2009). This focus has, therefore, mainly investigated the keyword search interaction that has been made well known by web 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 return to the user. Not surprisingly research in information retrieval has focused 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.

2.1.1.1 Retrieval Algorithms

Algorithms for matching queries to documents via constructed indexes have received a lot of focus, and much could be written here about them. As the focus of this work is not on improving such algorithms, however, a short overview is presented to provide context for how search interface designers expect the underlying systems to respond to interactions.

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 (Sparck Jones, 1972). This means that common words, such as 'the' and 'and' do not have a large effect on a results returned, because they occur many times in many documents[1].

Other advances showed that word stemming (Lovins, 1968) was important for indexing, and that finding the root 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 and Sparck Jones, 1976) and BM25 (Robertson et al., 1998), which is one of the most widely-used approaches now.

2.1.1.2 Measures of Success

Over time, two specific measures of success for information retrieval algorithms 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 potentially not relevant, and making sure that only relevant items are included reduces the number of documents retrieved.

[1] Although really common words like 'the' and 'and', referred to as stopwords, are usually filtered out from a retrieval algorithm.

Another predictable measure for retrieval algorithms is speed. Early publications from the founders of 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 a quarter of a second. More about evaluation measures is discussed in Section 2.6, and also by Wilson et al. (2009a) and Hearst (2009).

2.1.1.3 Testing Environments for Retrieval Algorithms

The desire to test any new ideas for algorithms has led to the development of communally available test sets. These test sets have been used regularly by the Text Retrieval Conferences (TREC) where researchers compete to provide the best results over different datasets (Harman, 1997). These test environments have allowed research into retrieval algorithms to thrive, by providing a common and controllable dependent variable across institutions. Developing these datasets has required the contributions of many researchers to produce both indexed document collections and human assessments of relevance for testing benchmarks.

2.1.2 Interactive Information Retrieval

The research into Information Retrieval has certainly produced efficient and important contributions for search systems, but another research stream has looked outside of the IR 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, Koenemann and Belkin investigate Relevance Feedback mechanisms (Salton and Buckley, 1990), which track implicit relevance judgements made by users to improve queries for future searchers. 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. The work by Koenemann and Belkin (1996) was one of the first to provide empirical evidence that improving interaction, by allowing users to control iterative query refinement, provided significant improvements for precision and recall over automatic methods.

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). Borland, for example, discusses measures that consider progress as an increased number of queries issued over time. Instead of evaluating how long a system takes to answer, and how accurate the results are, interactive information retrieval is more concerned with how long it takes users to achieve a goal. The desire to build consistent evaluation frameworks for interactive information retrieval is similar to the motivation that produced the TREC conferences, which consequently began an interactive track. The interactive tracks of the TREC conference, however, were retired in 2002[2], as it was hard to control the number of variables between systems (Dumais and Belkin, 2005).

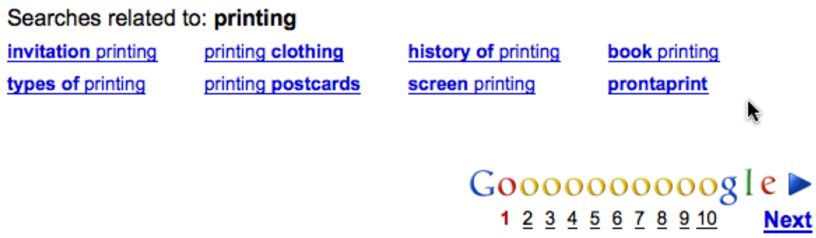


FIGURE 2.3: Query expansion in the commercial online search engine: Google.

These examples of a more interactive experience with a search system were often modelled as conversations between searchers and librarians. Several models, for example, were based on dialogues or conversations between typical searcher and librarian roles: Conversation for Action (Winograd and Flores, 1986) and the Conversational Roles Model (COR) (Sitter and Stein, 1992; Stein and Thiel, 1993). In these models, the system is considered as playing the role of the librarian, encouraging the searcher to expand on their information needs. Such models, however, were representative of the broader sense of user-centric IS, which is discussed in more detail below.

2.1.3 Information Seeking

Information Seeking represents a much wider view of search, covering activities beyond simply interacting with a keyword search system, to include the whole process from identifying an information need to resolving it. While, the research into interactive information retrieval proposed many advances to information retrieval, many more questions about users and their tasks were identified. 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

[2]The interactive track is being revisited in 2009, with results to come.

by Bates (2002) in Figure 2.5, but only breaks down users into their searching activities and their current understanding and motivations. It has taken much more research to begin to understand the complexity of human searchers.

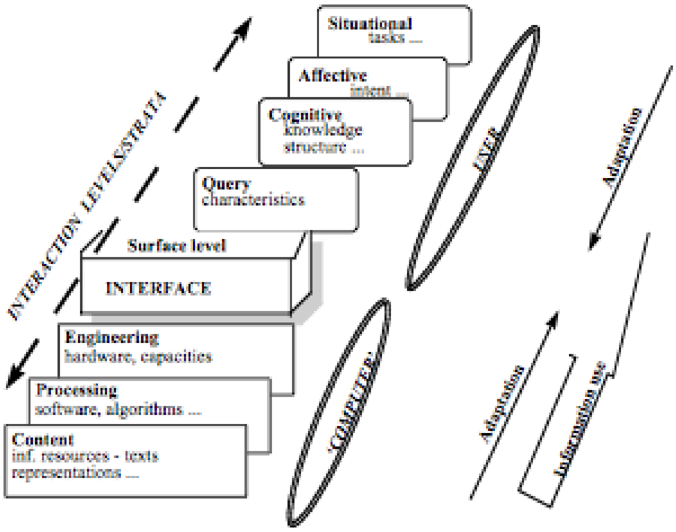


FIGURE 2.4: The stratified levels of a search system, involving both users and technology, from Saracevic (1997).

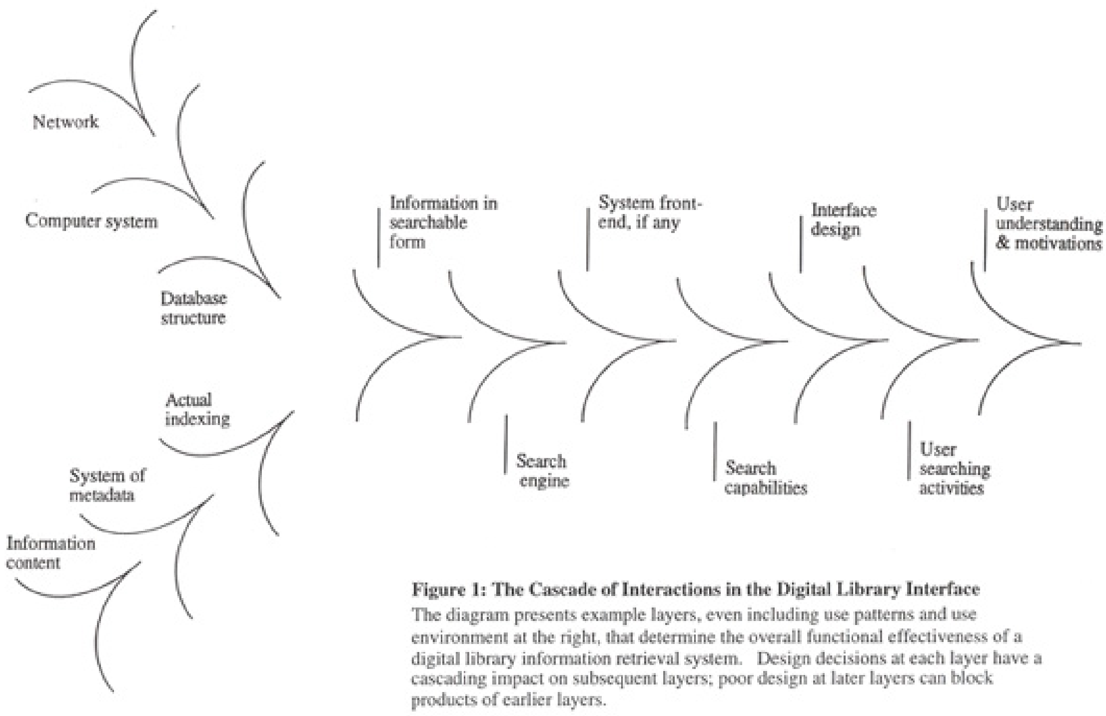


FIGURE 2.5: The cascading levels of interaction between parts of an information retrieval system, from Bates (2002).

Recent work by Jarvelin and Ingwersen (2004; 2005) has summarised a lot of research into the context of users with the model shown in Figure 2.6. In line with the descriptions above, the most central context (IR context) is mainly constructed of an iterative loop of matching documents to user queries. As with the model by Wilson (1999), in Figure 2.1, IR occurs within the context of IS activities. IR findings make up part of an IS process, which leads towards achieving a larger IS goal. Again, outside of the IS context, is the Work Task context. Work Tasks are the real world tasks which motivate or cause us to seek information, such as writing a report, or booking a holiday. Consequently, one or more IS tasks, which may be made up of one or more IR tasks, contribute to the process of completing a Work Task. The final level of context is the socio-organizational and cultural context, which represents the surroundings and environment that have led to the work task. Whether the work task is being completed in a place of work, or a hobby performed in the home, this final context will affect factors such as how thorough the search is, which service is chosen for the search, and for how long. As with the onion model by Wilson (1999), the support by Google typically only represents the inner-most context described in Figure 2.6.

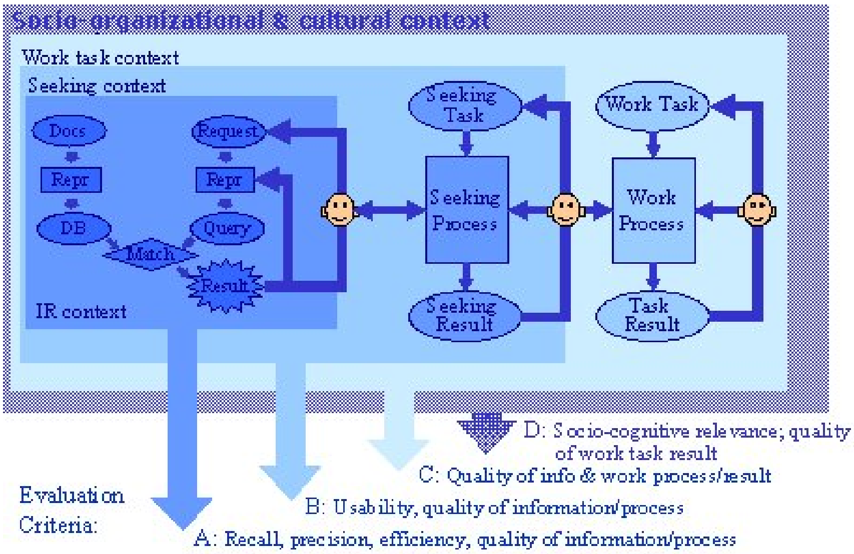


FIGURE 2.6: The model of search contexts provided by Järvelin and Ingwersen (2004)

Further to the model of contexts, Järvelin and Ingwersen (2004) break down IS research into nine dimensions, as shown in Figure 2.7. Each of these nine dimensions is used to further define, for example, the various research areas described so far. Various forms of happy or unhappy faces are used to loosely represent the amount of focus the dimensions have received in each field. The work task dimension is that mentioned in Figure 2.6, which is discussed in more detail by Byström and Hansen (2002). The search task is the lower level that covers both seeking and retrieval processes, this is discussed further below in the form of the various process models of information seeking. The actor dimension, also discussed further below, represents the search situations and contexts that affect users, such as the socio-organizational and cultural contexts (Figure 2.6) and the situational, affective, and cognitive levels of Saracevic's model (Figure 2.4). As the users' current state of knowlegde, including of their own needs and tasks (mentioned briefly by

Bates in Figure 2.5) is affected by their understanding of their own information need, both the perceived work and search tasks have also been researched. In Section 2.1.4, Exploratory Search is discussed, where users evolve in their understanding of tasks, needs, and even systems, while searching (White et al., 2006; White and Roth, 2009). The Document dimension covers different document structures. Some research has focused on the unique requirements for special formats such as XML documents (Fuhr et al., 2002) or for information within large documents like books (Wilkinson, 1994). The search engine dimension is largely focused on the algorithmic concerns of the Information Retrieval community, discussed above. The arguably misleadingly-titled interface dimension is concerned with the systems interface with service components, which has been part of the technological focus of the Information Retrieval community. Finally the Access and Interaction dimension is that which attracts particular focus by the Human Computer Information Retrieval community. Further, the aim of the work described in this thesis is to develop a usability evaluation method for designing better interactions, using research from the Actor and Search task dimensions. The Actor and Search task dimensions are discussed in more detail below.

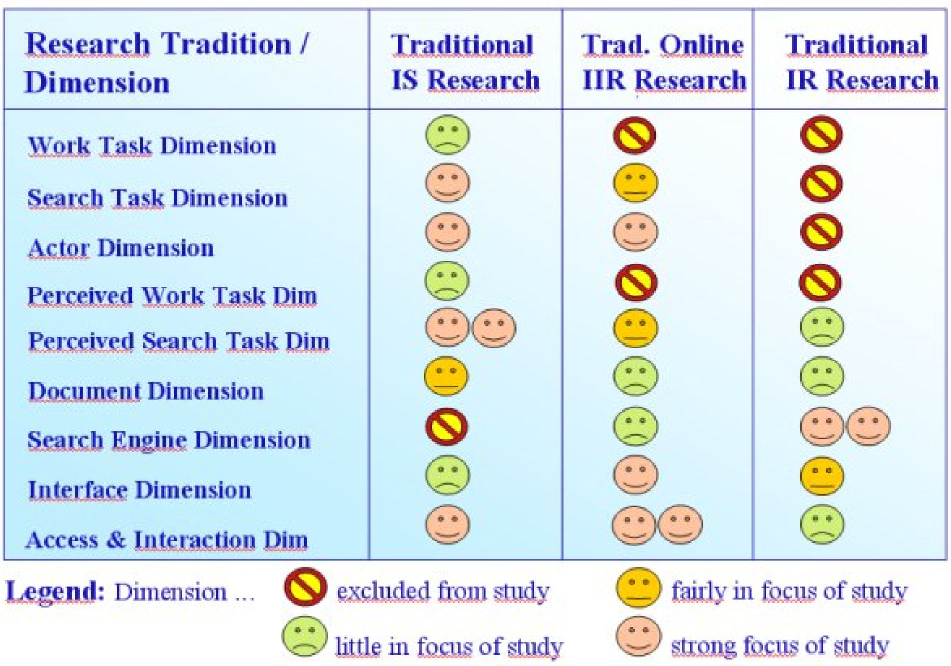


FIGURE 2.7: Research areas broken down by the dimensions of information seeking, provided by Järvelin and Ingwersen (2004)

2.1.3.1 Actor Dimension

Research by Pharo (2004) summarises much of the existing research into the situational aspects of users. First, Pharo highlights that the user is constrained by their knowledge of their work task, search task, and the search system being used, which relate to the two 'perceived' dimensions from Figure 2.7. Further, Pharo notes that the searcher's previous education is often considered a demographic factor for describing the distributions of study participants, along with gender and age. Factors such as motivation, tenacity (the inclination to invest in the search), uncertainty, and attention to the task are also considered factors. Finally, Pharo's definition of searchers ends by saying

> *Other factors are also often used to characterise searchers, for example background, age, gender, cognitive style, experience etc.*

Unfortunately, these dimensions of users, like those mentioned in the models by Bates (2002), Saracevic (1995), and Järvelin and Ingwersen (2004), have not been qualified, quantified, or defined in more detail.

One of the key models of the Actor dimension that defines both the facets of a user's search situation and values within these facets, was presented by Belkin et al. (1993), who broke down users into four binary dimensions, 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* 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[3] 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[4] (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).

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. Further, Marchionini (2006) notes that as search engines are primarily concerned with Precision, rather than Recall, the extent of support for ISS conditions is further reduced by poor support for learning (*Goal*). These conclusions drawn from the ISS conditions are somewhat validated by work done in 2004, which estimated that around 81% of search engine users

[3] CHI is an annual ACM conference on the human factors of computing

[4] The Association of Computing Machinery's digital library, available at http://portal.acm.org/portal.cfm

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.

viewed only one result page (Beitzel et al., 2004). Further, White and Drucker (2007) discovered that keyword search may only support around 17% of searches efficiently, and supporting the rest but ineffectively. 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 begin with 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*). Consequently, Google 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 Section 2.2.3, 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.

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. Cool and Belkin studied 14 people working in different roles within Boeing and extended the four binary dimensions by Belkin and colleagues, to five hierarchical dimensions, most with more than two options. Dimensions included Communication (Medium, Mode, Mapping), Behaviour (with 9 sub-dimensions such as create and evaluate), Information Objects (level, medium, quantity), Interaction (object, process, and degree), and Interaction Criteria (not fully defined, but includes alphabet, authority, date, person, and time). This extension was later tried and validated by Huvila and Widen-Wulff (2006) by applying the extended model to multiple scenarios, but can create 1944 unique conditions (using only the 8 suggested but not completely defined Interaction Criteria). Without the Interaction

Criteria, should we consider that it cannot be fully defined, the other four dimensions create 243 unique user profiles.

Despite the updated criticism of ISS's by Pharo (2004), Kriewel used Pharo's search situations in conjunction with the typical interaction patterns defined by Belkin et al. (1995) for each of their 16 ISS user conditions, to define the Digital Library system: DAFFODIL. The on-going work on the DAFFODIL system was designed to recognise some standard situations and recommend various functions that support the different ISSs. More recently, Kim (2009) extended the work to consider some web-specific behaviours, such as chaining (following links), which creates 54 specific ISS profiles, however the work then restricts the combinations to 14 which were frequently demonstrated by study participants. Part of Kim's contribution, therefore, was to begin to quantify the amount that certain ISS profiles occur.

Classifying search situations has naturally led to research into developing personalised search interfaces, where, for example, an individual searcher receives modified results according to a model of their search history. The extent of investigation into personalised search often extends to the communities that focus on modelling users over time. John and Mooney (2001), for example, applied user modelling techniques to develop adaptive search interfaces. Work from user modelling conferences and personalisation is not discussed here, as the concerns of this doctoral work are not in improving search experiences at an individual level, but to identify which types of user situations that a search system might support.

Similarly, situational factors are important for the design of location-based mobile services, which may include mobile search interfaces. Mizzaro et al. (2008) designed a contextually dependent mobile application framework that considered factors including location, time, and even posture, such as laying down. Mobile-dependent work is also not discussed here, but the applicability of Sii, the usability method being developed in this work, to mobile search interfaces is discussed in Chapter 5 regarding possible future work. More reseach into search situations is available at venues such as the Information Seeking in Context and Information Interaction in Context communities. These venues, however, are often concerned with investigating particular contexts, such as knowledge workers, school children, students, and the elderly, rather than classifying types of search situations like the research discussed above.

One of the key motivations behind classifying types of searchers is to understand differences in their search behaviours. It would be intuitive to assume that experts are more efficient at searching or have developed better strategies for fulfilling complex information needs. The next subsection discusses both search processes and different searching strategies in more detail.

2.1.3.2 Search Task and Strategy Dimensions

Process models are common in the field of information seeking (IS) and aim to describe the stages that searchers go through in order to achieve goals. Over time, these models have progressed from very linear concepts to iterative or dynamic representations. These are discussed below.

Linear Process Models

Figure 2.8 shows a comparison of two key IS Process (ISP) models by Ellis (1989) and Kuhlthau (1991), visualised together by Wilson (1999). By constructing this figure, Wilson noted the commonality of the two, and several other process models. Essentially, they typically each share the notion that searchers start with a realisation point, try to define their problem, perform some searching or browsing actions, analyse the results they receive, and stop when their need has been resolved. Wilson discusses these and other ISP models in more detail. Nearly all of these ISP models, however, have been supplied with a caveat: that users may jump back and forth through the process at will.

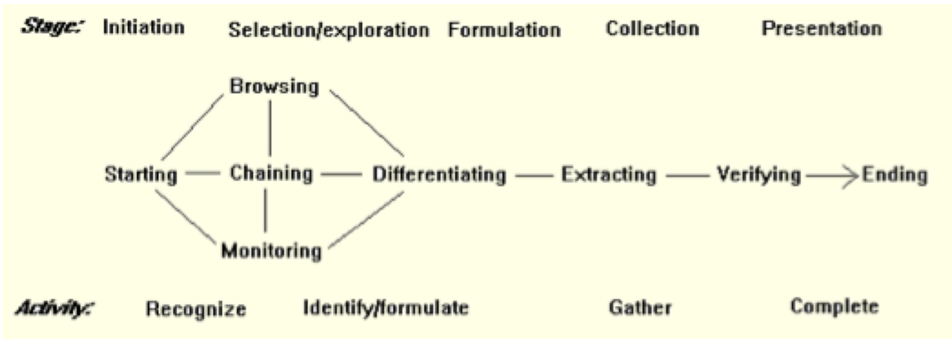


FIGURE 2.8: The process models by Ellis (1989) in the centre, and Kuhlthau (1991) at the top, as visualised together by Wilson (1999).

This freedom-of-movement caveat represents a common limitation of many ISP models: that users do not progress through linear phases, but jump between several active, passive, and reflective states in an unpredictable fashion. The largely unpredictable switching is modelled clearly in the ISP presented by Marchionini (1995), shown in Figure 2.9. In fact, the multitude of arrows highlight that the linear progression from left to right only represents the ideal or best path that can taken by a searcher and emphasizes that users are rarely able to take this best path.

While most of the key ISP models awkwardly cater for, ignore, or even abstract-out the fact that users switch frequently between stages, the model presented by Marchionini (1995) is the most explicit in representing the reasons and conditions in which changes occur. Marchionini crudely, as he describes it, models these switches by identifying both more and less likely paths that users may follow backward through the stages. Further, the absence of arrows between certain states implicitly highlights switches that do not occur. Another example to explicitly consider state changes was provided by Belkin et al.

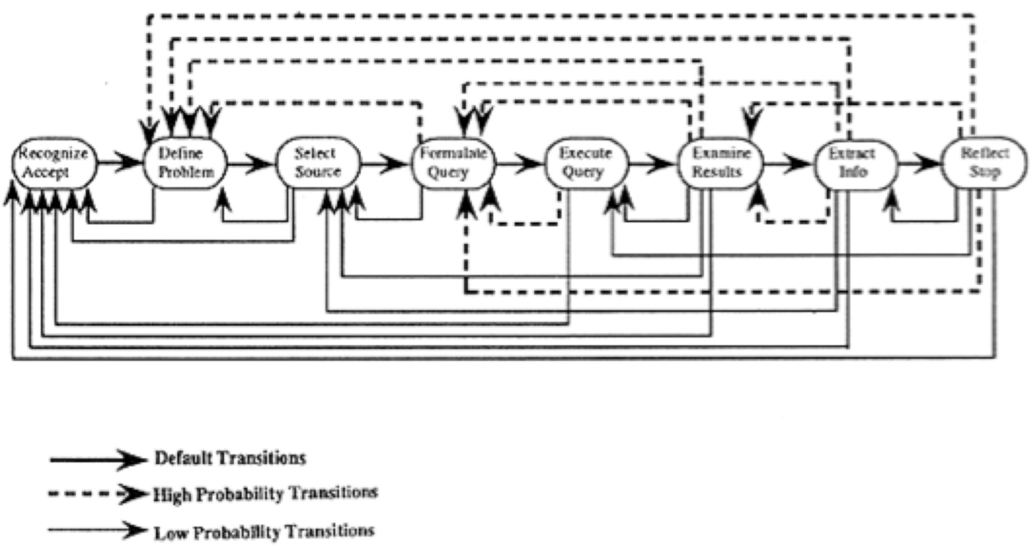


Figure 2.9: The information seeking process model stages defined by Marchionini (1995), noting the many potential backward loops.

(1995), who generated detailed scripts for the 16 ISS profiles described in the section above. Further, they identified many script entry and exit points where searchers could switch between searcher types. The full descriptions of these search episodes and scripts are extensive, and also fairly rigid, despite allowing users to transfer between them.

Non-linear Process Models

Rather than accommodating for non-linear behaviour, several other process models have integrated the notion of continuous or iterative search behaviour. The model of information seeking by Spink (1997), shown in Figure 2.10, clearly presents the ISP as made up of iterative cycles that contain feedback loops of searching, retrieving, and judging of results. Foster (2004) presented another non-linear model, which is described as analogous to an artist's palette. As a searcher/artist, defining (and re-defining), formulating, searching, analysing, and reflecting are used as needed to finish the job. Bates (1989) described such selective and continuous seeking behaviour as berrypicking, and Pirolli and Card (1995) described this behaviour as Information Foraging. Such models, as Bates points out, allow for the fact that real searching behaviour involves evolving knowledge and goals, and that resolving an information need can involve collecting pieces of information throughout a search session. Information foraging theory builds on this continuous seeking process, by discussing how labels, filenames, and textual descriptions, can all provide clues as to where sought information my lie in a system or repository.

Strategic Models of Information Seeking

From the continuous and non-linear models of information seeking, the question remains as to what tactics and strategies are used by searchers. Much research has investigated search strategies for information seeking (Moody, 1991; Pharo, 1999; Kriewel;

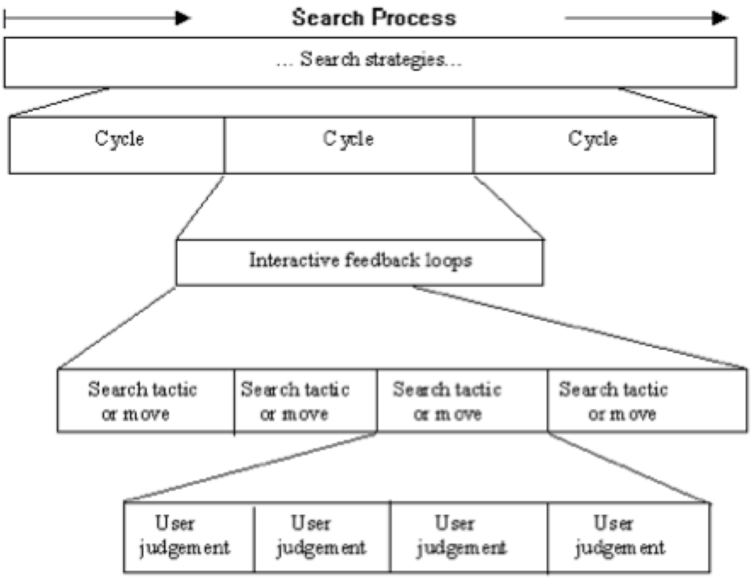


FIGURE 2.10: The cyclical model of the information seeking process by Spink (1997), highlighting that the search process involves iteration.

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 the aforementioned 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 the levels of automation provided by search systems. First, she identifies 5 levels of system automation ranging from complete user action to complete system automation. Although complete system automation would rely, as she discusses, on reading the searchers' mind, the process of describing these levels of automation allowed her to categorise the advances that had been made, showing some progression up the levels.

The levels of auotmation were then combined, as shown in Figure 2.12, with four levels of search activities: *Move, Tactic, Stratagem* and *Strategy*. Figure 2.11 presents these four levels with definitions. Each of these is briefly described here, and then discussed further in subsequent paragraphs. The lowest granularity level, a *Move*, 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 potentially endless combinations of *moves* that can be used to support a *tactic*, which depends on system implementation. *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.

LEVEL	NAME	DEFINITION
1	Move	An identifiable thought or action that is a part of information searching.
2	Tactic	One or a handful of moves made to further a search.
3	Strategem	A larger, more complex set of thoughts and/or actions than the tactic; a strategem consists of multiple tactics and/or moves, all designed to exploit a particular search domain thought to contain desired information.
4	Strategy	A plan, which may contain moves, tactics, and/or strategems, for an entire information search.

FIGURE 2.11: The levels of search strategies, presented by Bates (1990).

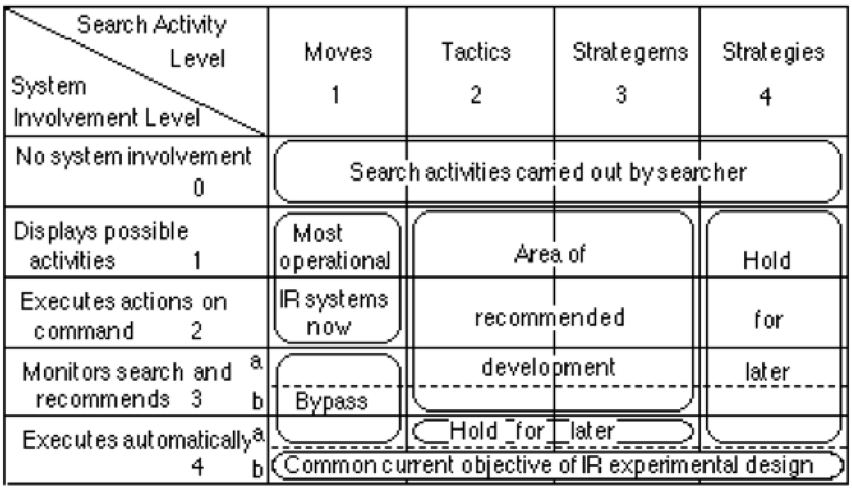


FIGURE 2.12: An overview of how different areas of research relate to system level automation and the levels of search strategies identified by Bates (1990). Automation level 3 a) relates to when a searcher asks for support, and b) when the system provides support regardless of need. Level 4 a) runs automatically and informs the searcher, and b) run automatically in the background.

Information Seeking Actions

Bates' definition of *Moves* is fairly flexible. She suggests that the evaluator may wish to consider, for example, entering a search formulation as a move, or that entering each search term be considered separate moves. The emphasis here is on consistency across evaluations, so that the evaluator uses a consistent level of granularity in to provide a fair comparison. This is a different concept to the notion of *Moves* defined by Fidel (1985), who suggests a set of moves that are more like specific instances of *Tactics* that still require several steps. Fidel, for example, identified re-ordering query terms as a move, which may involve several actions, such as removing a term from one position and re-entering it at another position. Further, as Bates defines *moves* as mental or physical, the acts of choosing which term to move, and then where to move it to, would also be considered steps (*Moves*) in the sequence.

Bates' notion of *Moves* is also similar to the concept of an Operator in the Keystroke Level Model (Card et al., 1980). Described in more detail in Section 2.4 on Usability Evaluation Methods, the Keystroke Level Model aimed to specifically measure the time taken to perform a task, by counting: keystrokes, moving the mouse, pressing a mouse button, releasing a mouse button, moving the hand between the mouse and keyboard, waiting for system response time, and any mental act. Card identified common or averaged timings for each such act. A keystroke, for example, depends on typing expertise, but on average takes around 0.28 seconds. Moving the mouse takes around 1.1 second. A mental move, which Card suggested may be cognitive (thinking) or perceptual (such as visually scanning), may take between 0.6 and 1.25 seconds. In their review of such cognitive modelling research, ten years after the Keystroke Level Model was proposed, Olson and Olson (1990) suggest that cognitive actions take on average 1.2s. Olson and Olson (1990) also discuss the applicability of such models, which is considered further here in Section 2.4. The notion that a mental act may take around 1.2 seconds, however, can be useful in estimating what counts as a mental *move* in Bates's model. Suggested examples of mental actions by Kieras (2001) include: initiating a task, retrieving a piece of information from memory, findings something on a screen, and choosing a search term.

Information Seeking Tactics

Tactics are sequences of *moves*, where the set of moves required to achieve a tactic is usually defined by the design decisions in the search interface. Where *moves* are like the Keystroke Level Model or GOMS (discussed in Section 2.4) operators, *tactics* are similar to the Method aspect of a GOMS model: a sequence of Operators. If, for example, a drop-down box automatically submits a form when the user makes a selection, then they are not required to perform the *Move* of pressing the submit button. In earlier work, Bates (1979a,b) identifies and defines 32 specific information search *tactics*, summarised in Table 2.2, and fully defined in Appendix A. Although identified at a time when boolean keyword search systems took minutes or hours to execute, the principles of these tactics still generalise to the designs of search systems today. While much research has abstractly discussed tactics and strategies performed by users, the list of *tactics* identified provided a key contribution to the information seeking community.

Tactics		**Short Definitions**
Monitoring Tactics	CHECK	To check that a search is providing results relating to the information need.
	WEIGH	To decide whether or not to continue with the current search.
	PATTERN	To identify a good or better way to resolve an information need.
	CORRECT	To correct an error made during search.
	RECORD	To keep found results that are useful.
File Structure Tactics	BIBBLE	To learn from metadata to understand structure and improve search.
	SELECT	To select part of an information need.
	SURVEY	To carefully review options if presented with a decision.
	CUT	To take an action that will cause a significant reduction of results.
	STRETCH	To use information or metadata for an unintended purpose.
	CLEAVE	To use a careful procedure to work through an ordered list.
Search Formulation Tactics	SPECIFY	To explicitly specify a desired item.
	EXHAUST	To expand on a query to widen recall to more documents.
	REDUCE	To refine a query to increase precision to fewer documents.
	PARALLEL	To expand on a query with additional synonyms.
	PINPOINT	To reduce a query with fewer synonyms.
	BLOCK	To explicitly remove results relating to a certain term.
Term Tactics	SUPER	To use a broader terminology.
	SUB	To use more more specific, or subordinate terminology.
	RELATE	To move to a similar synonym or term.
	NEIGHBOUR	To identify words commonly used together with submitted query terms.
	TRACE	To find additional terms in the results already found.
	VARY	To try variations in the structure of a query to see the effect on results.
	FIX	To see the effect of using different affixes or an alternative tense of a query.
	REARRANGE	To test the effect of term order on results.
	CONTRARY	To search by a term logically opposite to a preceding query.
	RESPELL	To try alternative international or incorrect spellings of terms.
	RESPACE	To try variations in spacing and grammar within queries.
Idea Tactics	RESCUE	To conceive an alternative search strategy to a currently unproductive approach.
	BREACH	To redefine and broaden ones definition of an information need.
	FOCUS	To narrow and refine ones definition of an information need.

Table 2.2: The 32 information seeking tactics, in five groups, identified by Bates (1979b,a).

The *tactics*, which were identified through empirical and qualitative investigations, fall into five categories. Monitoring tactics are performed to reflect on the process of seeking, such as deciding whether to continue on a search (WEIGH) and keeping track of key discoveries that form part of an information need (RECORD). File structure tactics, which are designed to take advantage of the classification schemes used, include task management activities like tackling parts of a problem in turn (SELECT) and looking up reference information for a result (BIBBLE). Search formulation tactics represent activities relating to the scope of a search, such as narrowing a result set (REDUCE) and excluding topics (BLOCK). Term tactics represent activities that affect specific queries, such as finding related terms (TRACE) and prioritising terms in the query (REARRANGE). Finally, idea tactics (Bates, 1979a) represent activities that relate closely to an evolving understanding of an information space, such as moving on from unproductive avenues of search (RESCUE).

As mentioned above, Fidel (1985) identified common instances of tactics: using AND to narrow a result set or adding a term to achieve a similar effect. Such instances of tactics were further defined by Shute and Smith (1993) and then later used by Wildemuth (2004) to analyse the differences in tactics between users of varying domain expertise. While these instances still remain, to some extent, implementation independent, Wildemuth (2004) criticised the notion of Moves, by Bates (1990), for being too focussed on implementation. Sii, the evaluation framework described in this thesis, is focused on implementation specifics, but the instances of tactics defined by Shute and Smith (1993) can provide examples of different ways the tactics defined by Bates can be achieved with keyword search interfaces. The notion of moves is discussed further during the explanation of the procedure for applying Sii, in Chapter 3.

Fidel's Move	**Definition**	**Bates** ***Tactic***
Expand 1	Enter a broader descriptor	SUPER
Expand 2	Group together search terms to broaden the meaning of the set	PARALLEL
Expand 3	Group together a descriptor with an equivalent role indicator	PARALLEL
Expand 4	Represent a query component explicitly only by qualifying another component with role indicators	PARALLEL
Expand 5	Supplement a specific answer set with sets representing broader concepts	SUPER

Table 2.3: Five example *Expand* 'moves' from Fidel (1985), with the correlating *tactics* from Bates (1979b).

Larger Strategic Activities

Bates' concepts of Stratagems and Strategies are much broader and and potentially un-boundable, and therefore undefinable terms, although many examples have been identified. Much research has investigated the strategies used by experts and novices (Hsieh-Yee, 1993), for example. They relate closely, for example, to analysing the tasks and sub-tasks involved in real working assignments. The example scenario used by Bates (1990) suggests that a strategy is to perform background research for a paper, and a stratagems might include scanning the index of a journal issue, or checking the previous publications of a particular author. As such, strategies are often discussed in the contexts of different working environments, such as the strategies of knowledge workers (Chin et al., 2009), or patent officers (Hansen and Järvelin, 2005). The research surrounding Exploratory Search, described in the next section, has been, in part, informed by the range of strategies that exist in information seeking.

2.1.4 Exploratory Search

The research into Exploratory Search (White et al., 2006; White and Roth, 2009) has focused on the design of alternative search interfaces to the basic, but dominant, keyword search style of information retrieval. Exploratory Search, therefore, is a form of information seeking. schraefel (2009) highlights the negative affect that the dominance of keyword search on the web has had on our understanding of how and when to provide alternatives appropriately. 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 strategies that are shown in Figure 2.13.

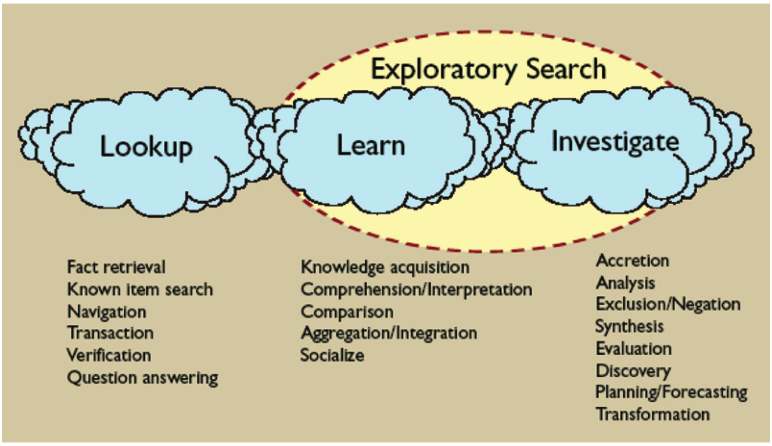


FIGURE 2.13: The activities associated with traditional and exploratory search, from Marchionini (2006).

It is the concern of the Exploratory Search community, that the strategies to the right of Figure 2.13, such as analysis, evaluation, planning, synthesis, and even relatively simple activities such as aggregation and comparison, are not well supported by the keyword search interfaces that are dominant on the web. Further, it is these complex strategies where tactics such as monitoring and idea *tactics*, for example, become particularly important to searchers. These circumstances also increase the liklihood that searchers will require tactics that support them in making many queries as they browse and explore related results. White and Drucker (2007) provide example empirical evidence towards the extent that exploratory searches occur. Only 17% of searches did not exhibit any exploratory behaviour, and 23% of searches exhibited almost entirely exploratory behaviour. Further, Teevan et al. (2004) noted that 61% of searches included more exploratory forms of search. These two studies, however, report on evidence from keyword-oriented search engines. In a study of an interface that supports both searching and browsing, Wilson and schraefel (2008c) showed that alternatives to keyword search were used in around 50% of searches.

In Table 2.4, the comparison of various search communities is extended to indicate which dimensions Exploratory Search (and HCIR discussed below) are related to the 9 identified dimensions.

2.1.5 Human Computer Interaction and Information Retrieval

As the investigation into more interactive alternatives to keyword search has developed another small community of HCI academics have focused on the design of search interfaces. Although significantly overlapping with the Exploratory Search community, research into HCIR is focused further on designs and implementations that realise the alternative strategies of information seeking, as shown in Table 2.4.

Dimension	HCIR	XS	IS	IIR	IR
Work Task Dimension	3	4	1	0	0
Search Task Dimension	3	3	3	2	0
Actor Dimension	4	4	3	3	0
Perceived Work Task	3	4	1	0	0
Perceived Search Task	3	3	4	2	1
Document Dimension	1	1	2	1	1
Search Engine Dim.	0	1	0	1	4
Sys. Interface Dim.	0	0	1	3	2
Interaction Dim	4	4	3	4	1

Table 2.4: The Search communities defined by Järvelin and Ingwersen (2004) across 9 research dimensions, and extended here to include the Exploratory Search (XS) and Human Computer Interaction and Information Retrieval (HCIR) communities. 0 represents the case where a dimension is not investigated, and 4 represents where it has been heavily investigated (the double smiley face in Figure 2.7).

Much research has been applied to design of more exploratory interfaces. Hearst (2006) suggests from experience that carefully constructed classifications support users better than automatically generated schemes. The reduction in workload afforded by automatically classifying results notwithstanding, Hearst suggests that thoughtfully designed classifications should be generated wherever possible. Enterprise search companies, such as Endeca for example, typically promote the use of semi-supervised classifiers to achieve the best of both automation and thoughtful design. Browsing and direct manipulation alternatives have shown significant improvements over keyword searching for users of mobile devices (Wilson et al., 2006b,a). Alternative and complementary modes of interaction, such as audio or other multimedia can encourage and support decisions making (schraefel et al., 2004). Further, research has shown that browsing interfaces can support users, if designed appropriately, in recalling additional factual information from areas surrounding points of interaction (Wilson et al., 2008). The next section discusses some of the interfaces that have been developed to support more exploratory forms of search and general information seeking.

2.2 Information Seeking Interfaces

In this section, searching and browsing interfaces are discussed that a) cover the range of search communities discussed in the preceding section, and b) introduce the reader to systems that are included in evaluations throughout the rest of the document. Significantly larger surveys of available search interfaces have been produced by Wilson et al. (2009a), and even more so by Hearst (2009). The section begins with familiar IR-oriented web interfaces including Google, and goes on to discuss examples that support more exploratory forms of search.

2.2.1 Typical Search Interfaces

The most dominant and familiar search interface online is Google, shown in Figure 2.14, which excels at providing cutting edge keyword search functionality. The basic interaction is simple. First, users choose one or more query terms that relate to their information need. After entering them into the search box and pressing the search button, a set of search results are returned in an ordered list. These results contain the title of the website result, provided as a link to that website, a (typically) two-line snippet of text that relate to the keywords entered by the user, and the pages web address. Although not the only functionality of this, and other search engines, the above interaction forms the basis of the way users typically use a search engine. If a desired result, assuming only one result is required to solve an information need, is not in the first ten results, the user is required to either view the next ten results, or change their query.

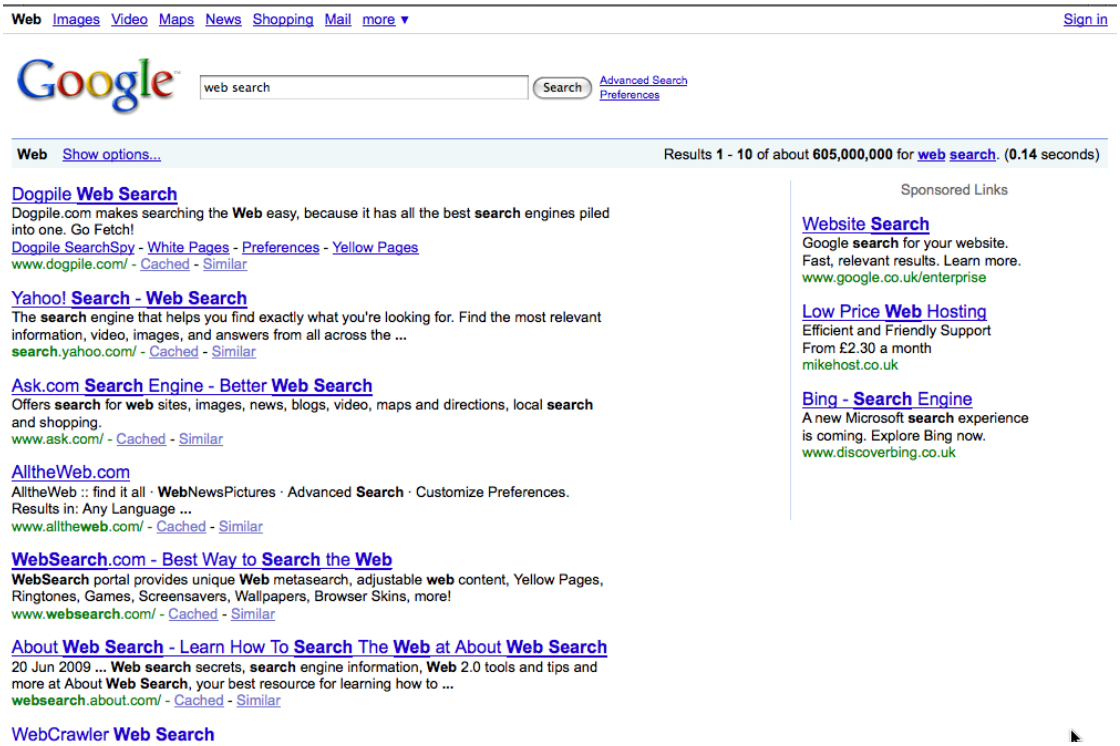


FIGURE 2.14: The core search interface provided by Google.

Google does provide additional features within their service. Searchers can use operators, although studies have shown that only around 2% of users do so (Beitzel et al., 2004), to block terms (using the - symbol), specify an exact ordered combination of terms (using quotation marks), and require specific terms to be included in results (using the

+ symbol). A ‘Similar Result’ link next to every result will return a set of results more closely related to the associated result, rather than the query. Typically, ‘Similar Results’ are other pages from the same website. Users with a Google account can see if they have visited pages before, and when, and remove results from, or promote results within, the results returned for a specific query. Google also provides spelling corrections, specific answers to mathematical formulae, and related queries. Some variations of their service, such as the shopping service[5] provide faceted classifications, as discussed below.

In the last few months, Google have announced some significantly different alternatives to their interface. Users can, in a relatively simple change, choose to see more than two lines of snippet text per result. Similarly, users can request to see sample images from the results. In a more extreme change, Google has provided the ‘Wonder Wheel’, as shown in Figure 2.15, which provides a topic-map style visualisation of related terms. Users can follow paths of these related terms, but by doing so, the queries are simply changed to these new terms, rather than refined by them. Following the term ‘Fruit’ from the term ‘Orange’ for example, simply returns results about fruit, rather than only fruit-related results about the term ‘Orange’.

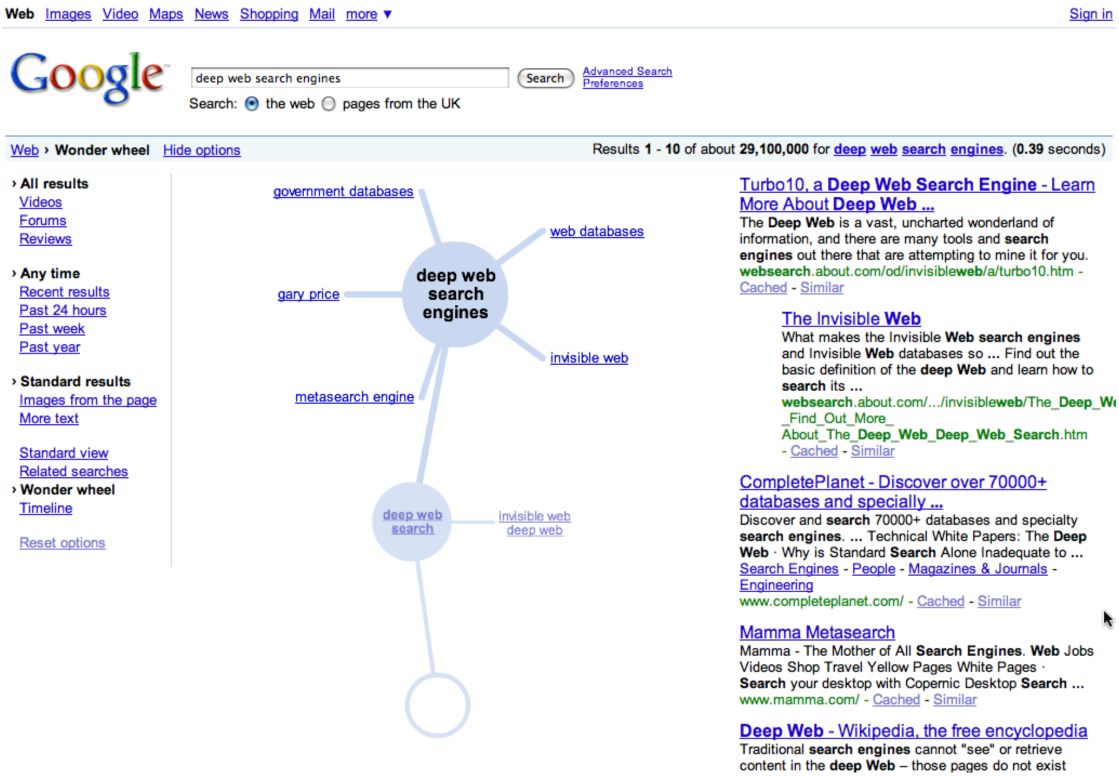


FIGURE 2.15: Google now provides ‘more options’, which includes alternative means of searching, such as the Wonder Wheel, which allows users to re-formulate queries by selecting related terms. Faded smaller circles represent previous queries.

[5]http://www.google.com/shopping - Google Shopping

2.2.2 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. One of the original ideas for this was demonstrated by search engine directories, such as Yahoo Directory [6] and Google's Directory[7] (shown in Figure 2.16). Such directory schemes, mostly out-shadowed by the power of keyword-search, provide a thematic hierarchical classification that can be used to find websites by their topics. Directory interfaces also suffered from the issues surrounding manual and automatic classification, as discussed above (Hearst, 2006).

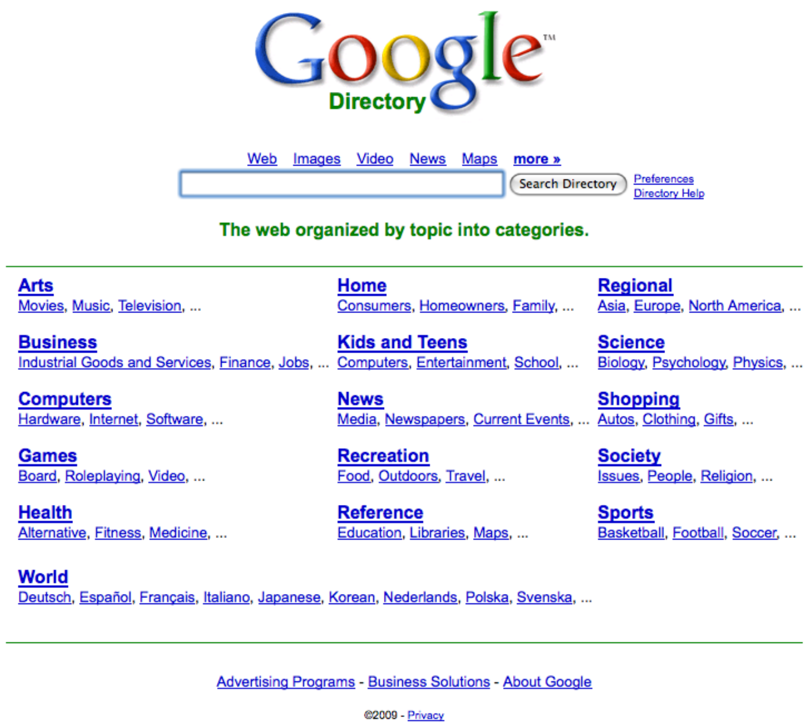


FIGURE 2.16: The Google directory provides a hierarchical classification scheme.

Where keyword search has become more popular than hierarchical classifications for endless unbounded collections like the web. Hierarchical classifications have proved successful for many constrained or managed collections. The Bureau of Labor Statistics[8] website, with the 2007 version (analysed in Chapter 4) shown in Figure 2.17, proudly foregrounds their hierarchical classification scheme on the front page.

Hierarchical classifications can often be limiting on their own, 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, optionally hierarchical, classifications are applied simultaneously to a dataset, so that users may use some or all of

[6]http://dir.yahoo.com - Yahoo! Directory
[7]http://directory.google.com/ - Google Directory
[8]http://www.bls.gov - U.S. Bureau of Labor Statistics

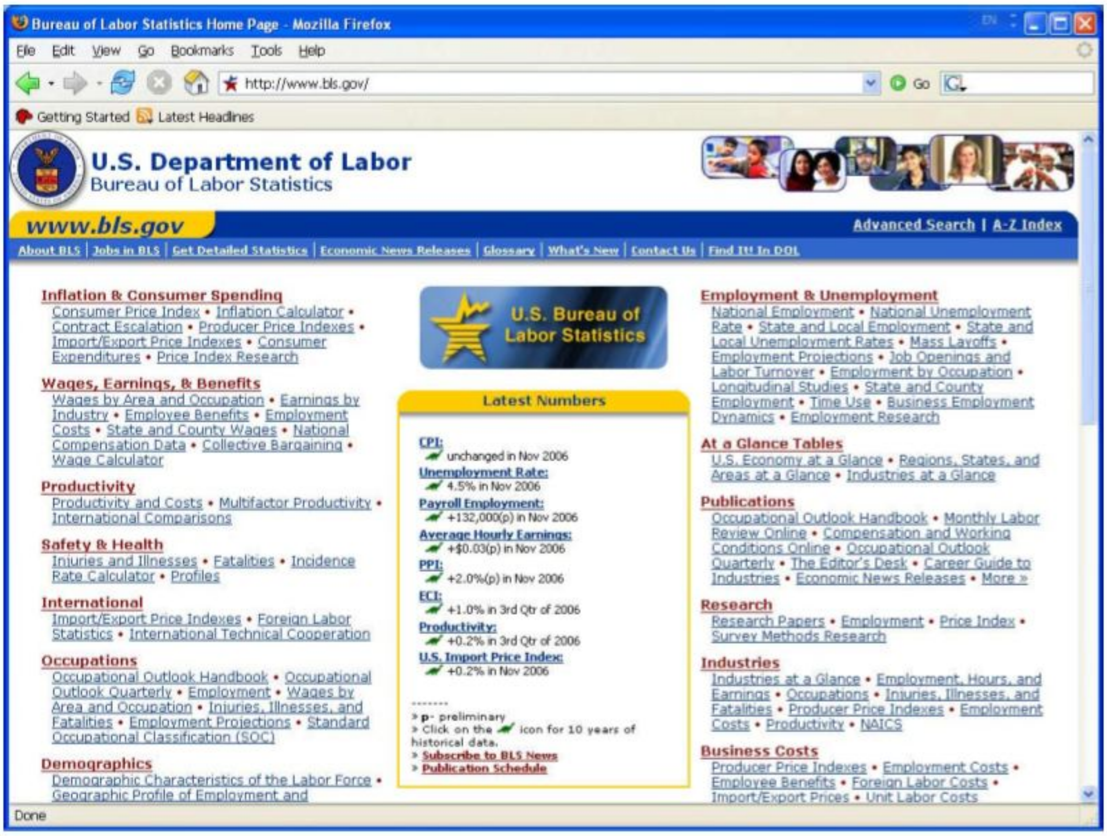


FIGURE 2.17: The Bureau of Labor Statistics website has its primary hierarchical faceted classification scheme on the front page (from 2007).

them to help find related information. The notion of such cross indexing was formed while creating reference structures for libraries (Ranganathan, 1960). Often, providing a faceted classification involves knowing the dataset in detail and carefully constructing the facets. We see this sort of multiple classification regularly on e-commerce sites like Walmart, eBay, and Borders Bookshop. Providing a faceted classification, however, not always possible with datasets that are continuously developing, like the web, although Kules et al. (2006) and, seperately, a commercial search engine called Exalead[9] have tried to identify facets that can apply generically to web documents.

McGuffin and schraefel (2004) discussed how faceted approaches affect search spaces by representing a simple model of three binary facets, as shown in Figure 2.18. As a user chooses one of the binary options from a facet (x, y, or z), the search space is reduced to the results only associated with that option of that facet. A second selection of a value in one of the remaining facets further reduces the result space to, in this model, the cubes that relate only to the selections made in the two facets. A final selection in the last facet has narrowed the space even further. Depending on the selections made a particular space is identified through the triangulation of the three facets.

[9]http://www.exalead.co.uk/search - Exalead: Choose a new search engine

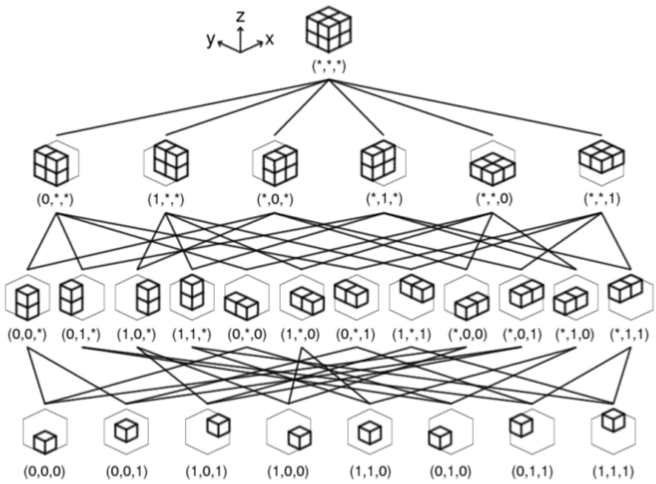


Figure 2.18: A small model of 3 binary facets, narrowing down a search space by subsequent selections made through one of the paths from top to bottom (McGuffin and schraefel, 2004)

2.2.3 Traditional Faceted Browsers

In the traditional design of faceted browsing, used by browsers such as Flamenco (Yee et al., 2003), shown in Figure 2.19, and the Relation Browser (Zhang and Marchionini, 2005), shown in Figure 2.20, and found in most online websites, facets are interdependent and each affects the other. For example, applying a constraint in one facet simply filters all of the remaining 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 of information. 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" (Lida et al., 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.

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

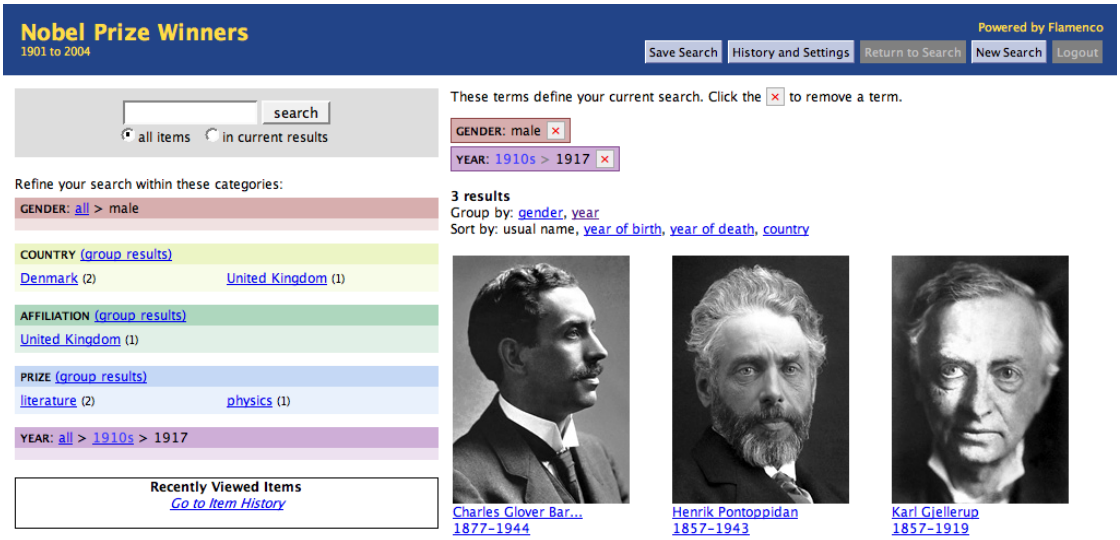


FIGURE 2.19: The Flamenco Browser, showing nobel prize winners with a faceted classification.

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 there, 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 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, however, is purely aesthetic and has no affect on the results produced. 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 should the user add that selection to the existing selections and, concurrently, the number of total target objects in the dataset associated with that label. 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.

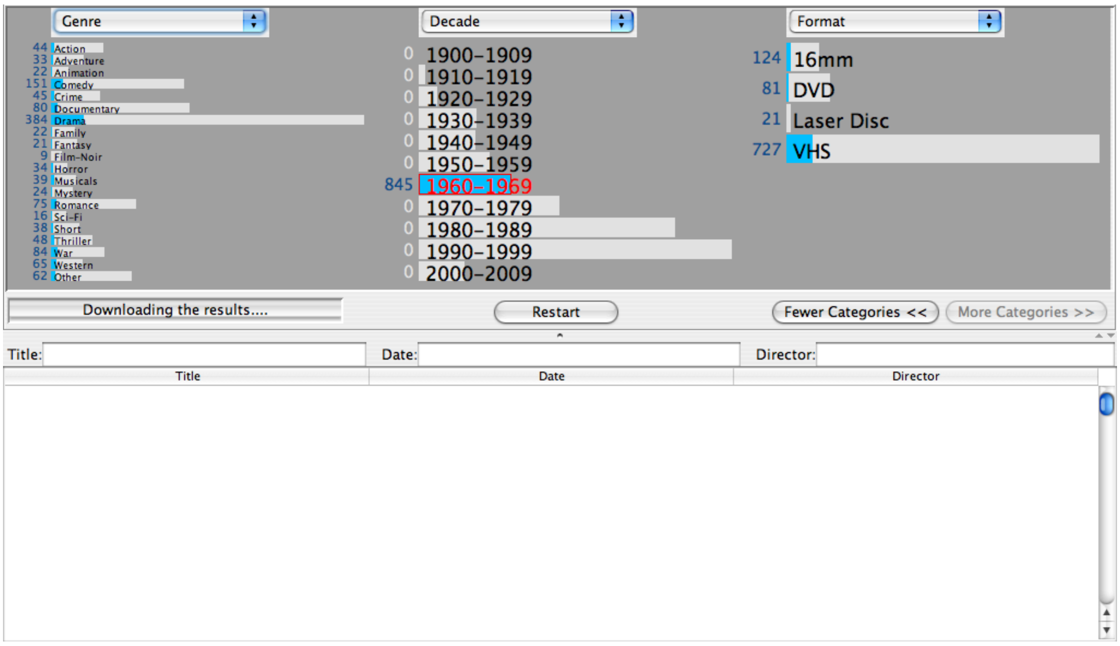


FIGURE 2.20: RB++, showing a movie archive with a faceted classification.

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.

Since this version of the relation browser, which is involved in two example applications of the Sii framework later in the thesis, Capra and Marchionini (2008) have released 'RB07', which advances their earlier design by adding keyword search, and the notion of 'facet clouds'. Similar to tag clouds, a facet cloud increases the size of labels within the facet according to the number of documents they are associated with. Further, RB07 takes a keyword-first view of faceted interaction, and so requires users to start by submitting a term query.

Many other faceted browsers appear online, and mainly in e-commerce situations. Endeca provide faceted search support for many high-profile vendors, including IBM, Walmart, and Ford. The service provided by Endeca, like most enterprise search platforms, is highly customisable, but an un-customised version of Endeca is studied as part of Sii's validation process in Chapter 4.

2.2.4 Advanced Forms of Faceted Browsing

An alternative approach is to provide a directional column-faceted browser like iTunes[10] or mSpace (schraefel et al., 2006), shown in Figure 2.21. 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. An advanced faceted example is described below, but both Wilson and schraefel (2008b) and Clarkson et al. (2009) discuss some of the variations that exist across different faceted interfaces.

FIGURE 2.21: The mSpace browser, showing a news video archive with a directional column-faceted classification.

mSpace presents facets as columns to create a hierarchy through the meta-data from left to right across the browser; called 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

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

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., 2008) 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 certain target objects without need for further refinement.

The final key section of the mSpace is the 'collection space'. This supports information triage (Marshall and Shipman III, 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.

Exhibit is another example of a more advanced faceted browser (Huynh et al., 2006), developed at MIT, which interacts with timeline and map visualisations. Exhibit provides some similar interactions as mSpace, including multiple selection and ordering effects on facets. The visual design of Exhibit, including the placement of facets, is more similar to

Flamenco. Consequently, order is supported according to the order of selections made, rather than the layout on the screen like mSpace. The key difference from Flamenco, therefore, is that used facets are not filtered or removed from the UI, so that the user can see the contexts of their previous selections and make multiple selections within the same facet. Whether this un-visualised ordering is harder to comprehend, or removes the need for users to understand the implications of order, has not been demonstrated so far.

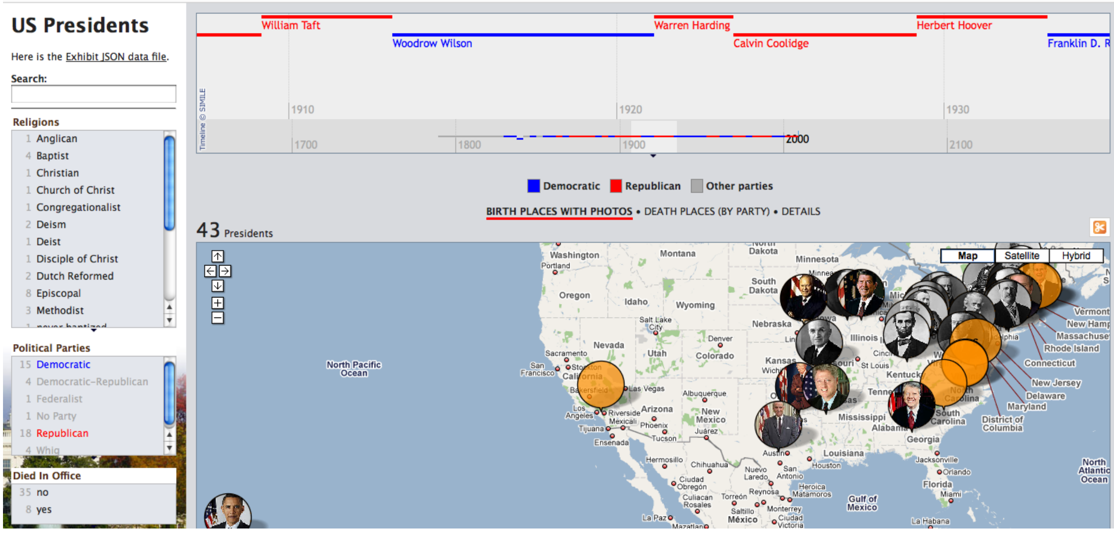


FIGURE 2.22: The Exhibit browser, providing faceted metadata about the US presidents, along with a map and timeline visualisation.

Like the majority of other faceted browsers, keyword search is also integrated into the exhibit browser, but rather than producing results that can be separately filtered by faceted selections, the keyword submitted is used to filter the facets. The result is that all the facets are reduced to showing the meta-data that relates to any records that contain a faceted index matching the term. Searching for democratic in the interface show in Figure 2.22, filters the political parties facet to show the democratic values only, and the other facets show any values that match the 19 democratic or democratic-replublican presidents. Highlighting of related-metadata, when selecting individual results on the map or timeline, however, is not provided.

2.2.5 Collaborative Search

Although the majority of Information Retrieval (IR) and Information Seeking (IS) systems have been designed for solitary use, recent research has shown that we collaborate on search activities with our colleagues, family, and friends, by asking for guidance, sharing links, and even dividing up tasks (Järvelin and Ingwersen, 2004; Morris, 2008; Twidale et al., 1997). Consequently, several novel search interfaces have been developed recently to support users in collaborating on shared search tasks (Amershi and

Morris, 2008; Morris and Horvitz, 2007b,a; Smeaton et al., 2006). Here, one particular Collaborative Information Seeking interface is noted: SearchTogether.

SearchTogether supports shared information goals by providing additional tools to the standard keyword search interface. First, users can see the search terms of other users, within a nominated known group. Second, users can communicate with an instant messaging style chat function. Third, users can rate and recommend websites to other users in the group. Finally, users can 'peak' at the pages currently being viewed by other users in the group. These, and a few more features, allow users to communicate and collaborate on problems. As the concept of searchers working together on a shared information need forms a fair deviation from the theory discussed at the beginning of this chapter, the interface, and a significant amount of further related work, is described in detail separately in Section 5.2.

2.2.6 Summary of Example Interface Designs

The review above is not, by far, a complete review of search interfaces, but instead provides an introduction to some search interfaces that a) are evaluated in this thesis, and b) demonstrate some of the advances from the fields of Exploratory Search and Human Computer Interaction and Information Retrieval. mSpace has often been used as a test-bed for research at Southampton University, where features that try to support exploration and learning (Wilson et al., 2008; schraefel et al., 2004) were implemented and evaluated. Typically these studies of mSpace followed a typical design process, of design, prototyping, implementation and evaluation. The work reported in this thesis has aimed specifically to bring our understanding of novel IS features earlier in the design process, rather than after building prototype implementations. In the next section, the design process is discussed in more detail.

2.3 The Interface Design Process

Effective procedures have been designed for software development, such as the waterfall model (Royce, 1970) and the V-model (BWB General Directive 250, 1992), which make sure that as many coding errors are found at different stages of development, including formal specification, module-level testing, and integration testing. Following these models, the Spiral model of software development (Boehm, 1986), was the first to explicitly highlight the benefits of iterative development phases, including multiple prototyping stages. Further, the Model-View-Controller framework provides guidance to software development and highlights that the user interface (*View*) should be separately designed from both the data structures (*Model*) and the functionality (*Controller*). In HCI, User-Centred Design (UCD) is a philosophy, which makes sure that users are at the focus of every stage of development (Norman and Draper, 1986). Typically, UCD evaluation is carried out in four main phases.

1) Requirements Gathering and Specification

The aim of this first phase, is to identify and organise what should be included in software and why. Although typical software development processes typically include the identification of functional requirements, the first phase of UCD focusses more on working practices and usage scenarios, for designing the effective user interface. Without clear user requirements, development time can be wasted and important functionality missed. Requirements Gathering (Holtzblatt and Beyer, 1995) is usually carried out first, using techniques such as interviews, questionnaires, and ethnography (Hammersley and Atkinson, 1995). Techniques such as personas (Cooper, 1999), which originally came from the field of marketing (Moore, 1991), and scenarios (Carroll, 1995) can provide focus for the discussion of exactly what the system is required to do, by examining the user and their potential environments.

2) Initial Prototyping of Designs

With the requirements in hand, rapid prototyping (Rettig, 1994), heuristic evaluation (Nielsen and Molich, 1990), and cognitive walkthroughs (Wharton et al., 1994) are examples of techniques that can be used to build and initially evaluate early interface designs. These processes are separate to the initial architecture and prototyping of software modules, but can be used to inform the connection between the 'View' and the 'Controller'. Further, the appropriate design of the user interface can produce requirements for the underlying data model too.

The aim of this phase is to identify the most appropriate interface design options that match the set of requirements generated in the previous phase. There are many ways in which specifications and requirements can be transformed into user interface designs, and so it is important to make sure that chosen options are going to be the most usable for the expected users, especially if they can be easily identified. Some techniques, including participatory design (Mumford and Henshall, 1978; Schuler and Namioka, 1993) and focus groups, can involve users at this early stage, and can identify some unexpected procedural problems.

3) Alpha/Beta Testing

Although software can be built from well-studied functional and usage specifications, and a strong design chosen from many potential options, no model or test on a prototype can fully identify all problems with a design. Early working implementations, therefore, are usually tested with potential users performing realistic tasks. A few final variations in design are often tested here, but in the form of semi-implemented applications that users can actually interact with. Evaluations often use measures such as accuracy and speed when completing tasks, to assess an interface. Other approaches, however, have looked at the ability to achieve a larger work-task, such as writing a report (Byström and Hansen, 2002), or how much users might have learned during a set period of time (Wilson et al., 2008). Such studies may be in laboratory environments, where variables are tightly controlled, or in environments where the software is expected to be used, often referred to as Field studies.

Higher fidelity, or more complete, implementations are often provided to a small set of users for sometimes a fixed period of time, with the caveat that the system being used is in a 'Beta' stage of development. Beta testing, therefore, is to study users for a longer period of time, as they use the software for self-motivated tasks, rather than lab-controlled tasks. Wilson and schraefel (2008c), for example, studied the use of a beta release of a video repository, over a month-long period. The known set of users were contacted on a regular basis, and their use of the software was logged. Beta testing is aimed at identifying unexpected usability issues that are not identified in the controlled user studies.

4) Ongoing Assessment

The final stage of UCD evaluation is to monitor on-going usage. Similar to Beta testing, on-going assessment aims to monitor realistic usage of a system, but after large-scale deployment. Usually such assessments are made with log data, but may involve asking users to fill out questionnaires. Although ongoing assessment can provide analysis of real usage behaviour, it is often at the cost of insight into the context of where, when and why users are using the system. Consequently, behavioural patterns are identified, but not necessarily explained.

Summary

The inspection framework described in this thesis is designed to support the early prototyping phase of development, and specifically for search systems, by rigourously and systematically assessing the types of search supported by the interface. The framework, for example, could be used alongside techniques such as rapid prototyping, cognitive walkthroughs, and heuristic evaluations, by providing a specialised focus on support for search. Sii may also be used, however, in the later design phases too, as it can be applied to both prototypes and working systems. Sii is carefully described in Chapter 3. One of the major concerns of the design and evaluation process, however, is to geneate effective and functional designs that are still simple and intuitive to use. Sii promotes additional functionality in a design, and so Section 5.1 discusses potential complementary methods that might help manage the trade off between functionality and simplicity. For now, the next section discusses some of the Usability Evaluation Methods (UEMs) mentioned above, along with a more focussed description of how UEMs can be classified.

2.4 Usability Evaluation Methods

The way that science has evaluated novel ideas has evolved dramatically over time, as well described by Shneiderman and Plaisant (2006). Several hundred years ago, the emphasis was on creating controlled experiments that can identify cause and effect, and measure significant differences in results. Although controlled experiments are still popular, the study of human factors in design has moved to more social and in-context research. In a recent analysis of UEMs, described below, it is clear that controlled user testing now forms only a small part of the range of methods available for evaluating user interface design.

COST294-MAUSE was a recent international and multi-institutional effort, reported by Law et al. (2009), aimed at better understanding the wide range of Usability Evaluation Methods (UEMs) that are available. The work was carried out over a four year period, and was split into four working groups (WGs). WG1, discussed in this section, was focussed on identifying, classifying, and cataloguing the available UEMs. WG2 focussed on evaluating and comparing these UEMs, which is discussed in Section 2.5. WG3 focused on the quality attributes considered by practitioners, and WG4 focussed on automatic UEMs. The following sections focus on the first two of these working groups, as they provided key insights into how UEMs can be related and compared. The structuring of the space of UEMs, here, provides context for the discussion of the new UEM being reported in this thesis in the subsequent chapters.

2.4.1 Classifying UEMs

In total, WG1 catalogued 39 UEMs using a single classification template. The classification template used considered many attributes, which are listed in the following four sets. Each bulleted classification field includes a short paraphrased description of the detail provided in the original larger report (Law et al., 2009). First, a set of identification fields were included, most of which allowed the classifiers to more qualitatively describe the UEM. Second, a set of method attributes were captured that relate to how, when, and why the UEMs are used. Third, the major advances and disadvantages were identified, usually in bulleted points. Finally, a set of attributes, estimated on likert scales between one and seven, were identified to represent the impact of the UEMs. These fields are shown in Table 2.5.

Many of the attributes chosen by WG1 are descriptive of the individual UEMs. The abstract and the history attributes, for example, provide some context about the origins of the UEMs, but do not provide much support for comparison. Attributes, such as the procedures, development stages, and scopes, however, provide us with the ability to find differences and similarities between UEMs. Further the impact attributes are quantitative measures that allow us to directly compare UEMs.

	Field	**Paraphrased Description**
Identification	Name	The name of the method
	Category	A category, described below
	Authors	The creators of the UEM
	Theoretical background	Any theory or models that provide the grounding of the model
	Abstract	A short abstract text of the UEM
	History	A summary of the significant stages
Usage Contexts	Function/goals	A summary of the intent of the UEM, such as what question it addresses
	Relationship to software engineering	In relation to ISO 9126
	Scope of GUIs	All, web, mobile, etc.
	Scope of sector	Specific domains, if any
	Applies to stage	Requirements, design, testing. See development process (ISO 12207)
	Procedures	Steps of applying the method
	Prerequisites/constraints	Comment on any constraints in using the UEM
	Research questions (unsolved)	Anything that still needs to be addressed
Pro/Con	Advantages	Description and source (empirical or author statements)
	Drawbacks	Description and source (empirical or author statements)
Impact	Use in industry	1-7, where 7 is high, with qualitative comments
	Acceptance by academia	1-7, where 7 is high, with qualitative comments
	Extensibility	1-7, where 7 is high, with qualitative comments
	Further comments	A space for any additional discussion if needed

Table 2.5: The classification scheme created by the MAUSE WG1 for cataloguing different Usability Evaluation Methods.

One particular field, discussed in the next subsection, is the categorisation scheme also created by WG1. As part of describing the categorisation scheme, several UEMs are briefly introduced. Several key UEMs, which are similar to the method described in this thesis, are then described in more detail. The full catalogue of UEMs, however, is significantly large that it was not even included in the 200 page final COST294-MAUSE report by Law et al. (2009). The full catalogue was reported separately by Scapin and Law (2007). The description of the Sii framework, in Chapter 3, will conclude by describing the new method, and other related UEMs, in terms of WG1's classifications and categorisations.

2.4.2 Categorising UEMs

One of the classification fields created by WG1 was a categorisation scheme, which was, in particular, used to described the breadth of techniques available and how they relate. WG1 defined methods as being either: a) Data Gathering and Modelling Methods (DGMM), b) User Interactions Evaluation Methods (UIEM), or c) Collaborative Methods (CM). These three types of UEMs, and their sub-categorisations, are described in the paragraphs below, and shown together in Table 2.4.2 along with example methods.

DGMM methods are used for gaining knowledge about users and their activities. Usually relating the the requirements and specification phase of UCD, DGMM includes Data Gathering (DG) methods, like surveys, cultural probes (Gaver et al., 1999), and Card

Categorisation	UEMs
Data Gathering and Modelling Methods	
Data Gathering	Surveys, Think Aloud, Card Sorting, Cultural Probes
Modelling Methods	Personas, GOMS, CPM-GOMS, KLM, scenarios, HTA
User Interactions Evaluation Methods	
Knowledge-based & Model-based	
Expert Evaluation	Expert Walkthrough
Document-based	Heuristic Evaluation, Ergonomic Criteria
Model-based	Cognitive Walkthrough, Abstract Task Inspection
Empirical Methods	User testing, ActA, Instant Data Analysis
Collaborative Methods	Focus Groups, Cooperative usability testing, Cooperative User Experience Research

Table 2.6: The categorisation scheme created by the MAUSE WG1 for classifying different Usability Evaluation Methods.

Sorts (Gaffney, 2000). DGMM also includes modelling methods (MM) for creating representations of users, or scenarios, or tasks. Modelling methods (MMs), therefore, include personas (Cooper, 1999), scenarios (Carroll, 1995), Hierarchical Task Analysis (HTA) (Diaper and Stanton, 2003), and the detailed GOMS method and variations (John and Kieras, 1996). HTA is a process designed to break down tasks clearly into a set of procedural steps. GOMS, standing for Goals, Operators, Methods, and Selection rules, provides a complex modelling scheme for known and specific tasks.

UIEM methods are designed to evaluate interactions with a design, and are broken up into either analytical Knowledge-based or Model-based Methods (KMbM) and Empirical Methods (EM). KMbM methods are analytical in that they are not based on real user interactions, but expected or estimated interactions provided by experts or designers. KMbM methods are further broken down into Expert-based, Document-based, and Model-based. Expert-based methods purely rely on the expertise of the evaluators and include Expert Walkthroughs (Følstad, 2007), where evaluators step through scenarios of use to identify usability problems. Document-based methods rely on guiding documents, such as Heuristic Evaluation (Nielsen and Molich, 1990) checklists, and Ergonomic Criteria (Bastien and Scapin, 1993). Model-based methods rely on theoretical models that are used to inspect the usabiltiy of interfaces. Model-based methods include the Cognitive Walkthough (Wharton et al., 1994), where experts step through scenarios of use repeated asking four questions regarding the intuitiveness of the interface design. Further, Sii described in this thesis, is a Model-based method. These types of methods that fall under the KMbM categorisation, of UIEM methods, are typically used in the second prototyping phase of the UCD evaluation process. Finally, the second broad type of UIEM methods is Empirical Methods (EM). EMs are typically the techniques used in the Alpha and Beta testing phases, involving studies of users performing tasks in laboratory environments.

CM methods are those that actively involve users during evaluation, usually performed as part of Participatory Design (Schuler and Namioka, 1993), including Focus Groups.

These categories of UEMs are re-used to discuss in the following sections, and in the description of the UEM being reported in this thesis, to provide context into how UEMs are designed.

2.4.3 Example Usability Evaluation Methods

While there are many UEMs that could be described in this subsection, including the all the methods that were catalogued by MAUSE's WG1, the methods discussed below relate, in some way, to the Sii framework. Some methods identify usability requirements, as discussed in the phases of evaluation above, but the methods described here are well known techniques that support the transition from requirements to effective designs and then implementations. Despite being an important topic, and discussed briefly above, the design of user studies is not discussed here.

It has been known for some time now that large amounts of resources (time, money, and work-hours) can be saved by finding usability problems in the earlier requirements and prototyping stages of the interface design process. Bossert, for example, showed that early evaluation of usability can reduce the development lifecycle by up to 50% (Bossert, 1991). Further, Lederer and Prasad (1991) showed that, while 63% of development projects over-run, the top four reasons were all related to unforeseen usability issues. Consequently, a number of techniques have been developed that allow early prototype designs of software to be assessed for their usability. While some methods, such as rapid low-level prototyping (Rettig, 1994), allow early designs to be evaluated by participants, Cognitive Walkthroughs (Wharton et al., 1994) and Heuristic Evaluations (Nielsen and Molich, 1990), known as inspection methods, allow interfaces to be evaluated by the designers alone.

Cognitive Walkthroughs (Wharton et al., 1994) allow evaluators to systematically step through example scenarios (Carroll, 1995) of use, usually produced in conjunction with Personas (Cooper, 1999) and Hierarchical Task Analysis (Diaper and Stanton, 2003). With each action, or *move*, required to achieve a task, four simple questions are systematically asked, including: 'does the user understand that a certain function is available?', and 'does the user receive feedback about their action?' This approach, which assumes users are novices or using the software for the first time, has been shown identify around 80% of usability problems and can be applied to early designs, and without the need of study participants. More recently a Streamlined Cognitive Walkthrough (Spencer, 2000), partly inspired by the earlier suggested Cognitive Jogthrough method (Rowley and Rhoades, 1992), was proposed, based upon the experiences of using the Cognitive Walkthrough method within real software development environments. The streamline method advocates, among several recommendations, focusing on key usage scenarios to save time, postponing the discussion of redesign until after the entire walkthrough evaluation is complete, and asking only two of the four questions at each step.

MAUSE categorises the Cognitive Walkthrough as Model-based evaluation under User Interactions Evaluation Methods. This category highlights that, based on a cognitive model of users, the aim of the method is to evaluate user interactions. The method models the experiences of first-time users of the software. The Cognitive Walkthrough can be used throughout the design process, but is designed to evaluate systems where the design, functionality, and even terminology used are well defined. This may include carefully specified paper prototypes, but is usually applied to higher-fidelity prototypes before empicial user testing. Procedurally, experts are expected to work separately, and then discuss their findings together. The key advantages lie in the speed and ease of use, without real users, and the drawbacks focus on the type of analysis (superficial, not functional), and the effect of evaluator skill on the results.

While Cognitive Walkthroughs focus on the learnability of an interface, in terms of how easily a user might learn how to use it, Heuristic Evaluations (HEs) focus on comparing an interface design with several recognized usability principles (Nielsen and Molich, 1990). Although often considered to be fairly informal, the process is widely used to make sure that simple usability issues, based around known principles, do not hinder the design and development process. Heuristics include: consistency, feedback, providing short cuts, clear error messages, and maintaining clear and natural language. MAUSE categorises HEs, as document-based rather than model-based, but is also under the bracket of User Interactions Evaluation Methods. Again, this means that HE uses a document to understand the user's ability to use the software. HE is designed for use by a single evaluator, although multiple evaluator opinions may strengthen the analysis. Like the Cognitive Walkthrough, HE is designed for evaluating high-fidelity prototypes, or cheaply improving finished products, but can be used with well thought-out low-fidelity prototypes. Like the CW, the benefits of HE relate to the speed and ease of use, but concerns relate to the quality of the results produced.

An HCI method that focuses on evaluating interfaces for how they support users in achieving their goals is GOMS (John and Kieras, 1996), which stands for Goals, Operations (user actions), Methods, and Selection rules. The aim is mainly to analyze users for their specific set of goals, the operations (or functionality) available in software for achieving these goals, the methods (or task structure) used to achieve the goals, and the selection criteria for choosing different methods of achieving the same goal. Consequently, MAUSE categorised GOMS as a Modelling Method, because mainly the method is designed to explicitly model users, goals, and tasks. To complete the evaluation, however, this method requires working software to be evaluated, so that time estimates for performing certain methods can be identified. The time estimates are created using the earlier Keystroke Level Model (Card et al., 1983), such as the model shown in Figure 2.23. GOMS was used, for example, to estimate that a new call-center workstation was less efficient, by 3%, for the tasks carried out by the staff Gray et al. (1992). The model found that the new software was too rigid and prevented staff from make efficient use of waiting time during calls.

While this sort of analysis cant be achieved through Cognitive Walkthroughs, the method is very complicated, does not consider different types of users (Preece et al., 2001), and does not evaluate the suitability of functionality in an interface, only the efficiency of implemented functions. Although the method, and its variations, were given high impact scores (5 and 6) for both academia and industry by the MAUSE project, the method has received some criticism about the cost of use and the benefit it provides. GOMS analyses the optimal actions taken by an expert user, with a specific set of functionality. Consequently, it does not allow for errors or optional interactions. Further, the results provided are typically represented by a difference in seconds for a single task. Consequently, the method has been used in industry more, where designers are building systems for tasks that are repeated many times, like the call centre mentioned above. The result has often been that GOMS does not often fit well into working practices, a topic discussed in the next section. Recent efforts from the original GOMS analysis have been to automate GOMS and incorporate it in software specification tools John et al. (2002) and story-boarding software (John et al., 2009).

Operator sequence

```
1. Initiate the deletion (decide to do the task) M
2. find the file icon M
3. point to file icon P
4. press and hold mouse button B
5. verify that the icon is reverse-video M
6. find the trash can icon  M
7. drag file icon to trash can icon  P
8. verify that the trash can icon is reverse-video M
9. release mouse button B
10. verify that the trash can icon is bulging M
11. find the original window M
12. point to original window P
```

Total time = **3P** + **2B** + **7M** = 11.9 sec

FIGURE 2.23: An example Keystroke Level Model analysis from Kieras (2001).

One of the more recent efforts in UEM design, was the RITE-method (Rapid Iterative Testing and Evaluation Medlock et al., 2002). Although not catalogued by the MAUSE project, the focus of the RITE-method is on rapid iterative testing. By performing regular, but small, user studies, with implementation changes in between, the RITE-method can plot progress and reduction of usability issues over time. Medlock considers this to be especially important, as it allows designers to fix usability problems and then make sure that the fixes work as expected and do not create knock-on usability problems. The RITE-method would be categorised under Empirical Methods, according to the MAUSE scheme, as it is based on studying real usage of a working system. A team of evaluators, designers, and participants are required, but the method can rapidly discover many types of usability issues. In particular, Medlock suggests that there are several reasons why the RITE-method can lead to successful evaluations:

- Medlock recommends that the project manager, and thus the decision maker (time and cost of redevelopment) should be present in studies
- using the principles of discount usability methods, small studies of a small number of participants identifies the key usability issues quickly
- task identification should be kept simple and quick, by keeping to the main uses of the software
- Medlock recommends that the evaluator should familiar with product and its aims
- Medlock recommends that the evaluator should know about implementation and what is possible to fix
- performing implementation fixes quickly and before studies continue means that improvement can be measured
- Medlock advises that software should be designed in such a way that such changes can be made quickly

Many of these suggestions relate to relationships between designers and evaluations, and to the efficient communication of ideas. These issues also, therefore, relate closely to the methods of evaluating UEMs that were agreed on by WG2 of the MAUSE project. The ways in which UEMs can be evaluated are discussed in the next section.

2.5 Evaluating Usability Evaluation Methods

The creation of UEMs has naturally caused many researchers and practitioners to ask which methods are 'best' or whether one method is 'better' than another. The words 'best' and 'better' are in quotes here as the majority of comparisons or evaluations of UEMs have fallen short of answering these questions.

Criticising the Evaluation of Evaluation Methods

One seminal discourse on this topic was captured in a special issue of the Human Computer Interaction journal, which through a process of critique, response, and reflection, highlighted exactly the significant challenges in comparing UEMs. Initially, Gray and Salzman (1998) critiqued five comparisons of UEM methods, for their ability to identify causal factors. Jeffries et al. (1991), for example, performed a comparison of Heuristic Evaluation, Cognitive Walkthroughs, guidelines, and user testing. Similarly, Gray and Salzman commented on a study comparing KLM, heuristics, and user testing by Nielsen and Phillips (1993). Each of the five comparisons has significantly influenced the HCI communities understanding of UEMs, but Gray and Salzman, as former industry practitioners, felt that the design of the studies undermined the credibility of the results.

In each of the five cases discussed, Gray and Salzman analysed the studies for four key validity factors identified by Cook and Campbell (1979). First, they considered the Cause-Effect validity issues of statistical significance, and internal validity, which is concerned with whether the notable effect is causal or based on a correlation. In particular, Gray and Salzman were concerned both that significance tests were often not performed, and the differences identified could not be attributed to a specific causal factor. Second, Gray and Salzman were concerned about construct validity, in that they felt many of the dependent and independent variables were not appropriately controlled. One study, for example, compared several evaluators independently performing one UEM, and then together performing a group-based UEM. In this case, Gray and Salzman argued that there should be several groups performing the group method. Finally, Gray and Salzman discussed issues of external validity, concerning the reasonable ability to generalise the results to larger populations. Their concern for many of the studies was that they were performed by graduate students, and then expected the results would also apply for experienced and expert practitioners. In summary of these issues, Gray and Salzman were concerned about the conclusions drawn by authors of the comparison papers, as each was affected by one or more of the validity issues described above.

The second article in the special issue was a collection of responses to Gray and Salzman, collated and edited by the editors of the special issue (Olson and Moran, 1998). The responses in these issues varied from defending methodologies, to suggesting alternative approaches, and also discussing the unfeasibility of achieving the ideologies highlighted by Gray and Salzman. Typically responses agreed that Gray and Salzman's suggestions represented the ideal, but several of the critiqued authors suggested that the best was achieved with the human resources and time available. The willingness for parent companies to allow staff so much time to take part in multiple lengthy evaluations was cited as an example. Several responses, however, took this issue further, suggesting that it was almost practically unfeasible to have enough participants, studies, and comparisons to reach statistical significance, for example. Alternative approaches including case studies were also suggested.

Finding Alternative Methods to Evaluate Evaluation Methods

In support of the responses to Gray and Salzman, WG2 of the COST294-MAUSE project started with similar aims to compare different UEMs. Despite the 4 year period, and contributions of many participants from multiple institutions across Europe, WG2 also failed to find a method of comparing different UEMs without facing similar validity issues. WG2 also suggested methods like case studies would help better understand the differences between UEMs, but concluded that no one measure could be used to say that one UEM was better than another. Several possible explanations were proposed. One was that UEMs find different types of errors (Blandford et al., 2008). Further, Furniss et al. (2008) highlighted that there are four dimensions on which you can evaluate UEMs: fitting with working practices, evaluator/designer relationships, identifying redesign options, and supporting other methods or secondary usability tasks such as reporting.

For this thesis, the four dimensions of working practices, identified by Furniss et al. (2008) are important to consider, as they determine the difference between the correctness and usefulness of a UEM. In Chapter 4, Sii is at least initially discussed according to each of these issues, although it is clear from the development of other methods that such understanding can take around 5-10 years (Blandford et al., 2008; John and Kieras, 1996; Nielsen, 1994; Spencer, 2000). Accuracy is important to demonstrate, in that the method identifies issues that relate to real usability issues. The extent to which methods can foster relationships between evaluators and designers has been identified as an important factor in the value of a UEM. As part of this, identifying and communicating redesign options is also important. In this light, a UEM may be considered effective if it can not only identify usability issues, but highlight how designs might be changed to remove them. Finally, fitting into practices, or desired practices, is also important. Effective UEMs have been associated with providing quick feedback, adaptive feedback for different audiences, supporting stakeholder interest, and encouraging team involvement.

Understanding these later issues of working practice somewhat requires initial uptake by evaluators, who are able to reflect on how methods worked in real usage scenarios. One of the only successful approaches to comparing UEMs was in the form of a survey of over 100 practitioner's experiences. Vredenburg et al. (2002) asked participants to identify the three key strengths and weaknesses of 13 different UEMs, and created the table shown in Figure 2.24. Again, however, we see in this comparison that practitioners still do not agree on the validity of most techniques.

2.6 Summary of Work Relating to the Evaluation of Search Interfaces

This chapter has summarised related work in several areas. The framework described in this thesis aims to utilise expertise from information seeking theory to develop a usability evaluation method. First, information seeking theory was discussed that models the way people search from different levels and contexts. Second, a set of information seeking interfaces were presented in relation to the aforementioned theories. The interfaces chosen represent a range of simple and complex interfaces, as well as those discussed throughout the remainder of the thesis. Third, the interface design process, especially from a user-centred perspective, was described to provide context as to when and how search interfaces are evaluated. With this process in hand, a classification scheme of usability evaluation methods was presented, in order to highlight the range of ways that interfaces can be evaluated during the design process. Finally, as this thesis describes

	Formal heuristic evaluation	Informal expert review	Iterative design	Prototype with no mention of user testing	Field Studies (including CI)	Usability evaluation	User interviews	Focus groups	Task analysis	Participatory design	User requirements analysis	Surveys	Card sorting
Input/Resources													
Low cost	9+	12+	+17	7+		6–	3–	4+		2–		3+	2+
Management buy-in				6+									
Availability of prototype			16–	3–					2+				
Availability of expertise	3–	4–											
Availability of information						3+	2+					3+	
Process													
Speed	10+	22+	24–	3–	17–	3–	2–	7+	7–			2–	2+
Participant cooperation					3+		2–			3+			
User involvement	7–	10–					4+			4+		2+	
Compatibility with practice					3+	3–							
Ease of execution									4–				
Versatility						4–							
Ease of documentation						3–	2–				2+		
Finding right users													
Outcome													
Validity/quality of	6+	7+	24+	6+	9+	8+	2+	14+	3+	5–		3–	
Results	10–	17–	20–	8–	8–	3–	4–	9–	8–				
Understanding context			19+		31+	7+	2+		10+	4+	3+		2+
Credibility of results					9–			5–				2–	
Impact/follow-up													
Timely Results													
Totals (+/-)	25/20	41/31	60/60	19/14	46/34	18/22	10/13	25/14	15/19	11/7	5/0	8/7	6/0

FIGURE 2.24: A summary of the 3 key strengths and weaknesses identified in a survey of over 100 HCI practitioners, to identify and analyse the most widely used UEMs (Vredenburg et al., 2002)

and validates a new usability evaluation method for search interfaces, the ways in which such methods can be evaluated were described in Section 2.5.

Clearly, the overwhelming abundance and variation of available information seeking interfaces indicates that evaluation must have been performed. The beginning of the chapter describes how search systems are often benchmarked against each other to demonstrate efficiency or accuracy. There are two ways, however, that the usability of information seeking interfaces have been evaluated. First, the range of UEMs described in Section 2.4 have been used to broadly assess whether the interfaces are appropriately designed according to generic usability issues. Second, user studies, which represent only a small portion of the available methods, have been tailored for search tasks (Kelly et al., 2009). Evaluators may time users, for example, for how long they take to find a piece of information (Dumais et al., 2001; Paek et al., 2004; Capra et al., 2007), or examine the amount of information that users can find through exploration and put into written reports (Brookes, 1980; Capra et al., 2007; Kammerer et al., 2009). No methods, however, tell us whether the right kinds of search are supported.

The nearest option to evaluate search interfaces specifically, but without user studies, is a set of search-oriented heuristics, proposed by Louis Rosenfield [11]. These 44 heuristics can be informally used to check 7 aspects of search interfaces: starting location, scope of content, query entry, retrieval results, query refinement, interactions with other aspects of a website, and what a user can do with the results they have found. In personal communications Louis suggests that he has used this when consulting on some projects, but has by no means validated or formalised the work since it was originally proposed in 2004. Further, these heuristics focus narrowly on the dominant keyword-search experience provided by services like Google, including broad examples like 'are stopwords removed?' and 'is there an advanced search option?'. While providing a good checklist, generated through practical experience, these search interface heuristics still do not answer the question: Does this interface support the right kinds of search?

The Sii framework described in this thesis is specifically designed to analyse search interfaces, or even potential designs, to evaluate the kinds of search that are being supported. The framework, described in detail in Chapter 3, is a usability evaluation method that can be used throughout the design process, and is carefully rooted in established information seeking theory.

[11] http://www.louisrosenfeld.com/home/bloug_archive/000290.html - Bloug: IA Heuristics for Search Systems

Chapter 3

The Sii Framework for Evaluating the Types of Search Supported by Search Interfaces

If there is a choice, test early, because more than 50% of all defects are usually introduced in the requirements stage alone.

Edward Kit, Founder of Software Development Technologies

After discussing a wide range of Information Seeking (IS) theory, IS interfaces, and usability evaluation, Chapter 2 concluded by identifying that while there are many ways to assess the generic usability of a search interface, there is no method that evaluates a design to make sure that it supports the right kinds of search and for the right kinds of searchers. Järvelin and Ingwersen (2004) also suggested that 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. More recently, a workshop on Information Seeking Support Systems, funded by the National Science Foundation (NSF) and summarised by Marchionini (2009), highlighted three continuing research challenges: (1) generating robust human-information interaction models; (2) building new tools, techniques, and services to support the full range of information seeking activities; and (3) developing techniques and methods to evaulate information seeking across communities, platforms, sources, and time.

This chapter describes the main product of this doctoral work, in the design and development of a novel analytical inspection framework designed especially for information seeking interfaces. Sii aims to bridge the gap between the human-computer interaction and information seeking communities, who are currently both evaluating search interfaces, but independently rather than in combination. Sii specifically assesses, as far as possible, the full range of possible search tasks that a search interface should support.

Further, it assesses these tactical options from the view of 16 searcher profiles, according to dimensions such as their existing knowledge and clarity of their perceived goals. In achieving the aims of the NSF workshop, Sii addresses the third strand of research into evaluation of information seeking activities, and supports others concurrent research into the second strand of developing new tools and techniques for supporting a wide range of searching behaviour. Further, Sii achieves these aims in a manner that can be used throughout the design process, by supporting the evaluation of both prior art and prototype systems.

Sii is described, across four main sections. First, Sii's application procedure and outputted analyses are described. Second, as a means to ground the description of using Sii, three increasingly complex example applications of the framework are presented. The wider use of Sii, as partially captured by these three examples, is also discussed. Third, the framework is further situated through the discussion of similar usability evaluation methods. Finally, before summarising the strengths and weaknesses of Sii, an online version of the Sii framework is introduced.

3.1 Sii: the Search interface inspector

Sii, also described by Wilson and schraefel (2007); Wilson et al. (2009b); Wilson and schraefel (2009c), combines three established models from Information Seeking, which were previously described in Chapter 2. First, the two finest-grained and well-defined levels of Bates' model of seeking strategies (Bates, 1990) are used to assess how many *moves* it takes to apply different *tactics* with an Information Seeking Interface (ISI). Second, the 32 specific *tactics* identified by Bates (1979a,b) are used to define the range of *tactics* that are considered when using the framework. Finally, the 16 different searcher (ISS) profiles, defined by Belkin et al. (1993), are used to simultaneously assess an ISI from many perspectives. The three models are combined with a novel mapping between the needs of different ISS profiles and the *tactics* they might choose to employ, described below in Section 3.1.3. The choice of models, and the way that they are used together is discussed further and validated in Chapter 4. Below, however, the Sii framework is described in more detail.

One way to imagine the Sii framework is that it looks at search interfaces through two sets of filters, shown in Figure 3.1. At any one time, a user is viewing an ISI from one of 16 user conditions, and sees it in terms of the *tactics* she can employ. Further, 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. Thus, each *tactic* receives a total score of how many *moves* it takes to use different parts of an ISI. In turn, when a user looks at the potential tactics through one of the 16 ISS profiles, they see how many *tactics* they can use and the average number of *moves* it would take to do so. The conjecture here is that any ISI should aim to support users in freely and easily performing any *tactic* they might need. Similarly, Bates (1989) suggested that *'the searcher with the widest range of search strategies available is the searcher with the greatest retrieval power'*. Other research methods have also aimed to minimise the number of *moves* required to perform a task (John and Kieras, 1996; Card et al., 1983). The Press On Framework (Thimbleby, 2007), for example, models software with state transition diagrams and aims to reduce the shortest path between any two states. The exact procedure of using the Sii framework online[1], and type of analyses it produces, are described below.

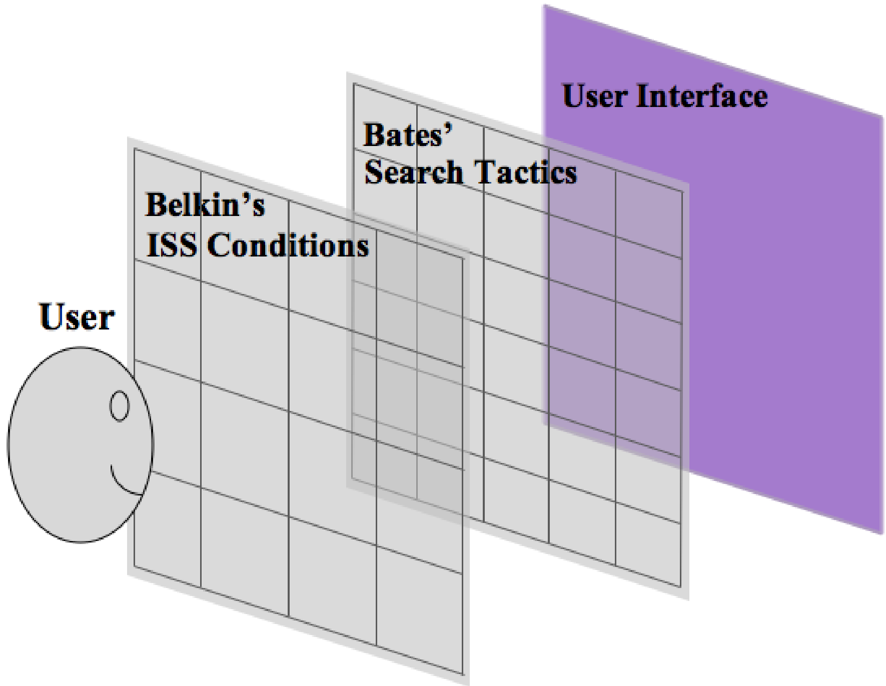


Figure 3.1: The interaction of the models within the evaluation framework. The parts of each layer act as a viewfinder on to the next layer, from Wilson et al. (2009b).

3.1.1 Evaluators and Preparation

Inspection methods, by definition (Nielsen, 1994), are designed for use by expert evaluators to make assessments of interfaces independent of users. Empirical methods, on the other hand, are those where evaluators study and measure the behaviour of human participants. As an inspection method, the preparation, procedure, and analysis is designed for ISI designers and evaluators, in a similar manner to checking that ISIs conform to requirements or heuristic guidelines. Although Sii can be applied by a single evaluator, as in Heuristic Evaluation, multiple evaluators may wish to work together or work seperately and compare their assessments. Issues relating to inter-evaluator consistency are discussed in Chapter 4.

[1]http://mspace.fm/sii - Sii: the Search interface inspector

Once the number of evaluators, if greater than one, has been chosen, there are two preparation tasks for evaluators to perform before beginning the procedure of analysing the interfaces. First, the evaluator must choose the interfaces being evaluated, which, as discussed later in the chapter, can range from prototype designs to fully implemented systems. Evaluators may wish to compare optional proposed designs, or compare a new design to an existing or popular system. The first example described in Section 3.2, analyses a single established and popular web search engine. The second example analyses two versions of a new interface feature that was added to the mSpace interface described above. The third example analyses three academic faceted browsers.

The second preparation task is to decide which features of the user interface are being evaluated. ISIs typically contain many features, such as a keyword search box, a results list, related searches, and spelling corrections. This second preparation task, therefore, is to explicitly decide and define which features to evaluate. It is especially important that multiple evaluators agree on these features and definitions before beginning the analysis procedure described below. It is rarely possible to objectively or automatically identify the different features available in an interface, but evaluators, however, may only be concerned with certain features of a design, or may agree on different granularities of features. In the third example discussed in this chapter, the functionality of the faceted classifications were separated into: a) individual selections, b) multiple selections within a facet, and c) de-selection, as the three interfaces varied in their implementations of these three aspects. If the concern of the evaluation is not on how the facets are implemented, then an alternative evaluation could consider these three aspects as a single interface feature and compare it against keyword search, for example. For consistency of evaluation, like with other methods (Kieras, 2001), it is more important that the chosen features are identified and the evaluators, if more than one, all agree on their definition and differences.

3.1.2 Data Entry Procedure for the Sii Framework

The process of entering data into the Sii framework, put simply, is to calculate the number of *moves* required to perform each *tactic*, with each feature of each interface. Consequently, one very small but key step 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 involved in the assessment. 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. In the example comparison of three browsers in Section 3.2.3, there are 12 interface features contributed by at least one of the designs. The third and final loop, Loop L3, is to repeat the main step for the 32 defined *tactics* (Bates, 1979a,b) that users may want to carry out. The labels L1-L3 will be used to reference these loops throughout the rest of the thesis. Combined together, we are assessing the ability to carry out each possible *tactic* (L1) with every feature (L2) of each interface being analysed (L3). The process of moving through these loops, therefore, causes the evaluator to rigourously consider how a every feature of each interface can be used to achieve each of the 32 tactics.

```
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, the notion of a *move* from Bates' model is used. A *move* can be either mental or physical. When utilising 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. As discussed in Chapter 2, several existing models and analytical methods include the notion of mental, or cognitive *moves*. In the Keystroke Level Model, mental *moves*, which Card et al. (1983) call 'Operators', can include cognitive actions, such as choosing a query term, or retrieving information from long term memory. Similarly, *moves* may include perceptual actions, such as reading, or scanning through a list of options. Both cognitive and perceptual moves, which Olson and Olson (1990) suggest may take on average 1.2 second, may not involve any interaction with the interface, but many direct interactions with software will be preceded by a mental *move*. Making a selection in a facet, for example, will often be preceded by both the perceptual scanning of options and the cognitive choice of which option to select. As with the identification of features in the second preparation task, and the advice given for counting keystroke level operators, choosing what is and is not a *move* is less important than maintaining a consistent level of granularity across evaluations.

There are several notes, or caveats, for the choice, and counting, of moves:

1. Repeat *moves* are ignored in counting, unless the repetition is required. 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. Most repeat *moves*, like when adding more than one keyword to a query, can be considered as Optional, as described in the next point.

2. Optional *moves* are ignored in counting. 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 size of dataset, for example, and instead measures the interface. Consequently, we ignore optional *moves* such as scrolling.

3. The counting of *moves* required to perform each tactic should be considered from the position that the *tactic* will be useful. If the tactic relates to monitoring ones progress, the count should assume the user has already begun their search. Consequently, the evaluator should not count submitting a query as part of CHECKing that a search is still on the right track. Similarly, VARYing the terms in a query assumes that a query has already been submitted. Thus, the counting of *moves* should start from choosing which term to change.

3.1.3 Internal Working of the Framework

In order to produce the analysis graphs described in the next subsection, Sii goes through several stages of internal processing. The processing, as well as the method for generating the graphs, is written in Javascript, but the algorithm is broken down into steps and described below. Javascript was chosen so that the tool could be used freely, across different computing platforms, on the web. Further, Javascript allows users to interact with the graphs, without having to install additional software plug-ins.

Step 1: Data Collection

First, the data entered by the evaluator is collected into a three-dimensional array, correlating to the three loops described above. Essentially, the three dimensional array constitutes an array, or table, of tables, one of which is partially represented by Table 3.1.3. The array, therefore, contains the data entered by the evaluator for each tactic, achievable by each part, of each interface. Zero is stored where the evaluator decided it was not possible to perform a tactic with an interface feature. A row of zeros, therefore, is stored where an interface does not include a feature. These zeros make it possible to perform basic sum and average calculations later in the internal processing algorithm.

Step 2: Inverting the non-zero data.

Once the three dimensional array has been generated, the data in the array are inverted to be a value up to a maximum of 1. Improving support for a *tactic* means reducing the number of *moves* down to two or, at best, one single action. Zero, however, represents the inability to perform a *tactic*. By inverting (calculating one divided by) the data, figures that are not zero then increase towards one, which represents the optimal support. Any additional *moves* reduce towards zero, which represents no support. The second result

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: Sii generates a javascript array of these tables, one of each interface being compared, where the number of *moves* is stored in the intersection of the features, listed down the left, and the 32 *tactics*, listed across the top.

of this inversion is that the graphs that are later produced have a natural feeling that a taller bar, or a higher value, represents stronger support.

Step 3: Calculating totals for the Tactic and Feature Graphs

For the first two graphs described in the next section, the rows and columns of each table in the inverted-data array are summed. By summing the columns of each ISI table, the total support provided for each *tactic* is calculated. By summing the rows of each ISI table, the total support contributed by each feature is calculated. The summed feature totals, and tactic totals, for each interface are captured in two bar graphs, described in the next subsection.

Step 4: Calculating values for the User Graph

One benefit of the Sii framework, is that it calculates support for different user profiles, such as those learning about a topic, or searching for things they have found previously. To produce this graph, a mapping was generated that specifies which *tactics* are designed to support each of the binary options of the four dimensions. Producing such a mapping is by no means trivial, as the relationships between *tactics* and dimension options are often many-to-many. Each *tactic* 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 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*. Unfortunately, as the mapping is between textual definitions created independently by separate academics, there is no way to objectively calculate a mapping. Part of the future work discussed in Chapter 6 is to follow the example of Kim (2009) and try to count the instances that certain behaviours occur. Studying a large and broad enough sample of an entire population's seeking behaviour is by no means a small task.

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., 2009b), shown in Table 3.2, was built through careful literature review, analysis, 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). It should be noted that the Validation process in Chapter 4 produced a refined mapping. The other main contributions from this period were the novel method of integrating the chosen models to produce a new metric, and the pilot analysis of three faceted browsers, which is described as an example of use in Section 3.2.3.

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. This mapping was later modified during the validation work described in Chapter 4, as shown in Table 4.2

The mapping within the Sii algorithm is represented as a matrix of ones and zeros that is used as a multiplier over the inverted-data tables, where a one represents that a *tactic* is important for a user profile. After applying the multiplier, the inverted data for all tactics that are not considered important for a particular user profile are reduced to zero. Consequently, only inverted values for *tactics* related to a user profile are used in sum and average calculations for the user profile graph.

3.1.4 Analyses Produced by the Framework

Sii produces three separate graphs by summarising the inverted figures as described in the internal workings above: Graph G1 provides an analysis of the support contributed by each feature of the interface; Graph G2 provides an analysis of the support provided for each type of seeking *tactic*; and Graph G3 provides 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 throughout the thesis, but their purpose is first discussed in more detail below.

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 that 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[2].

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 have 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 tactical support.

Graph G3 is the product of the novel mapping described above, which is used to summarise the support provided for each type of user profile. G3 could tell us, for example, 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. More specifically, G3 shows the *average* support of each *tactic* that is relevant to

[2]Understanding whether a certain design decision will end up improving user experience is discussed later during Validation in Chapter 4 and also considered further in the Future Work described in Chapter 6.

the user; the summed support of each relevant *tactic* is divided by the number of relevant *tactics*. As the 16 user types created by Belkin et al. are created by combinations of dimensions with two values, the graph is interpreted in patterns of halfs, quarters, eighths, 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 increased support for Learning (as their Goal) rather than Selecting. To support interpreting the graph, analysis should be performed in reference to Table 2.1, where each user type is defined[3]. The interpretation of this Graph G3 should become clearer with the examples below.

Together Graphs G1-G3 provide three complementary views over the ISIs being analysed: tactics, interface features, and user profiles. From the analyses we can tell that one design of a feature is providing better support for users, but we can also see that this support is particularly good for certain types of users and may have little effect on other searcher profiles. We can design features that aim fill the gaps in support for *tactics*, or try to support certain user profiles better. Finally, designers may want to consider ways that existing features can be extended in their design to provide a wider range of support, rather than introducing whole new interface features. In the next section, a series of example analyses and a range of scenarios are presented to ground the use of these graphs more clearly.

3.2 Example Analyses

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 compare multiple potential designs. Finally, three faceted browsers are compared to show how we can analyse larger and more complicated interfaces. It should be noted that these original pilot analyses were performed with the original mapping described above, and are revisited in Chapter 4 after the mapping was studied in more detail. Section 3.2.4 extends the discussion of the following specific examples by discussing the full range of scenarios that Sii can be used within.

[3]This table is shown next to each instance of Graph G3 throughout the remainder of the document.

3.2.1 Keyword Search

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, the familiar online experience of keyword search is analysed. The aim of carrying out such an analysis, therefore, is to understand the contribution and value of this commonly used feature. Google's keyword search implementation is chosen, as it represents the most popular instance. The value of loop L1 in this case is 1, as we are only considering Google's keyword search. 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, which without comparison to another design, tells us nothing.

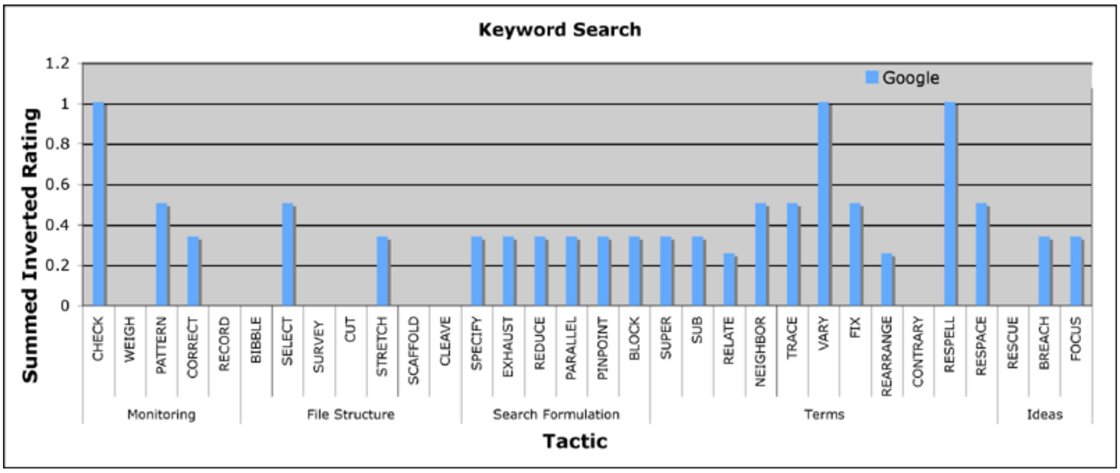
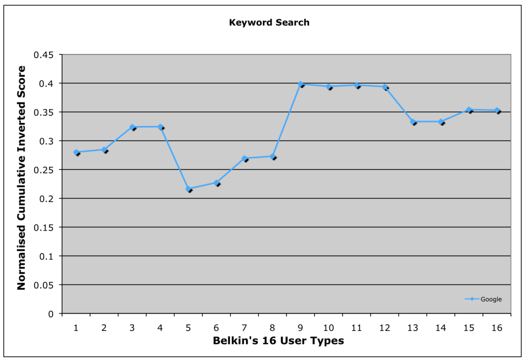


FIGURE 3.3: Graph G2 showing the support provided for 32 *tactics* (Bates, 1979b,a) by Google's keyword search, where taller bars represent stronger support for a *tactic*.

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. If Google did not maintain the current search in the text-box, users would be forced to return to the previous page to see what they had exactly searched for. VARY and RESPELL are well supported, in a single move, by the query expansions and the spelling suggestions that are provided by Google along with search results. The spelling corrections, however, could be considered as a separate feature to keyword search, which would make support for the full range of *tactics* by keyword search even more sparse. Notably, there are several known search *tactics* that are not supported at all by keyword search. 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. If Google did not keep a persistent search box on every page, however, then support for these tactics would be reduced, as the user would have to return to the search page to modify their query. One conclusion from G2, is that simply and naively adding keyword search to an interface does not provide complete support for searchers. Another conclusion, that is demonstrated well by Google, is that the support provided by Keyword Search is increased by the consistent and easily accessible search box *and* current query.



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

FIGURE 3.4: Graph G3 showing the support provided for 16 searcher profiles (Belkin et al., 1993, re-shown to the side), by Google's keyword search, where peaks represent stronger support. **NOTE:** this graph is revised using a refined mapping in Section 4.1.3.

From reading Graph G3 (Figure 3.4) in the patterns determined by layout of Table 2.1, shown side-by-side, 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 over recognising. The least supported user type is ISS5, who is hoping to find a specific item (Select in the Goal column), but does not know if one exists (Scan in the Method column), and is unable to clearly specify what they need (Recognize in the Mode column). Finally, ISS5 is looking for specific content in the results (Information in the Resource column) and so cannot tell from the search result pages alone if a result fulfils their need. From G3, we can also conclude that the majority of the support provided by keyword search is in tactics that allow the user to change and evolve their query while seeking a known target (ISS9-ISS12). The support for this is slightly more than the support for the ideal case that users simply specify and select a known item (ISS15), which is an activity that potentially requires less support from the interface. The support for ISS9-ISS12 over ISS15 seems slightly counter-intuitive, and is revisited in Chapter 4 in light of the modified mapping (used to produce G3).

3.2.2 The Backward Highlighting Technique

Wilson et al. (2008) report on a study that examined a new feature of mSpace called Backward Highlighting, shown in Figure 3.5. 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 meta-data and relationships to the user, certain relationships are not conveyed from right-to-left that would be shown by traditional faceted browsers, such as Flamenco and RB++. More detail on this problem is described by Wilson et al. (2008). 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 traditional and directional styles.

FIGURE 3.5: A prototype of Backward Highlighting was created on a set of mSpace columns and studied separately from the typical style of mSpace interface, shown in Figure 2.21. As mSpace columns filter left-to-right, Backward Highlighting highlights relationships backwards against the flow of filtering.

FIGURE 3.6: Bucket Highlighting, which copies highlighted items into a separate list at the top of each mSpace column, was prototyped in a similar stripped back interface and compared to Backward Highlighting.

In trying to determine the strongest implementation of Backward Highlighting, two designs were evaluated with user studies, where the second is referred to as Bucket Highlighting, shown in Figure 3.6. 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 or Bucket Highlighting technique alone. The move data (main step) for each tactic (L3) was entered for the one feature in both designs.

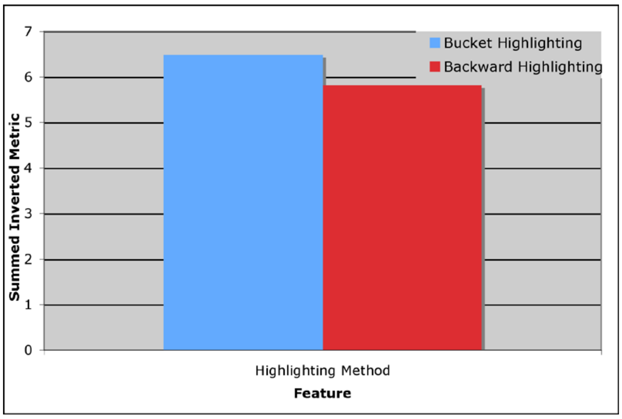


FIGURE 3.7: Graph G1 showing the support provided by the two different designs of Backward Highlighting (Wilson et al., 2008), where taller bars represent stronger support.

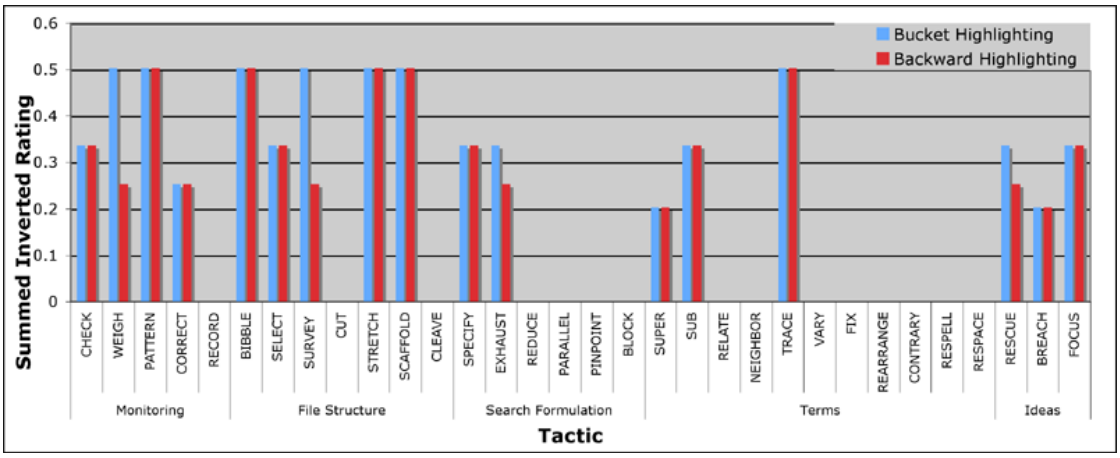
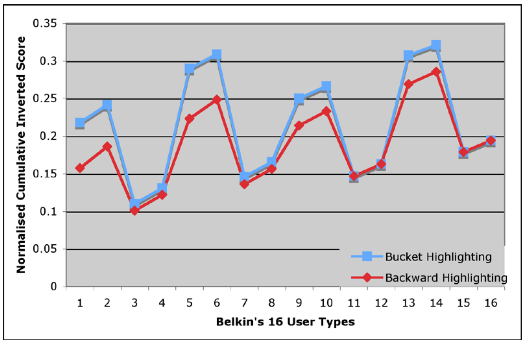


FIGURE 3.8: Graph G2 showing the support provided for 32 search *tactic* (Bates, 1979b,a), by the two designs of Backward Highlighting (Wilson et al., 2008), where taller bars represent stronger support.

As there is only one feature, but two designs, Graph G1 (Figure 3.7) only shows one feature (the new technique) and suggests that Bucket Highlighting design provides slightly more support for searchers. Graph G2, shown in Figure 3.8, shows us which tactics the Bucket Highlighting design provides additional support for. In particular, Bucket Highlighting 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. In general, however, we can see that the principle of Backward Highlighting, supports mainly monitoring, structure, and idea tactics.

Graph G3 tells us that backward highlighting technique, regardless of which design is used, least supports user type 3 and most supports user type 15. Referring to Table 2.1

shown to the side of the Graph G3, user type 3 is in the position where they are trying to learn about specific information by scanning for an unknown data source. 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. While some of these results are expected, such as the emphasis on recognition over specifying a need, it seems counter-intuitive that Backward Highlighting better supports users who are searching for a known target, than benefiting from learning about newly presented relationships. Graph G3 from this analysis is revisited in the light of a modified mapping generated by the validation process in Chapter 4.



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

FIGURE 3.9: Graph G3 showing the support provided for 16 searcher profiles (Belkin et al., 1993, re-shown to the side), by the designs of Backward Highlighting (Wilson et al., 2008), where peaks represent stronger support. **NOTE:** this graph is revised using a refined mapping in Section 4.2.1.

3.2.3 3 Faceted Browser Comparison

In previous work (Wilson and schraefel, 2007; Wilson et al., 2009b), the Sii framework was applied to three faceted browsers: mSpace (schraefel et al., 2006), RB++ (Zhang and Marchionini, 2005), and Flamenco (Yee et al., 2003). This example provides a much richer, but more complex analysis, compared to the two preceding evaluations. The three graphs provide a deep and rich insight into the strengths and weaknesses of each design. This richer example is reflected in the amount of discussion presented below. 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 (features of the interface) is 12, which is the unified list of the all features of the three browsers. The three browsers were described in detail in Section 2.2, and the 12 identified interface features are listed in Table 3.3. The value of L3, as ever, is 32 for the number of Bates' *tactics*.

#	Interface Element	Description
1	Favourites	This feature allows users to save facet values or results that they like. This does not include the ability to save a URL, which is possible in both mSpace and Flamenco, but instead looks at the ability to keep items as part of the user interface.
2	Preview Cues	Previews of information provided as the user hovers over a potential selection. In mSpace this provides an example piece of multi-media. In RB++, this previews the filtering effect on the values of each facet.
3	Make Selection	This is the ability to select a value in any facet as a filter on the remaining facets and/or the results.
4	Multiple Selection	This feature specifically describes the ability to select more than one value within a facet, which provides different results to selecting a single value in a facet.
5	Facet Organisation	The ability for users to control the layout and order of facets displayed in the interface.
6	Breadcrumb	An overview of the selections that have been made and the filters, therefore, that are in force.
7	Change Selection	The ability to unselect a facet value and select another from the same facet.
8	Keyword Search	The ability to search by keyword for results.
9	View Item	The ability to see a page specifically about a single result item, to find more detail.
10	Filtering	The ability to reduce the number of items in a list, such as the values in a facet, or the results in the results box.
11	Sorting	The ability to sort a list, such as the results, or the values in a facet.
12	NVIs	Numerical Value Indicators, that tell the user how many result items are associated with a particular facet value.

TABLE 3.3: The 12 interface features evaluated in the comparison of mSpace, RB++, and Flamenco. The list represents the union of the features of the interfaces, not the features that only occur in all three.

Graph G1 (Figure 3.10), 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++ respectively. It may also be noted that Flamenco has no preview cue, and thus the appropriate bar is absent from the graph. The ease of making multiple selections in a facet using 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 are 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 sorting is well supported by RB++ and Flamenco. In Flamenco, a user is able to group the results by

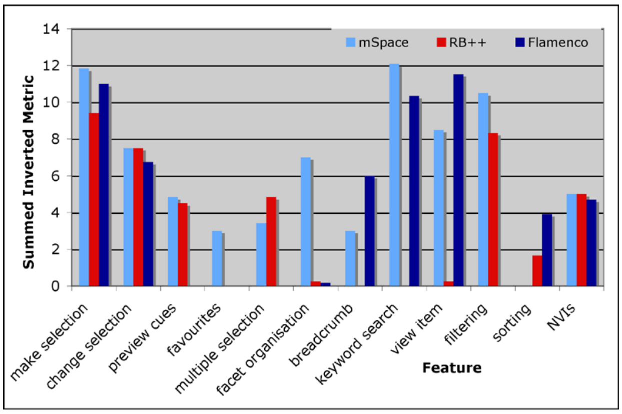


FIGURE 3.10: Graph G1 showing the support provided by the features of each faceted browser, where taller bars represent stronger support.

any of the facets in the system and provides the strongest implementation of a sorting method. Flamenco, however, does not support filtering. In mSpace, a 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.

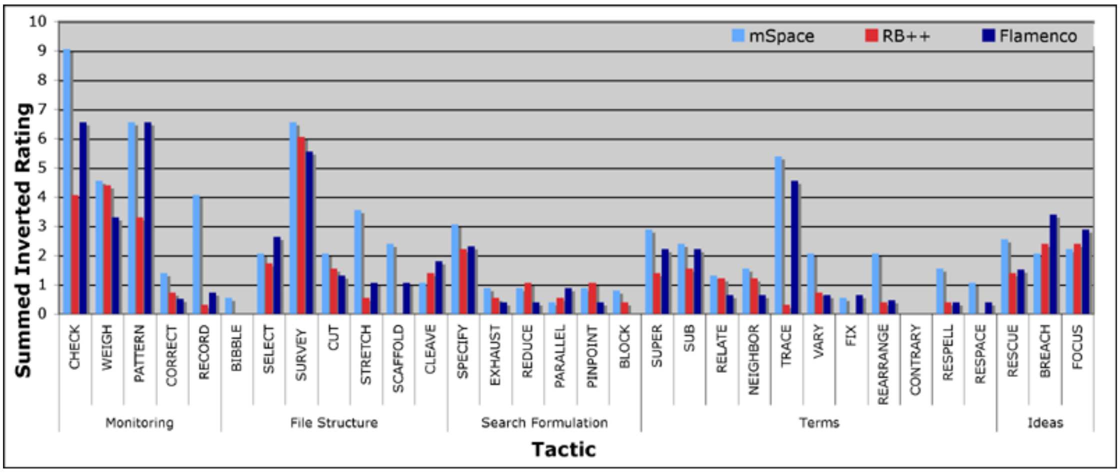


FIGURE 3.11: Graph G2 showing the support provided for 32 search *tactics* by each faceted browser, where taller bars represent stronger support.

Graph G2 (Figure 3.11) shows the support provided by each interface for each of the 32 known tactics. A number of observations can be drawn from Figure 3.11. First, each interface has a tall 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).

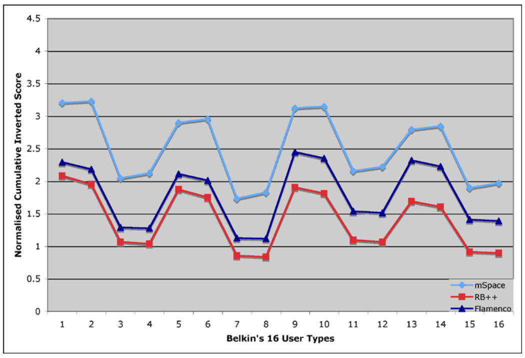
The first tactic, CHECK, has different levels of support in all three interfaces: this tactic is to see what actions have been made to corroborate them with the current aims. In RB++, although previous selections are highlighted in the interface, no representation of order is given and so a lower support 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 3.11).

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 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 not only bring up 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 is called a *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.



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

FIGURE 3.12: Graph G3 showing the support provided for 16 searcher profiles (Belkin et al., 1993, re-shown to the side), by the three faceted browsers, where peaks represent stronger support. **NOTE:** this graph is revised using a refined mapping in Section 4.1.4.

There are three distinct lines in Graph G3 (Figure 3.12), showing that mSpace provides the widest support for search. The separation of these three lines is perhaps not surprising given the consistent difference in support identified in Graph G2. Quite clearly, however, 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 line 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 the 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: 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 3.12) 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. Like the previous examples, this graph is re-examined in Chapter 4 after the user-tactic mapping is validated and subsequently modified.

3.2.4 Additional Scenarios of Use

The specific examples provided above provide a clear grounding for the type of analyses Sii can produce. These examples are only partially representative, however, of the full range of occasions that Sii can be used. Below, five separate usage scenarios are discussed that describe the breadth and variation of Sii's applicability.

1) Understanding Prior Art

The first keyword example described above shows how designers can individually assess the ways in which a particular design idea might support users. This can help designers to better understand existing features, like keyword search, or analyse a new idea in terms of how it might help users. For example, although it may be well known that many people find Google effective for finding information, a careful analysis of Google and all of its functionality, including 'similar pages', spelling corrections, and advanced queries, would reveal how each part contributes to the overall design. An analysis of Google, therefore, reveals which tactics, and thus which user types, Google's keyword search is particularly useful for, and what the effect of adding spelling suggestions has for different users. The different features can be added or removed one at a time to understand what makes the overall service. In this scenario, the first phase, and the transition into generating early prototypes, is supported. Evaluating specific and well known systems in this way can be, therefore, very informative when it comes to designing new ISIs.

2) Comparing Alternative Designs

The Backward Highlighting (Wilson et al., 2008) example above represents the way that Sii may be frequently used: in understanding alternative designs for a new ISI, or ISI feature. In the study, two alternative designs for backward highlighting were evaluated. The analysis identified the particular tactics and user-types that would be supported by the two different designs. Although in this case, the analysis was performed on a high-fidelity prototype, it could have just as easily been performed on low-fidelity paper prototypes. If Sii had been applied to such low-level prototypes alone, then the analysis would have provided early insight into what exactly the different benefits were for each design, and may have promoted a wider range of more informed alternative designs. Consequently, the inspection framework supports the second prototyping phase of the user centred design process, by allowing designers to compare and choose between viable ISI alternatives.

3) Informed Rapid Prototyping

In particular, this scenario emphasises one of the more pragmatic uses of the Sii framework. Given that the inspection framework can provide insight into even paper prototypes, it becomes very easy to quickly answer 'what-if' questions about the design of a feature. Sii can identify that certain features require convoluted sequences of steps to achieve a tactic, or that whole tactics are not supported, or that certain types of users may struggle. The designer can use what-if tests to quickly analyse, therefore, the effect caused by small changes to the interaction of a feature, or by the addition of a completely new feature. Any number of incremental changes can be designed and analysed, all requiring very little time, no users, and no additional expense. The inspection framework further supports the second prototyping phase, therefore, by allowing users to rapidly analyse design revisions.

4) Explaining User Study Results

Later in this thesis (Section 4.2.2), an analysis is performed, in which the framework analyses the support provided by three interfaces included in a very large user study, by Capra et al. (2007), to help explain why they may have been 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. In particular, the user types produced, according to the tasks set to participants, can be identified by this inspection framework and the analysis reveals that user-types created represent the three user-types who have almost equal support across the three interfaces compared. Other user-types, however, actually receive very different amounts of support across the three interfaces. A follow-up user study, however, involving different user-types with uneven support by the three interfaces may still allow the researchers to accept their original hypotheses. The framework, therefore, also supports the third alpha/beta testing phase of software evaluation, as it can be used in conjunction with, or in preparation for, user studies of working implementations.

5) Overcoming Difficult Comparisons

The final way that the framework may be used is in situations where a user study may be difficult to carry out, such as the the faceted browser comparison described in the previous section. Part of the original motivation for this doctoral work was to identify the strengths and weaknesses of the many different faceted browses that have been implemented with roughly the same requirements or originally identified problem scenarios. While many of the interfaces had been evaluated, each have been studied independently, and so it is hard to explain what is novel about their individual contributions. Their very different implementations, different datasets, diverse technological requirements, and often licensed code or poorly documented academic implementations, all make it very hard, expensive, and time consuming to perform a comparative user study. The study provided by Capra et al. (2007), for example, describes both month-long period spent with users as well as the months of preparation and analysis. Further, the results were unexpected, and their hypotheses rejected, with the authors identifying many further open questions about the benefits of the three search interfaces. As this framework does not require the interfaces to have the same data and does not require actual users, the analysis can be performed on existing search instances exactly as they stand. There is no need, therefore, to produce local instances of each system, find instances of each with exactly the same dataset, or co-ordinate with the their developers to produce instances with the same dataset, for multiple ISIs to be compared. Consequently, the Sii framework permitted the comparison of the same three ISIs in only a few hours that required months of work to compare in a user study. This scenario highlights that the framework can be used to make otherwise difficult comparisons at any stage of development. In fact, a new paper prototype could be compared to a fully implemented system that is already in widespread use on the internet.

3.3 Similar Evaluation Techniques

It is clear from both the examples and additional scenarios that Sii can be used in many situations, and can provide rich insights into the support provided by ISI designs, for different tactics and user profiles. Further, the insights and benefits that Sii provides are distinct from other usability evaluation methods that already exist. In this section, the relationship between the Sii framework and these other methods is discussed. To make these relationships explicit, Table 3.4 compares a few key techniques across several criteria. The majority of the criteria, listed down the side of Table 3.4 are paraphrased parts of the COST294-MAUSE Working Group 1 classification, discussed in Section 2.5. Futher, however, the table includes some issues listed by a) the RITE-method (Rapid Iterative Testing and Evaluation Medlock et al., 2002), and b) the types of insights that UEMs can produce, as defined by Blandford et al. (2008). The three Usability Evaluation Methods being compared to Sii are: Cognitive Walkthrough (CW), Heuristic Evaluation (HE), and GOMS, which are each described in Section 2.4. These methods are not exhaustive, but represent methods that are a) similar in some way to Sii, and b) had high impact scores in the MAUSE WG1 catalogue.

Informally, viewing Table 3.4, would suggest that Sii's underlying procedure and style is similar to the Cognitive Walkthrough, the requirements for performing Sii are similar to the Heuristic Evaluation, and the outcomes are similar, but less narrowly-focused, to those generated by GOMS. Novelly, however, Sii evaluates search interfaces from multiple user perspectives and contexts, where as the other methods evaluate interfaces from the view of novice or expert users only. Further, as the Sii framework is not

Criteria	**CW**	**HE**	**GOMS**	**Sii**
MAUSE Category	Model-based Interactions Evaluation	Document-based Interactions Model	User Modelling Method	Model-based Interactions Evaluation
Intent of Method	Check that an interface is learnable	Check that the interface is learnable	Model the goals and tasks of users	Check for appropriate functionality
Method Expertise Required	Medium	Low	High	Low
Type of User Modelled	Novice	Novice	Expert	Novice to Expert
Procedure	Step through scenarios of use	Step through checklist	Identify and model scenarios of use	Step through fixed scenarios
Prerequisites	Scenarios	None	Task Analysis	None
Level of implementation	Hi-Fidelity Prototype	High-Fidelity Prototype	Complete	Low-Fidelity Prototype
Scope of interfaces	Any	Any	Any	Search Interfaces
# of Evaluators	2+	1+	2+	1+
Timeframe for Evaluation (per UI)	12h	2h	24h	2h
Downstream Utility Assessments	Repeat CW	Repeat HE	Modify Model	Modify Model
Types of Error	User Misconceptions	User Misconceptions	System-design	System-design and Use in Context

Table 3.4: Sii is compared to three other key and closely related methods across multiple criteria: Cognitive Walkthroughs (CW), Heuristic Evaluation (HE), and GOMS (Goals, Operators, Methods and Selection Rules). Table cells are coloured gray to highlight similarities between Sii and other methods.

dependant on chosen terminology or layouts, for example, it can be applied to lower-fidelty prototypes than the other methods. These benefits listed, however, were also the aims that motivated the design of Sii. Sii was designed to be an inspection method, like those listed in Table 3.4, so that it could be applied in similar circumstances to Heuristic Evaluations.

It should be noted that Sii is the only method to cover a limited range of interfaces. Specialised but limited scope, however, is not uncommon in UEM methods. SUE is a method that is designed for Hypermedia applications (Costabile et al., 1997; De Angeli et al., 2000, 2003), which is based on a model-based evaluation technique called Abstract Tasks. Futher, many variations of the Heuristic Evaluation method have been produced, such as for Ambient Displays (Mankoff et al., 2003) or mobile devices (Bertini et al., 2009). More importantly, however, it is this speciality for search that means, by using Sii, the kind of results produced by GOMS can be generated using a similar procedure to Cognitive Walkthrough, and with the speeds and costs similar to Heuristic Evaluation. Further, it is this speciality that means a range of user perspectives can be considered. Finally, Chapter 2 has already emphasised the importance of getting search right, which further calls for a more focused specialised evaluation method.

Table 3.4 shows that the types of results produced by Sii are similar to the findings discovered by the GOMS technique. The Sii inspection framework presented in this thesis, however, is much more lightweight than GOMS, can be applied at earlier stages of design, provides an overview over entire interfaces, and assesses the functionality for different types of searchers. While GOMS provides an analysis of how fast expert users can achieve a given task with an interface, Sii assess how much support searchers are provided with for general search tasks, where support constitutes flexibility of actions and short sequences of steps. It is up to the evaluators, when using GOMS however, to choose which, and how many, tasks to evaluate. The analyses provided by Sii, by default, assess ISIs for a set of known tactics, which are similar to small tasks that can be used as needed. Sii furthers this analysis by then considering the need for applying such tactics by different user profiles. Consequently, by applying Sii, which takes much less time than a GOMS analysis, evaluators get a much broader view. Finally, instead of trying to decide which interface is fastest to use, which has been considered as questionably useful by lin, Sii produces a measure that correlates to a level of support. Consequently, Sii can easily determine that certain designs provide broader and better support for a wider range of tactics and user profiles.

3.4 The Sii Website: a Tool Available Online

Although the original analyses presented above were created within a spreadsheet, an online tool[4], shown in Figure 3.13 has been built to allow evaluators to apply the Sii framework to their own designs. The website, built primarily in javascript and using the DojoX Charting plugin[5], provides interactive forms of the graphs, as shown in Figure 3.14, which can be made full screen for closer inspection. The steps of applying the framework, including the preparation tasks, are supported by a series of online forms. The exact process of using the Sii website is not described in detail here, but included as Appendix C at the end of the document. Each of the examples discussed in this dissertation, however, are available to view online. Further, the Sii website was used to demonstrate analyses to participants of both the case and pilot studies described in Section 4.3.

The website provides the means for practitioners to view, temporarily modify, and discuss analyses. Further, the tool is free for practitioners to register and use for their own analyses. The development of the tool, however, is on-going, and part of the focus of the future work discussed in Chapter 6. While fully functional, additional work is required to make it more accessible and intuitive, and part of the validation discussed in the next chapter provides insights into way participants reacted to seeing and using the online framework.

[4]http://mspace.fm/sii

[5]http://docs.dojocampus.org/dojox/charting/

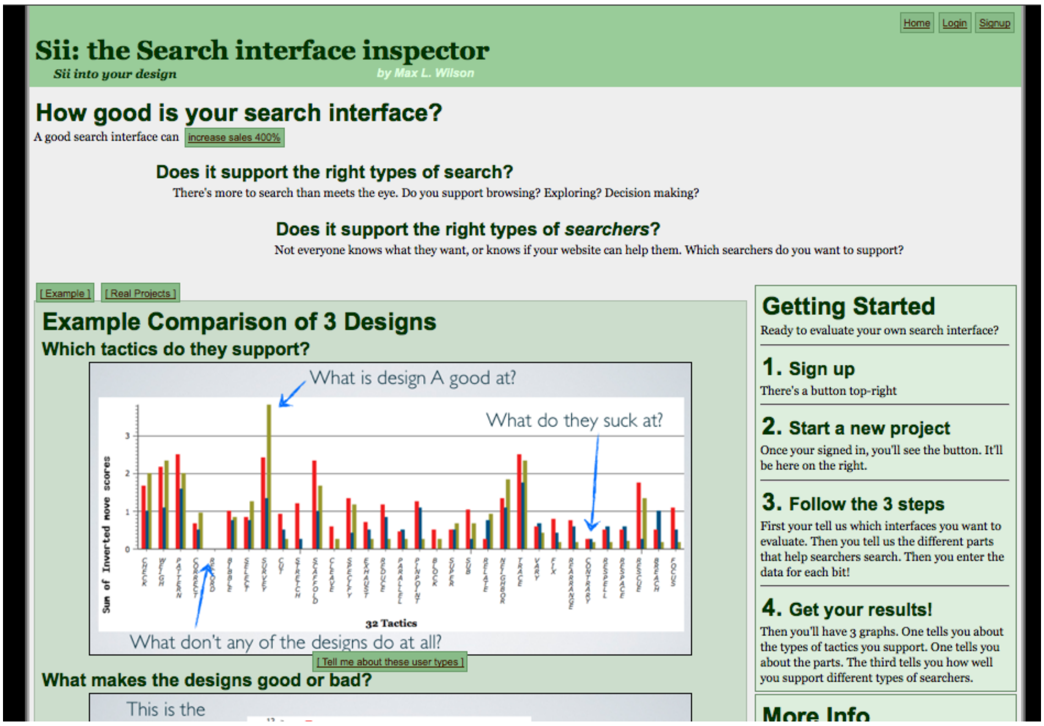


FIGURE 3.13: Evaluators create a project, with a description and motivation for the evaluation.

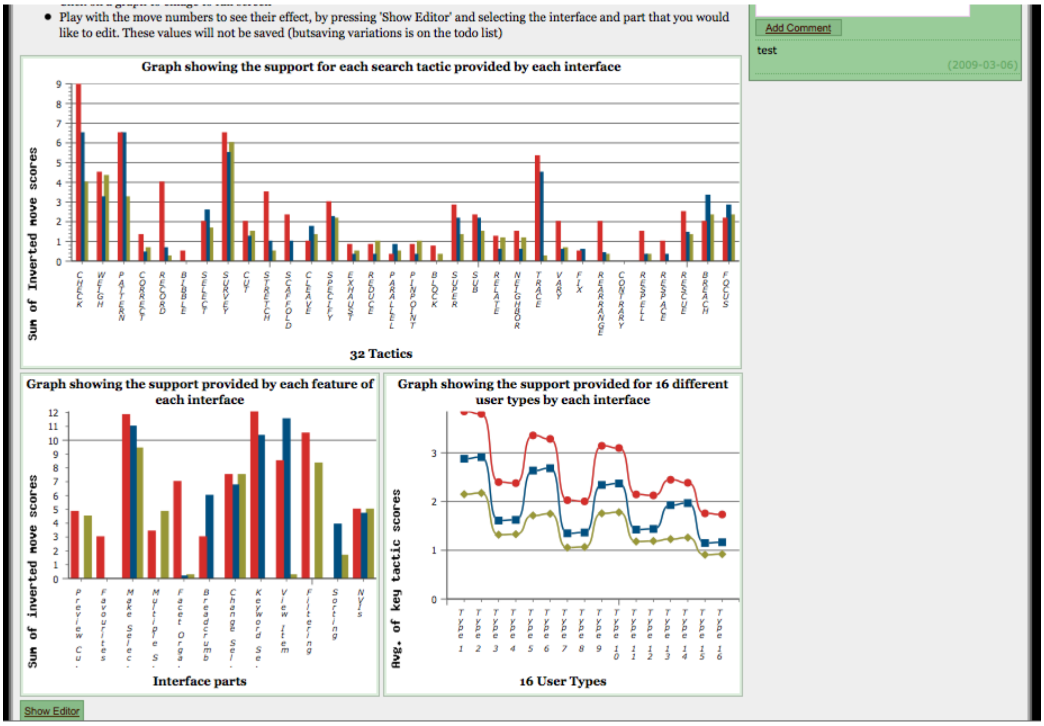


FIGURE 3.14: Evaluators are provided with the three graphs through which they can analyse the strengths and weakness from designs. G2 is shown on top, as it requires a large amount of horizontal space. G1 and G3 are shown side-by-side underneath G2. The evaluators, and anyone who views the analyses, can discuss the results on the right hand side. Evaluators can temporarily dit the data, using the button under the graphs, which updates the graph in real time.

3.5 Summary of Framework

In this chapter, an inspection framework, especially designed for search interfaces, has been described. The framework systematically analyses the different interactive features and functions of search interfaces, for their support for known Information Seeking and Retrieval tactics. By summarising this support, and modelling the tactics typically used by different user types, multiple searcher profiles can be considered. Consequently, Sii allows designers to discover the strengths and weaknesses of different design approaches, without the need of human participants, and in very little time.

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, CA, and Chapel Hill, NC. To compare these interfaces in a fair user study, in the style of 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, in order to have control over the interfaces as an independent variable, 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. (2009b), could provide the means to revive the interactive streams of the TREC conferences (Harman, 1997) that have stopped because of such barriers to controlling independent variables.

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. Once familiar with Sii, it only takes a couple of hours per ISI to apply the framework and interpret the graphs.

The Sii 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 Information Seeking models that make up Sii's 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 performed user studies.

Chapter 4

Validating the Sii Framework

The searcher with the widest range of search strategies available is the searcher with the greatest retrieval power.

Marcia Bates, Professor VI Emerita at University of California, Los Angeles

Chapter 3 described a new and novel Usability Evaluation Method (UEM), called Sii, designed to provide a primarily functional analysis of how Information Seeking Interfaces (ISIs) support different search tactics and searcher profiles. While previous research has shown the limited benefits in proving that one method might be better or more accurate than other methods[1], this chapter validates the Sii framework by a) reviewing decisions made during its development, b) demonstrating its accuracy in relation to empirical studies, and c) assessing how it fits within established working practices. Although it is not yet possible to determine the impact and acceptance of the Sii method[2], the focus of this chapter is to demonstrate a validation of Sii so that practitioners, whether from academia or industry, can use the new evaluation method with confidence. These validations further aim to answer the research questions listed in the Introduction.

In order to begin validating the Sii framework, Section 4.1 reviews the decisions made when choosing the theoretical models used and when generating the mapping that unites them within the one UEM. Chapter 2 discussed much of the seeking theory available from research literature, and so Section 4.1 begins by discussing the appropriateness of the chosen models and those that were not used. The section concludes with a study of the mapping used to integrate the chosen models, in which six judges independently generated a mapping between the models. The six mappings were compared with the

[1]Section 2.5 reviews the history of evaluating usability evaluation methods, describing some key limitations discovered in directly comparing them

[2]Chapter 6 discusses the kind of timeframes that are required to estimate the impact that methods have on academia and industry, which may be approximately 5 to 10 years after publication.

original mapping and discrepancies discussed, leading to the generation of a revised mapping.

Section 4.2 provides the key part of the validation process discussed in this chapter, by comparing the analyses generated by the framework with the results of empirical user studies. Inspection UEMs, including the Sii framework, aim to identify and remove the usability problems that would otherwise be discovered later by participants of expensive user studies, or even by users of deployed software. Consequently, Section 4.2 demonstrates Sii's capability in correctly identifying strengths and weaknesses of designs that correlate with the findings of empirical research. In doing so, the section also shows that the revised mapping created in the previous section provides more accurate analyses of ISIs than the original mapping.

Finally, to assess how Sii might fit within the working practices, the four contexts of usability evaluations presented by Furniss et al. (2008) are discussed within three short case studies. While, in all but one case, analyses were not performed by the participating experts, the potential of the framework to fit into existing practices, facilitate communication, and foster relationships was discussed with practitioners in the contexts of real projects. Further, several pilot studies are presented that provide insight into the practicalities of applying the Sii framework and interpreting the graphs. The results of these preliminary studies partly inform the future work discussed in Chapter 6.

4.1 Validating the Structure of the Framework

There are two aspects of the framework's structure that can have an affect on Sii's ability to correctly analyse search ISIs. First, three information seeking models were chosen from the abundance of search theory available. As the reasoning for this choice of models was not included in the description of the Sii framework, this section first discusses the reasons why these models were chosen. Second, to combine these models within one framework, a mapping was created between the tactics identified by Bates (1979b,a) and the user profiles identified by Belkin et al. (1993). Consequently, this section also reports on an evaluation of the mapping produced.

4.1.1 Validating the Choice of Models Used in the Framework

One of the first and most important steps in validating the framework is to be confident in the models chosen to produce the analysis. The notion that *tactics* are made up of physical and mental *moves*, or actions, performed by users is widely accepted and included in many UEMs, as discussed in Chapter 2. The Sii framework, however, is heavily dependant on 32 specific identified tactics by Bates (1979b,a) and 16 user profiles identified by Belkin et al. (1993). As both models are the building blocks of the framework, the accuracy of the framework is heavily influenced by their accuracy.

Three factors are considered in the discussion of Bates' *tactics* and Belkin's user profiles below. First, reuse and acceptance in subsequent publications within the information seeking community is used to discuss how established the research has become. Second, the appropriateness of the model for use within a reasonable evaluation procedure is discussed. Finally, bibliographic impact is also considered. Many impact measures have been identified for collections of papers produced by authors, institutions, and journals. The h-index (Hirsch, 2005) balances number of publications with the number of times each publication is cited. Similarly, the g-index (Egghe, 2006) goes further to solve some of the criticisms of the h-index by taking into account co-authors from different institutions and countries. These methods, however, cannot directly assess the impact of one paper. Although co-citation analyses are being investigated for assessing the impact of individual papers (Tarrant et al., 2008), the citation counts produced by Google Scholar[3] were used in a recent scientometric analysis of the Human Computer Interaction (HCI) community (Bartneck and Hu, 2009). The citation counts used below were taken from Google Scholar in July 2009. For context, key UEMs from HCI, such as the Cognitive Walkthrough (CW) method, and the Heuristic Evaluation (HE), have been cited 383 and 912 times, respectively.

4.1.1.1 The 32 Tactics Defined by Bates

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 her 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 et al., 1992; Smith et al., 1989).

Several alternative sets of tactics, including those proposed by Fidel (1985) and Shute and Smith (1993), were also discussed in Chater 2. Of the available sets of tactics, Bates' original *tactics* are both the least implementation specific and the most holistic. Both the alternatives cited here, for example, focus on active refinement tactics for keyword searching. Bates' tactics, however, include progress monitoring and analytical bibliographic style tactics may not involve direct interaction with a user interface (Wilson and schraefel, 2009b). Bates', therefore, also provides the largest number of tactics to consider. While the number of tactics has implications for the time it takes to apply

[3]http://scholar.google.com/

Sii, the completeness of Bates' tactics make them more appropriate for the analysing the support that a search interface provides. This choice does not preclude the future refinement of chosen tactics, or the move to an alternative model. The CW method, for example, was streamlined by Spencer (2000), reducing the number of questions asked at every step of interaction from four to two. Any change to this model in the future, however, will require the generation of a new mapping from the alternative set of tactics to Belkin's user profiles.

Finally, in terms of scientometric impact, Bates' *tactics* have been cited 354 times, and of the key articles that cite the tactics, the top five have been cited 2352 times. The two most highly-cited publications that cite Bates's tactics include the seminal work produced by Marchionini (1995) and Ingwersen (1992). Her work has clearly made a significant and strong contribution to Information Seeking research and provides strong evidence that the *tactics* have received strong but implicit validation. In comparison to the alternative models, the moves identified by Fidel (1985), which are similar to Bates' tactics (as discussed in Chapter 2), are cited 78 times, and the most significant citing paper was written by Bates herself, which has in turn been cited 228 times. Similarly, the tactics identified by Shute and Smith (1993), have only been cited by 49 publications. Neither alternative set of tactics have been cited and used by such a large number of authoritative texts.

4.1.1.2 The 16 Searcher Profiles Defined by Belkin and Colleagues

Nicholas Belkin has been cited as one of the more prominent researchers in the field (White and McCain, 1998). Although there has been very little direct validation of this model of searcher profiles, the same paper that proposed the four dimensions (Belkin et al., 1993), and a follow-up paper (Belkin et al., 1995), both use the profiles to build systems that support various types of users. While the searcher profiles were identified through previous empirical studies, user studies were unfortunately not performed on either system to demonstrate the value of using the model. In response to research by Pharo (1999) suggesting that the model may be insufficiently exhaustive for some conditions, however, Cool and Belkin (2002) produced an extended set of user profiles that goes into much more detail. This extension was then validated by Huvila and Widen-Wulff (2006) by applying the extended model to multiple case studies. Even more recently, Kim (2009) provided a modified version of the original profiles designed specifically for web search. By extending the original set and not the extended set of profiles, Kim's work further demonstrates that although work continues in developing potential alternatives, the original 16 profiles provide a strong core that remains well established in the community.

In terms of the appropriateness of each model for the Sii framework, like with Bates' tactics, the original model of searcher profiles remains the most appropriate. Kim's latest profiles developed especially for the web are not appropriate for the framework, as it would limit Sii's scope to only web search interfaces, rather than all search interfaces. Personal communications with Nicholas Belkin have also suggested that not all of the dimensions of the profiles created by Cool and Belkin (2002) could be applicable to the design of the framework and thus the appropriate parts of the model would have to be carefully researched and chosen. Further, as discussed in Chapter 2, the extended version creates 1944 types of searching profiles. It would not be reasonable, within the analysis method, for evaluators to consider so many perspectives. Even by using only the relevant dimensions, 243 unique profiles are created. The remaining research, discussed in Chapter 2, into information seeking contexts has not identified specific sets of profiles, and so cannot be used within the framework.

Finally, as well as seeming to be the most appropriate for use within the Sii framework, Belkin's original profiles have received a much higher number of citations, even considering the date at which they were published. Although it is too soon to tell the citation impact that Kim's web-based profiles will have, the profiles identified by Cool and Belkin (2002), have received only 16 citations in the last 7 years (just over 2 per year), where as Belkin's original 16 profiles have received 143 in the last 16 years (just under 9 per year). Further, the follow-up paper using the 16 profiles has received 179 citations in 2 fewer years. The original 16 profiles have been cited by the seminal work of Wilson (1999), receiving 426 citations, whereas my own earlier doctoral research represents one of the most cited (5 self citations) documents that cites the extended profiles by Cool and Belkin (2002). The validation of the profiles created by Cool and Belkin (2002), provided by Huvila and Widen-Wulff (2006) has also been cited five times, and by papers that I have co-authored. This does not mean to say that the extended classification of searcher profiles is not more complete and more detailed, but is so far not as established within information seeking literature, nor as appropriate in scale for the Sii framework.

4.1.1.3 Summary

The discussion above has considered the choice of models used within the Sii framework, and the alternatives, based upon three factors: their established position in subsequent information seeking literature, their appropriateness for the use within the framework, and their relative citation counts. Together, these factors provide solid reasoning for the choice of models used. As mentioned, it may be that future experience using the framework leads myself or others to re-structure the Sii framework, but for now these models provide a solid foundation. In the following subsection, the mapping that has been used to join these models together is evaluated.

4.1.2 Validation of the Novel Mapping between Models

One of the key contributions of this doctoral work, which has enabled the use of three established information seeking models within one UEM is the novel mapping used to identify which tactics are important for each of the Belkin's searcher profiles. The construction of the novel integration of the models presented by Bates (1979b,a, 1990) and Belkin et al. (1993), described in Section 3.1.3, was a non-trivial process, as each *tactic* (Bates) cannot be obviously, clearly, or absolutely attributed to any specific value of the user profile dimensions (Belkin). Further, as well as being difficult to state that a *tactic x* is associated with a Dimension A, we cannot easily calculate a weighting for these relationships. Consequently, it is important that the chosen mapping, which was carefully reasoned and constructed, be validated so that the margin for error in the non-trivial integration is reduced. As there is no fixed process or metric to objectively produce the mapping, it can only be discussed with and supported by independent judges. For this validation process, 3 search experts and 3 researchers from other academic fields, with little or no knowledge of information seeking, were involved in assessing the existing mapping. The method, results, and discussion of this validation are described below.

4.1.2.1 Method

To formalize the mapping assessment, rather than simply performing structured discussions, an analysis method was designed to: a) clearly present the models to multiple judges, b) collect mapping suggestions, c) identify variations in opinion, and d) produce a refined mapping.

To present the two models being integrated to participating judges and to collect mapping suggestions from them, an online form was built,shown in Figure 4.1. The online form clearly presented each of Bates *tactics*, one at a time, along with a detailed description from the original publications. Below the sequentially presented *tactics* was a persistently available description of each of Belkin's dimension values. Each of Belkin's four dimensions had 2 values, and so a total of 8 dimension value descriptions were shown. For every *tactic* (Bates), shown one at a time, the participating judge was asked to select a dimension value (Belkin) that it most, second-most, and third-most supported. This procedure was similar to the process originally undertaken to produce the initial mapping described in Section 3.1.3, and typically took participants between 2 and 4 hours. The three decisions for each *tactic*, by each judge, were stored in a database and the six completed sets were exported to a spreadsheet for further analysis.

As a first pass for processing the decisions provided by judges, the number of times each dimension value was selected for each *tactic* was summed and the most popular choices highlighted. This spreadsheet analysis provided three types of information. First, it identified parts of the mapping that were unanimously agreed upon by all judges,

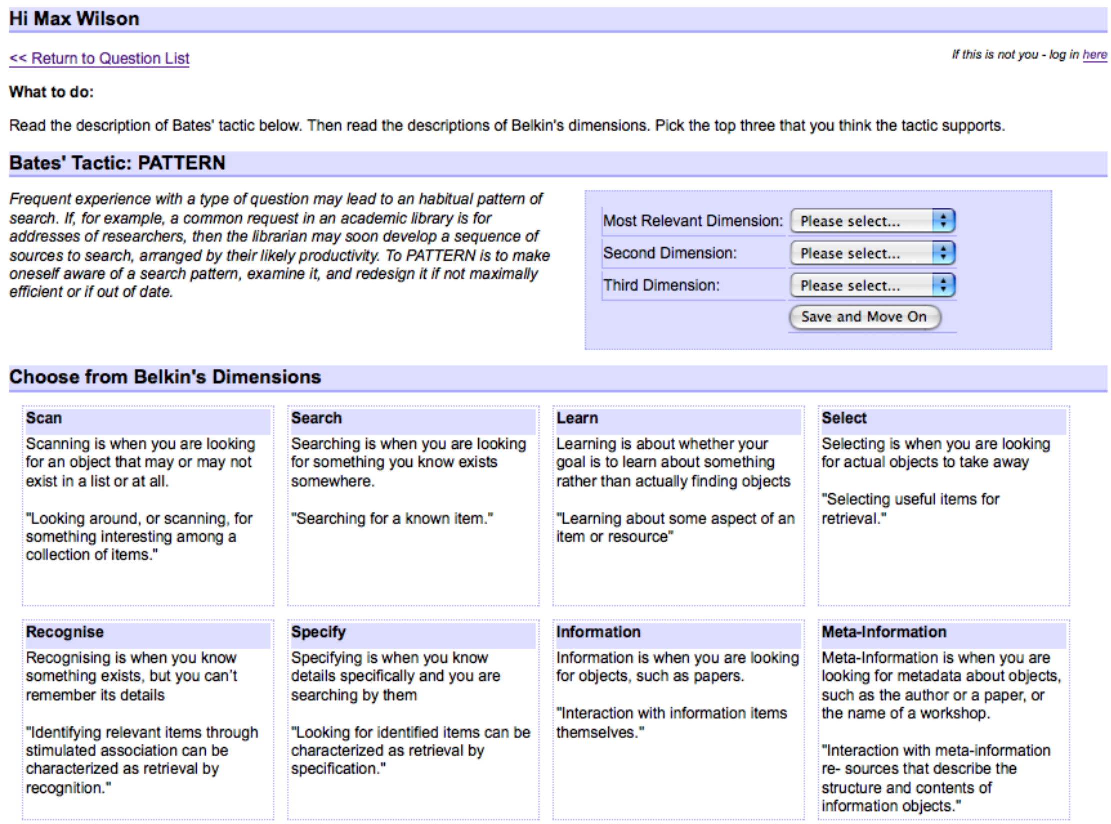


FIGURE 4.1: Online form used to collect expert and novice judgements about mappings between the Bates and Belkin models. The *tactics* are displayed one at a time, along with a definition, and the range of Belkin's dimension values are persistently available for reference. The participant can then select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

including the original mapping, expert, and novice opinion. Any such decisions were accepted without further discussion so as to reduce the additional participation time required of the already generous participants. Second, the process identified 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 analysis and the following discussions are presented below.

4.1.2.2 Results

The process of validating the mapping was successful in that several aspects were discussed and and a revised version was produced. In evidence that producing a mapping between the two models is non-trivial, only 34% of the tactics were unanimously agreed upon without need for further 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 4.1. 38% of the *tactics* received a high agreement, and the decision was taken on the side of the experts in all but one case, where the second highest agreement of the experts for the *tactic* matched the original mapping and the highest agreed *tactic* of the novices. Almost a third of the tactics had to be carefully researched and discussed. In the worst case, the first choice mapping for the RELATE *tactic* was different for every participant.

	Unanimous Decision	**High Agreement**	**Split Decision**	**Varied Opinion**
of the 32	11	12	6	3
%	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

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

The RELATE tactic refers to the movement from one search constraint to a synonym or similar topic. For some judges, this tactic was more closely related to the ability to learn about the contents of a search system. One judge felt that RELATE was more appropriately related to the meta-information within the system, given that a user would be making use of alternative but similar metadata. Another felt that this switch would be in the pursuit of selecting a particular known item, and having simply used the wrong term originally to describe it. Another felt that by changing the search terminology, the user is specifying more accurately what they were looking for. The final decision, informed in part by the second and third choices provided by the judges, was that it would be most appropriately associated with the Search value of the Method dimension, as it aided the identification of a known item by varying the terminology until the right query was submitted to return it. Further, the debate over this tactic represents one of the tactics that was differently assigned compared to the original mapping.

The agreement 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. In particular, this revised mapping affects the information conveyed by Graph G3, as it controls the way that the information from Graph G2 is summarized for each user profile. The revised mapping is now used within the online Sii service.

4.1.2.3 Discussion

From the combination of varying expertise across participant judges taking part in the mapping validation process, several alternative approaches could have been taken to generate a mapping. The mapping could have, for example, considered only the first choices provided by judges. Alternatively, the mapping could have been generated from the expert judges alone. In fact, four processes were considered: the first choice of all participants, the first choice of expert participants, the three choices of all participants, and the three choices of expert participants. The most similar mapping to the finally chosen, discussed, and considered mapping, was the one generated based upon all three choices of the experts, which agreed for 94% of tactics. The least consistent mapping, compared to the chosen mapping, was the first choices of all the participants, which agreed for 84% of tactics. The expert-only mappings of both the first choice and three choice were agreed more closely with the final mapping than the mappings using all six judges.

Although these four automatically generated mappings were extremely similar to the final mapping, each individually contained some indecision. Even the mapping generated from all three choices of the experts alone contained split-decisions, or multiple candidate mappings, for four tactics. Consequently, the indecision on one tactic within a mapping was often resolved, in the final mapping, by considering the opinions of the alternative mappings.

Finally, another alternative approach to generate a mapping could have been to use the decisions made the participating judges as empirical weightings. Although briefly entertained, it was immediately clear that such a system would promote a high margin of error and muddy the analyses of ISIs provided by Graph G3. Instead the process used above promoted the identification debatable regions within the mapping and made sure that they were carefully discussed. The following subsections further discuss this revised mapping by revisiting some of the example analyses described in Chapter 3. Later, Section 4.2 demonstrates that this revised mapping provides insights that correlate more closely to the findings of empirical user studies.

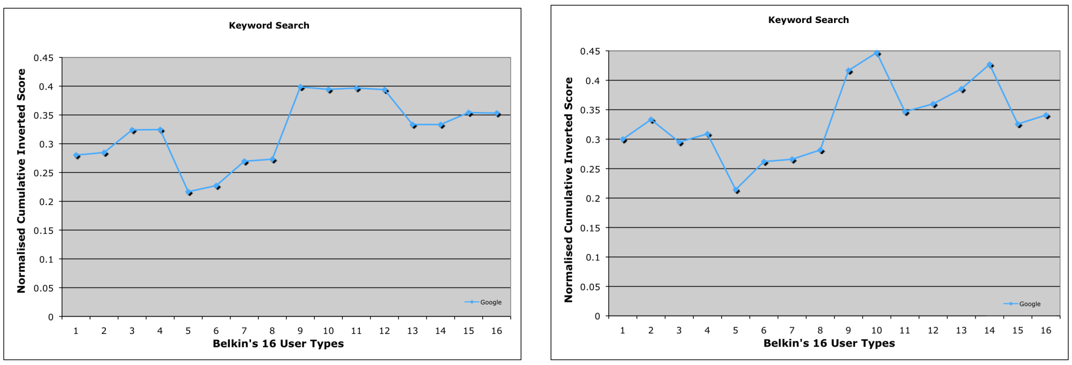
4.1.3 Revisiting Keyword Search

In this section, the earlier analysis of keyword search, presented in Section 3.2.1. In particular, Graph G3 is altered according to the new mapping generated in the validation process above. The affect of this new mapping on the analysis produced is discussed and compared to the original version below.

Figure 4.2 displays the user profile graphs (Graph G3) generated by the original and revised mappings side by side, with the profiles described by Table 2.1 re-shown underneath. First, it is clear that the two graphs are approximately the same shape. The latter half, for example, is higher on average than the first half, indicating that keyword searching still provides more support for those searching for a known target, than those scanning for a potential target. In particular, the least supported user is the one who is unable to specify what they are looking for when they do not know if a useful result exists (ISS5). There is also still a slight emphasis on learning because the main support for selecting results is in the way results are presented, rather than the interaction with the search box. Notably, however, there are two major shifts in way user profiles are supported. First, there is greater support for meta-data over information, which makes more sense as the keyword search box deals primarily in indexing terms, or terms that describe a webpage, rather than the webpages themselves. Further, the emphasis has move from learning in the Goal dimension (odd quarters in Graph G3) to recognising in the Mode column (odd pairs in Graph G3).

Although keyword search is usually described as allowing people to specify their information need, it is actually very hard to be very specific about things such as authors, domains, etc. Instead keyword search implementations, of which Google's is a popular example, provide many ways to support query refinement, such as auto-completion and spelling correction. Consequently the two most-supported user profiles are ISS10 and ISS14, which represent users who are working with and looking for appropriate meta-data about a known target. Further, however, it should be noted that the support for people specifying meta-data about known targets is still higher than for users who are trying to find potential but unknown targets.

In summary, we see that the revised mapping provides an analysis of support for different user profiles that sits more closely with our knowledge of keyword searching interfaces. First, as one would expect, there is much greater support for users who know what they



(a) Original Mapping (b) Revised Mapping

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

(c) User Profiles

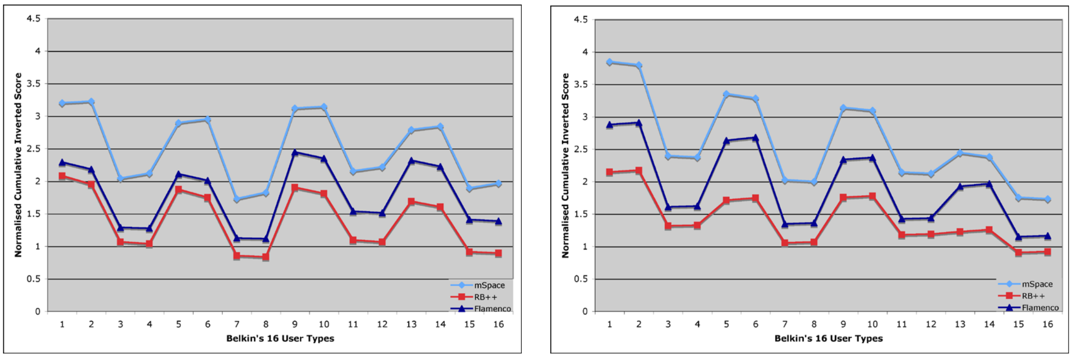
FIGURE 4.2: The original and revised versions of Graph G3, showing the support provided for 16 searcher profiles (Belkin et al., 1993, re-shown below), by Google's keyword search, where peaks represent stronger support.

are looking for (right hand side of the graph and bottom half of the user profile table). Second, keyword search deals primarily with query terms that represent metadata about search results and so we see a peak for every even point of Graph G3, with the revised mapping. Third, keyword search least supports users who are unable to specify the result they need, and do not know if such a result even exists. Finally, the design of good keyword search interfaces support users in recognising more appropriate meta-data for describing their needs.

One open question, for future research, is to examine the increased support for ISS1-4 over ISS5-8, which suggests that the freedom of expression provided by keyword search makes it slightly better for exploring than for browsing (Wilson, 2009). This is potentially in line with previous research (Wilson and schraefel, 2008c; Capra et al., 2007) that suggests the primary benefit that faceted browsers provide over keyword search is in being very specific about a required set of results, rather than free exploration. It is easy, for example, to define four meta-data elements in quick succession by making selections in four facets. In keyword search, a searcher would have to be familiar with constructing boolean queries and using query operators in order to define a relevant set of results, but can freely explore the effects of submitting various keywords. Perhaps, therefore, faceted browsing is better for exploration when users are not feeling creative. In the next section a larger analysis of three faceted browsers, from Chapter 3 is revisited with the revised mapping.

4.1.4 Revisiting the Earlier Comparison of Faceted Browsers

Like the previous section, this section reviews the changes to Graph G3, based on the new mapping, by comparing the revised version to the original version described in Section 3.2.3. Figure 4.3 shows a revised version of Graph G3 on the analysis of 3 faceted browsers reported by Wilson et al. (Wilson and schraefel, 2007; Wilson et al., 2009b). In comparison with the previous graph, shown in the same figure along with Table 2.1 describing the 16 user profiles, we can see three specific improvements in what Graph G3 tells us about the three faceted browsers.



(a) Original Mapping (b) Revised Mapping

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

(c) User Profiles

FIGURE 4.3: The original and revised versions of Graph G3, showing the support provided for 16 searcher profiles (Belkin et al., 1993, re-shown below), by the three faceted browsers, where peaks represent stronger support.

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 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 making a series of selections. Consequently, users cannot progressively recognize that their selections have found the right results. 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 an extra facet specifically for target 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. The next stage, discussed in the section below, is to evaluate the whole framework, using this revised mapping, by comparing it to the results of empirical user studies. The versions of Graph G3 produced by the original mapping are also included to demonstrate where the improvements lie.

4.2 Validating the Framework's Accuracy against Users Studies

In light of the challenges experienced when comparing UEMs (Law et al., 2009), the limitations of comparisons (Law et al., 2009; Gray and Salzman, 1998), and the varying types of results found within (Blandford et al., 2008) previous evaluations of UEMs, discussed in Section 2.5, the sections below do not try to prove that Sii is better or more accurate than other methods. Instead, as often suggested in response to the these evaluation issues (Law et al., 2009; Olson and Moran, 1998), the focus remains grounded in demonstrating the insightfulness and value of the analyses by comparing them to the results of empirical user studies. The aim of analysing these example cases is so that practitioners can use Sii both confidently and appropriately.

The method used for both examples below was the same. The designs or systems evaluated in the studies were analysed 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, these analyses show how the study results can be explained and evaluated in more detail than through user studies. To further discuss the previous validation of the mapping used within the Sii framework, Graph G3 is shown in both forms for each study. Where appropriate, the more accurate representation provided by the revised mapping is discussed.

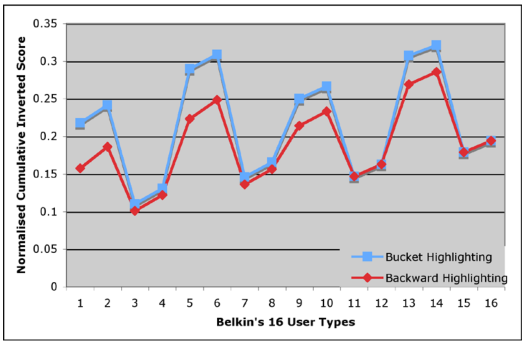
4.2.1 Backward Highlighting

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 between the models of tactics and searcher profiles. 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. (2008). To aid comparison, the new version of Graph G3, using the refined mapping, is shown alongside the original version in Figure 4.4. Further, Figure 4.4 includes the 16 searcher profiles from Table 2.1 to support interpretation of the graph.

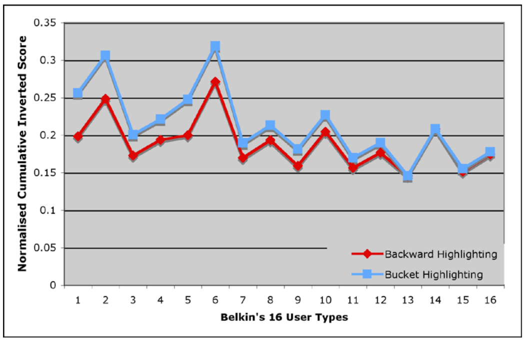
4.2.1.1 Results

In the user study performed by Wilson et al. (2008), a control condition without highlighting was compared to two implementations called Backward Highlighting and Bucket Highlighting. These two implementations are shown and described in more detail in Section 3.2. To demonstrate the advantages of these highlighting, and the difference between the two implementations, users were asked to perform three sets of tasks with all three conditions. The first task was to learn related facts about a specific item of metadata in the facets, which, aside from finding the initial item, required participants to Scan to Learn Meta-information about a topic they could specify. This is primarily ISS4, but does not preclude the use of ISS1-3, although the highlights primarily present metadata (ISS2 and ISS4). The second task was to re-find facts, chosen by the evaluator, from the list they had written down. We expected that users would be able to use ISS14, which was to use the Meta-information they had learned to Recognize ways of Selecting known items (Searching not Scanning), but instead we saw people simply repeating their previous actions, but searching for a known item (ISS16). Finally, users were required to recall as much information about the original target as possible and write them on paper. This final task, although not a seeking task, indicated to what extent learning had occurred in the first ISS1-4 task.

The results of the user study indicated that there was very little difference between the two designs, but four key findings were presented by Wilson et al. (2008), and Table 4.3 shows which of these findings can be seen in Graph G3 using the old and new mappings. First, the study showed that for task 1, more related meta-data could be



(a) Original Mapping



(b) Revised Mapping

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

(c) User Profiles

FIGURE 4.4: The original and revised versions of Graph G3, showing the support provided for 16 searcher profiles (Belkin et al., 1993, re-shown below), by the two Backward Highlighting designs (Wilson et al., 2008), where peaks represent stronger support.

discovered with both highlighting implementations (#1 in Table 4.3). Both the original and revised mappings showed this emphasis on meta-data (odd half of each pair), but the revised mapping highlights this difference further. The results of task 3 provided statistical evidence to show that slightly more about the meta-information could be learned with the Bucket Highlighting condition (#3). This is shown in Figure 4.4 by the most significant gaps being on the left of the graph, where users are scanning and learning more often. In particular, we can see extended support for ISS 4 in the revised mapping that is not present in the version using the original mapping. It should be noted, here, that Belkin's notion of learning relates to the potential to learn, rather than the actual recallable information consequently stored in long-term memory. Although the revised version of G3 indicates that more can be learned, rather than simply recognised, the difference in amount of information later recalled was unfortunately not significant within the bounds of the sample size used.

The Backward Highlighting user study also showed no change in behaviour of users searching for known items (#2). The original mapping placed more emphasis on selecting (or simply finding) information, rather than exploring and learning. The user study, however, indicated that both implementations could browse and find facts significantly more than without backward highlighting. The revised G3 represents this finding more accurately by placing the emphasis on the learning parts (odd quarters) of the graph. Further, the original mapping indicated that Bucket Highlighting would provide more support for people recognising previously used metadata (ISS14), but this behaviour was not seen in the user study. Finally, the overall increased support described by the graphs could have predicted that the users, overall, would have preferred the Bucket Highlighting technique (#4). Overall, as shown in Table 4.3, the revised version of G3 shows all four of these findings, where as the original version only showed two of these findings correctly.

#	Study Finding	Shown by Graph G3 Original Mapping	Revised Mapping
1.	Improved discovery of meta-data	Yes	More So
2.	Did not affect selecting known items	No	Yes
3.	Additional facts retained in memory	No	Yes
4.	User preference for Bucket	Yes	Yes
	Total	**2/4**	**4/4**

Table 4.3: Table indicating the findings from the user study of Backward and Bucket Highlighting (Wilson et al., 2008) that are identified by the use of the Old and New Mappings.

4.2.1.2 Discussion

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 incorrectly indicated that backwards highlighting is in general better for users searching for a known item (right half of the graph) where as the highlights actually well supports users who do not already know the relationships (scanning) even to recognize them. 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 4.4) may also be more accurate than the opposite trend portrayed by the original mapping.

This revised analysis suggests that ISS6 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 backward highlights, which are designed to reveal related meta-data in the columns of a directional faceted browser. The highlights provide the least support, however, for user type 13, who is searching to select a known item, but is unable to specify its details, which makes sense as Backward Highlighting has little to do with the core information objects, which makes them hard to recognise.

4.2.1.3 Summary

In analysing the user profiles created by the tasks of the user study (Wilson et al., 2008), it is clear that the results found correlate more closely with the revised mapping than the original mapping. Further, from the discussion presented thereafter, it is also clear that a much richer analysis of the two implementations can be produced than by the empirical evidence provided by the user study. In particular, the user study included tasks that represented only a subset of the user profiles shown in Graph G3. G3, therefore, provides a much richer analysis from a wider set of user perspectives. In particular, where the user study told us that users were able to learn more about a particular piece of meta-information, and the original mapping suggested this was purely because of the increased support for recognising, the new mapping shows that it enables learning and the scanning of previously unknown but related meta-data. Overall, this comparison provides evidence that, especially using the revised mapping, the results of the user study can be identified using the Sii analytical inspection framework. The following example provides similar evidence, but in a much larger comparison of three complex browsers.

4.2.2 3 Faceted Browser Study

Capra et al. (2007) reported on a user study comparing two alternative faceted search interfaces with an existing website, in a user study with similar motivations to the analysis of three faceted browsers compared in Section 3.2.3 (Wilson and schraefel, 2007; Wilson et al., 2009b). The original website was the Bureau of Labor Statistics, which presents a hierarchical classification on its homepage that categorizes US government reports. The website was compared to both the RB++ browser and an un-configured, or 'vanilla, version of Endeca[4]. Both browsers included faceted classifications over the same goal objects: government reports on labour statistics.

Capra et al. (2007) performed two studies: one between participants and one within participants. The first study was designed to provide empirical results and the second to provide qualitative data and 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. The types of task used in the study break up into two types of user according to Belkins dimensions. The two lookup tasks both placed participants into ISS13, as they knew a particular report existed in the system (Searching) and their Goal was to select the

[4]Endeca is a commercial vendor of enterprise search, which has a default and configurable interface to the structured metadata they can produce from corporate data. They supply customers including Walmart, IBM, and Ford

answer to show they have completed to the task. As they did not know all the facts about the target report, they could not specify (Mode) which report they needed but could recognize reports that might contain the answer. Finally, the participants were looking for an answer in the reports, rather than in the classification schemes, and so they were looking for Information, not Meta-Information.

The exploratory task placed participants into ISS1 and ISS2, 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 was no specific answer to the question, the participants were scanning in order to learn more about the topic. Like the previous task, participants were only be able to recognize relevant reports as they saw them. The following subsection discusses the results of this user study in comparison with the analyses provided by the framework, presented in Graphs G1-G3 (Figure 4.5, Figure 4.6, and Figure 4.7).

4.2.2.1 Results

The results of the study performed by Capra et al. (2007) were not as expected, as no browser in the study particularly outperformed 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 revised Graph G3 (Figure 4.7) that the point where the three browsers provide the most even support is at ISS13 - the user type that represents the simple and complex lookup tasks. Notably, ISS13 is the user profile least supported by the RB++ browser according to the analysis provided by Sii. Further, we can see that for the exploratory tasks (ISS1 and ISS2), the website even provides slightly more support than the RB++ browser. These findings could have predicted that the differences between the ISIs 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. Knowing these differences could have helped Capra et al. (2007) to design the study, as discussed in the scenarios of use described in Section 3.2.4, so that it included tasks that highlighted the differences between the three interfaces. 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. This finding is in line with the longitudinal analysis of the mSpace browser provided by Wilson and schraefel (2008c), which showed that the facets were most frequently used for quickly specifying multiple constraints over a dataset.

Notably, the G3 analysis provided by the original mapping does not correlate at all with the findings of the empirical tasks of the user study. In particular, the original G3 analysis suggests that the RB++ browser would significantly outperform the other two

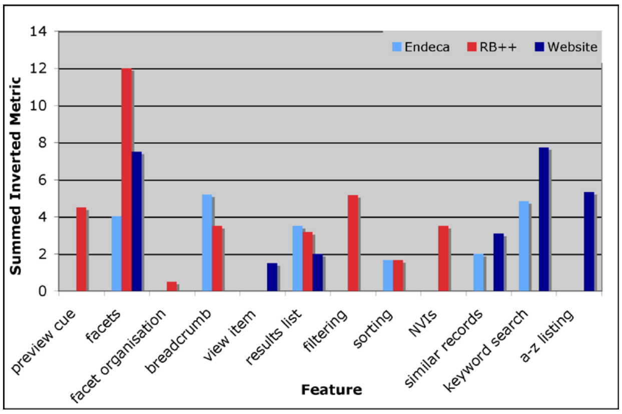


FIGURE 4.5: Graph G1 showing the support provided by the features of the three interfaces studied by Capra et al. (2007), where taller bars represent stronger support.

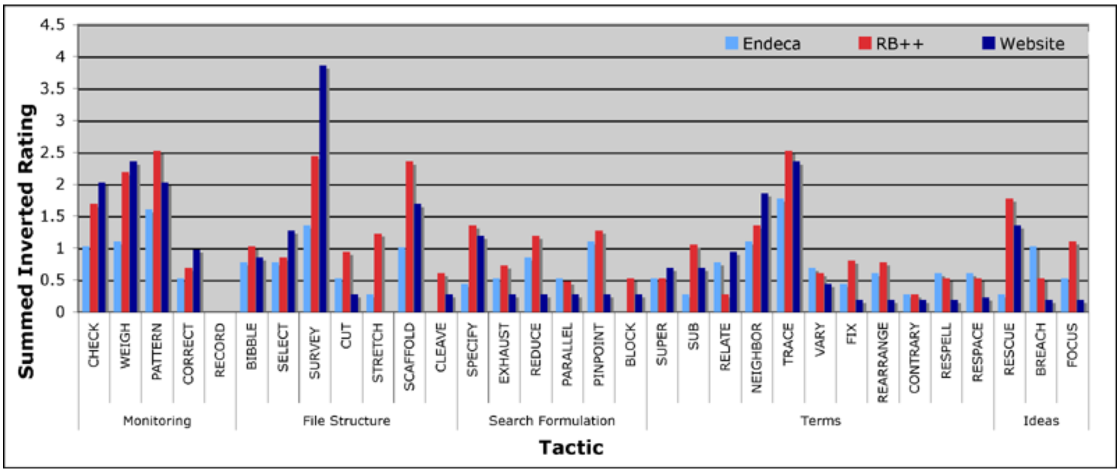
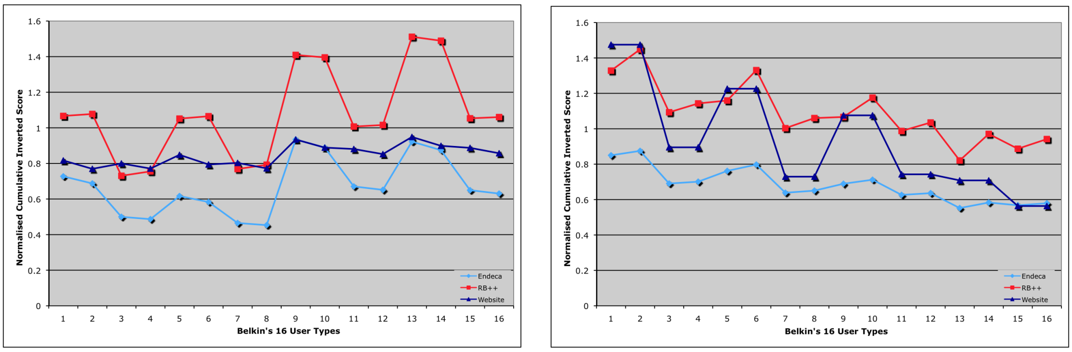


FIGURE 4.6: Graph G2 showing the support provided for 32 search *tactics* (Bates, 1979b,a), provided by the three interfaces studied by Capra et al. (2007), where taller bars represent stronger support.

browsers in both tasks (ISS 13, and ISS1+ISS2). This provides further evidence for the improvement to Sii's internal mapping.

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 4.4. Graph G1 (Figure 4.5), the analysis by feature, could have also predicted these results. According the results of the framework, the original website provided the strongest keyword search function (#1 in Table 4.4); 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



(a) Original Mapping (b) Revised Mapping

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

(c) User Profiles

FIGURE 4.7: The original and revised versions of Graph G3, showing the support provided for 16 searcher profiles (Belkin et al., 1993, re-shown below), by the three interfaces studied by Capra et al. (2007), where peaks represent stronger support.

documents were to be found given certain selections (#10). RB++ provided numeric indicators in two forms, specific values (NVIs in Figure 4.5) and previews of affect before clicking (Preview cues in Figure 4.5).

4.2.2.2 Discussion

Although the majority of results could be shown through the graphs produced by Sii, there are two results that cannot be so clearly explained by Graph G1 (Figure 4.5). The Sii analysis indicates, for example, that the representation of results in Endeca was 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++ (#12). In the paper, Capra and colleagues suggested that the number of items in the facets 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. Another possible explanation can be drawn from Graph

	P/C	Feature	From G1 Feature Analysis (Figure 4.5)
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.
Vanilla 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 4.5 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 4.4: List of identified pros and cons of the interfaces that could have been predicted by the 'by feature' analysis shown in Graph G1 (Figure 4.5)

G2 (Figure 4.6) that the original website was particularly strong for tactics such as SURVEY, WEIGH, and CHECK. Notably, the clear layout of the classification on the front page of the website supports the ability to SURVEY a wide range of options. It could be that participants rated the structure of facets negatively for its lesser support for the SURVEY tactic, in comparison to the stronger support provided by the BLS website (#12). Regardless of the potential explanations for this unpredicted result, it's presence provides a challenge for the on-going research into the framework. Chapter 5 begins to address this challenge by investigating a possible extension to the framework that assesses the simplicity or complexity of ISIs.

Finally, although it was not included in the most and least favorite features, participants noted that that selecting a result in either of the two faceted browsers, the participant was temporarily transferred to the BLS website in order to view the document. When using the original website, however, the users did not experience this discontinuity across interfaces. This disconnect is also shown in Graph G1 (Figure 4.5), where the support when viewing an item is only present for the website.

4.2.2.3 Summary

It is clear from the analysis above that Sii, and only specifically using the revised mapping, could have predicted the majority of the results discovered by the study provided by Capra et al. (2007). Although the faceted browsers, and in particular the RB++ browser, were expected to better support exploratory searches, the original website provides a very clear hierarchical representation of the dataset. Further, the known item search tasks actually represent the user profile least supported by their browser. Graph G3, however, clearly identifies times when the RB++ browser does provide significantly more support than the other two interfaces, and future studies could be designed to highlight these occasions where users are asked to specify more complex information needs in the system, for example. Further, the most and least preferred features of the three interfaces, as reported by Capra et al. (2007) match the analysis provided by G1 very closely. When the results were not so obvious in G1, significant differences betwen the designs could be seen in G2. Overall, Sii provided a very detailed analysis of the three complex information seeking interfaces, of which many parts accurately correlated to empirical data.

These two comparisons provide strong support for the Sii framework, which is further enhanced by a number of other analyses presented throughout the rest of the Chapter 4 and the example studied in Section 5.2. The next section, in particular, discusses short case studies of how Sii might fit into the working practices of practitioners and real projects.

4.3 Validating the Use of the Framework in Context

The validation process above has so far provided evidence for the accuracy of the Sii framework. First, the process has defended the choice of models being used in the framework. Second, the novel mapping between the two models has been examined by multiple judges to the extent that a revised mapping has been proposed based upon the consensus of a group. Third, the whole framework, including the new mapping, was shown to provide accurate analyses of ISIs that have been evaluated by empirical user studies. This section aims to further validate Sii by studying how it relates to working practices in two phases. First, case studies of actual or potential use of the Sii framework by practitioners are presented. Second, several pilot studies are presented that provide preliminary evaluations of the practical issues when applying the framework.

4.3.1 Case Studies of Practitioners

Furniss et al. (2008) described four important contexts of usability practices that influence the use of UEMs. First, Furniss suggests that it is important for UEMs to fit suitably within the working practices of both usability practitioners and their larger organisational contexts and procedures. Second, it is important that methods foster relationships both with clients and with colleagues. In part, fostering relationships is concerned with the third context: supporting communication of ideas. Finally, there are many other aspects to usability working practices, such as reporting and even visibility within communities, that usability methods can support. Notably, however, it can take 5-10 years to have a full understanding of how a method fits in with these contexts of real usage. Cognitive Walkthroughs, for example, evolved many times between being originally published (Lewis et al., 1990) and publishing the procedure that is considered most definitive (Wharton et al., 1994). Further, it wasn't until 6 years later that context of real working practices prompted Spencer (2000) to published a 'streamlined' cognitive walkthrough that better supported these four contexts. Similarly, the MAUSE WG1 review (Scapin and Law, 2007) noted that it was too soon to identify the impact several methods published between 2001 and 2005, including K-MADe, SEEM, CASSM, and the User Action Framework.

Excluding the early workshop publication on the design of Sii (Wilson and schraefel, 2007), Sii has only just been formally published (Wilson et al., 2009b). Refinements based on experience and usage, such as those discussed in this chapter, will contribute to the understanding of Sii's impact over time. Chapter 6 discusses the on-going refinement and future work leading towards a retrospective understanding of Sii's impact. Instead, this section focuses on discussing the framework's potential to support these four contexts, with either early adopters or those hoping to use the framework in their forthcoming projects. Three short case studies are considered over the following three subsections. Some of the participants, who generously gave up their time to discuss if or how Sii fits into their projects, chose to remain anonymous or to keep their projects confidential.

4.3.1.1 Case Study 1: An Academic Study of Interactive Information Retrieval

Overview

In the early phases of designing alternative representations for interactive query refinements, which were later presented at the Joint Conference of Digital Libraries (Diriye et al., 2009), a doctoral student at University College London analysed two potential designs, along with a baseline interface, using the original version of the Sii framework. Built using the Google and Yahoo! search APIs (Application Programming Interface), Diriye and colleagues tested the hypothesis that adding result-like snippets to query expansion terms would support users in seeking and exploring tasks. Figure 4.9 shows the main experimental condition of their study, where the query expansions each have a short descriptive snippet to further define them. Figure 4.8 shows the standard query expansion interface and, for a final control condition, a baseline interface was created that did not have any expansion terms.

According to the theory of poly-representation, described in more detail by Ingwersen and Järvelin (2005), the extra representational text describing the expansion terms should support better judgements. Participants were asked to perform simple known-item searches, complex known-item searches, and exploratory search tasks similar to those used by Capra et al. (2007) in the comparison the three faceted browsers described in Section 4.2.2. Diriye and his colleagues noted the frequency of certain behaviours, and used grounded theory (Strauss and Corbin, 1998) to analyse the qualitative comments from think aloud and interview data. The analysis revealed that although users preferred the baselines for its familiarity, especially in the known-item search tasks, but the poly-representational version reduced confusion from ambiguity during exploratory tasks.

FIGURE 4.8: The standard query expansion interface studied by Diriye et al. (2009).

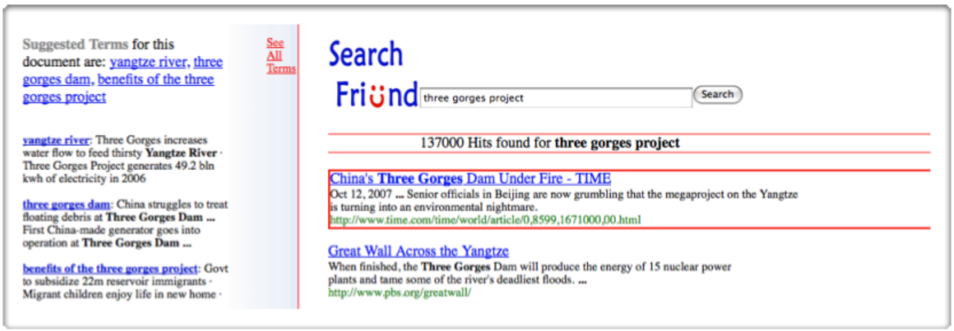


FIGURE 4.9: The poly-representational query expansion interface studied by Diriye et al. (2009).

Results of Sii analysis

Reported separately in an unpublished internal report, Diriye (2008) analysed the three conditions by building a spreadsheet analysis using the original framework described by Wilson (2007), before the online framework was available[5]. Diriye considered each

[5]This example can also now be found online, but using the refined mapping, and correlates more closely to the advantages provided by the extra snippets that were identified in their paper.

of the interface conditions to be incremental, so that the query expansion condition included the baseline and the expansion terms as two interface features. Similarly, the poly-representational condition included the baseline, the extra query terms, and the snippets as three separate interface features.

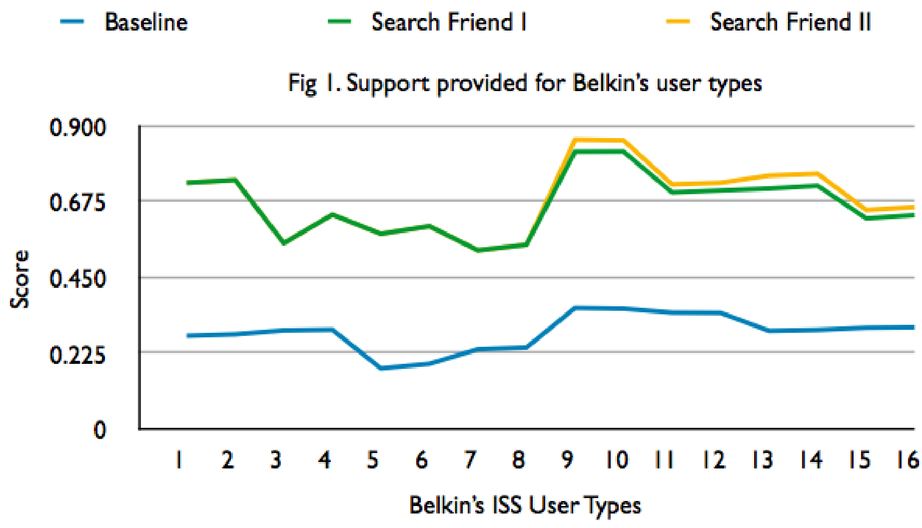


Figure 4.10: Graph G3, as generated by Diriye (2008), showing the support provided for 16 searcher profiles (Belkin et al., 1993), by the three conditions in his study of query expansions, where peaks represent stronger support for search.

While the analysis, shown in Figure 4.10, indicated that the poly-representational condition would be best for all cases, their study results showed that the baseline condition performed best for known-item searches. Proposed as one of the main scenarios of use, the analysis performed by Diriye was to study the effect of adding a new feature to an interface. What this clear example of comparing three incremental designs demonstrates is that the Sii will always value the addition of extra features in a design. In Diriye's terms, the framework "identies how well an interface is able to support different strategies, but fails to address how precisely".

Suggested refinement to Sii

The suggestion put forward was that the functionality of the baseline is targeted more at known-item searches, while the additional support provided by the query expansions, and snippets, means that the functionality aims to support refinement and exploration. Consequently, Diriye suggested that the values shown in the G3 analysis be normalised by the overall support provided for all user profiles. This normalisation means that graph G3 "is able to identify how well a user type is supported in respect to all other user types".

The result of the normalisation, shown in Figure 4.11, is that the three lines generated in Graph G3 are placed on separate scales so that instead of showing the total support provided the interface in relation to the total support provided by other interfaces, the

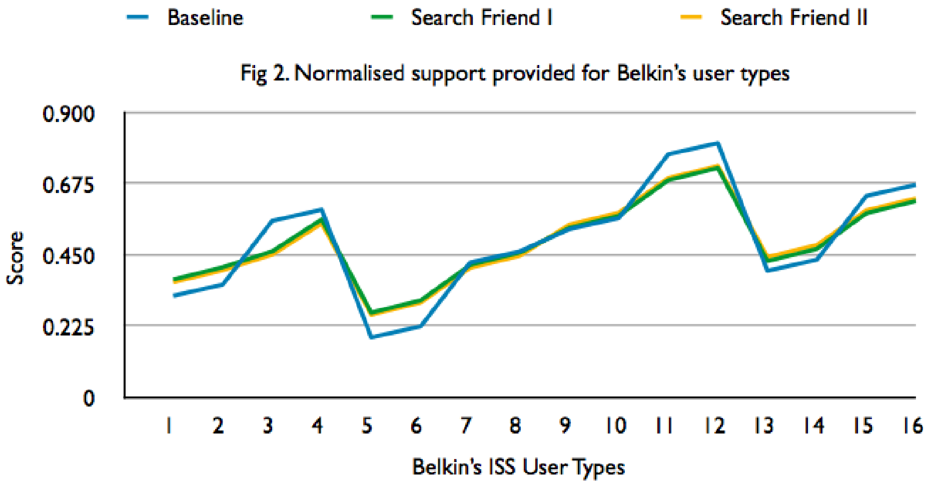


FIGURE 4.11: A suggested normalisation of Graph G3, generated by Diriye (2008), showing the support provided for 16 searcher profiles (Belkin et al., 1993) by the three conditions in his study of query expansions, where peaks represent stronger support for search.

different interfaces can be compared in terms of which user types the more support than others. The result of this normalisation meant that Graph G3 then showed results that correlated more closely with their study findings. In Figure 4.11, we can see that the support of the baseline interface tends towards supporting user types who can specify their desired items (known-item searches) than those who will depend on recognising results.

Discussion

It is not clear, at this stage, whether this transformation is appropriate a) all of the time, b) some of the time, or c) never. It is possible that this more accurate correlation with their study results is coincidental. It has worked particularly well, in this case study, because each interface has a different number of features. Additional normalised analyses of the examples discussed so far in this dissertation are available in Appendix B. In the analysis of three faceted browsers in Section 3.2.3, however, Graph G3 suggests that the mSpace browser provides more support across all interfaces because it has more features than the other browsers.

Applying Diriye's transformation in both these cases allows Graph G3 to ignore that one interface has more features than the other, and instead analyse which features implemented features better support. A transformed version of G3 from Section 3.2.3, however, suggests that Flamenco provides more support for users who are recognising behaviour, where as mSpace is best in situations where users want to specify the items they would like to select. This seems counter-intuitive, as Flamenco specifically narrows down information in the facets, making it easier to specify information, but the process of removing information, however, can make it harder to recognise metadata. Similarly, a transformed analysis of Backwards Highlighting indicates that the un-grouped implementation provides better support for specifying during known-item searches. This also seems counter-intuitive, as grouping the highlights would support the fast selection of metadata that the user knows is related to their target search. Further it weakens the emphasis that the grouping effect has on the recognising and learning user profiles. This example shows clearly that the transformation can make it appear that adding a feature reduces the support for some user profiles, which may be incorrect.

In the analysis of the faceted browsers studied by Capra et al. (2007), described in Section 4.2.2, the transformation narrows the difference between the known-item search task user profiles, but increases the gap between the interfaces in the exploratory search task ISS profiles, suggesting that the original website provides much more support for browsing conditions. While no interface performed significantly better in any of the tasks in their study, the subjective survey responses did provide higher ratings for the original website in questions surrounding exploration and learning. Consequently, this transformation does potentially provide a more accurate analysis of the three ISIs studied by Capra et al. (2007).

Alternatives to the Normalisation

As fully acknowledged in our previous publications and my doctoral reports, and discussed further in Chapter 5, there are other factors that could have affected Diriye's findings. Section 5.1 discusses an extension to the Sii framework that provides an opposing force to simply adding functionality, by assessing the cost of complexity imposed by adding the new features. It may be that Diriye's findings showed that users performed better in known-item tasks with the baseline, because there was less additional functionality to confuse or distract their searching behaviour. This theory, which also forms part of the on-going work described in Chapter 6, allows for the fact that adding textual snippets to query expansion terms does increase the support for all user types, but may come at a cost to another factor of usability: simplicity.

Summary

It is clear from the varying accuracy created by applying the results that further research is needed to understand its benefits[6]. Potentially, Diriye's normalisation might only apply when ISIs have an unbalanced number of features, or when the comparison involves the incremental addition of features. Notably, in the third case study described in this section, the transformation suggests that the original interface is better than the new version for some user profiles. This transformed analysis, however, was considered less accurate by the team working on the project.

So far this case study has not provided much discussion on the usefulness of Sii within the four contexts described by Furniss et al. (2008): existing procedures, communication,

[6]The transformation can be optionally applied to projects using the online version of the framework.

relationships, and secondary activities. In this scenario, however, the framework clearly did fit into the working practices of the academic team, but the results were rejected. It is not clear whether the framework requires modification to be more accurate, but it is also possible that their study results would correlate more closely with other usability methods such as heuristic evaluation, which considers the intuitiveness of a design. Diriye's reflection on using the framework, however, is that Sii does not account for which user profiles the functionality provided within a design is trying to support. It is clear, therefore, that the benefits of the framework need to be carefully defined so that practitioners can a) be confident in the analyses provided, b) be clear about the kinds of results it produces, and c) be clear as to when it is appropriate to use. Finally, as an academic team, which both developed and studied their designs, there was little space in the case study for assessing how the Sii framework can foster relationships. The following case studies investigate these factors further.

4.3.1.2 Case Study 2: A Digital Library Interface Consultant

Overview

Having seen some of the early work during the development of Sii (Wilson and schraefel, 2007), an industry practitioner, who works as a consultant on a wide range of projects, agreed to meet for two hours to discuss how the framework could have supported a concluding project with the American National Archives[7]. Due to the nature of the project, exact details and a Sii analysis are not discussed, but the ability for Sii to support the work is not dependant on these specifics. As a consultant, the participant was used to working, potentially, on multiple projects with different clients at any one time. Further, issues of communication were key for both maintaining relationships with clients, understanding the needs of the clients' users, and describing designs, ideas, and reasoning. In terms of the fourth usability context defined by Furniss, it was considered important for the interviewee to be able to generalise the findings or principles from one project to communicate with future clients and guide the design of future projects.

First, the principle of Sii's analyses were discussed in the context of several of the examples presented above. The online version of Sii, described in Chapter 3, was then presented until the participant was comfortable with the procedure of use and the types of results produced. The discussion that followed was positive, and provided an insightful perspective on the value of the framework. Throughout the discussion, however, one key theme was prioritisation, as discussed below.

Participant's perspective of Sii

The participant, here-forth called the consultant, described the framework as being a less of a usability evaluation method, but more of a functional screening tool. The consultant

[7]http://www.archives.gov - National Archives and Records Administration

noted, in particular, that such an action-counting method did not consider many aspects of usability, such as aesthetics, preference, and ease of use. ISO Standard 9241-11 defines usability as "the extent to which a product can be used by specified users to achieve specified goals with *effectiveness, efficiency* and *satisfaction* in a specified context of use." Sii does consider both effectiveness, in achieving a set of search tasks, and efficiency, in how many moves it takes to achieve the tasks. Further Sii is very closely set in the context of searching, where as many other inspection methods do not consider specific contexts of use. The point raised by the consultant, however, is that empirical studies, or methods like the Cognitive Walkthrough, consider the user's reactions and performance in tasks, rather than the functionality provided by a user interface to support tasks. For the consultant, this perspective was important, as the results of Sii when used in real projects could not be taken as declarative.

The consultant did, however, value the three elements of the Sii analyses to be important. Graphs G1, G2, and G3 were described as analyses of the tools, processes, and people respectively. In the words of the consultant, the framework provides a "people-process-tools" analysis. The "tools analysis" described the components of a design that are contributing to the experience. The profiles provide a view of goals and motivations, which the consultant described as critical. Finally, the tactics graph provides a view of the processes that people might take. Together, these things were considered valuable, but the consultant thought the framework needed a "contextual overlay".

Fitting with working practices

For use within the consultant's working practices, the issue of prioritisation was discussed from many perspectives. The realistic fact of the consultants previous experiences was that every project involved negotiation between what was 'the ideal interface design' and what was possible given the time and financial constraints. Dependant on this factor, the most appropriate UEMs were those the consultant could use to prioritise what was important for the project. In this respect, understanding the contributions of each feature in designs incrementally would be a valuable feature of the tool. It was suggested that practitioners using Sii could incrementally include and exclude features in the analysis to see how a) their individual effects, and b) the compounding effects in different combinations.

The prioritisation of features for implementation, within project constraints, was further situated within type of data being provided and the expected use of the system. The example considered during discussion was between a document archive, and a census archive. A document archive might be primarily for retrieval, but one of the key expected uses for a census dataset would be to explore personal histories. In this latter example, the consultant wanted to remain focused on this more exploratory perspective and prioritise the functionality to support them, rather than all profiles. In some ways the break down of user profiles is useful for prioritising certain profiles. The consultant, however, considered the maintained representation of the broad spectrum of users to be unnecessary. In a similar manner to incrementally analysing interface features, the consultant wished to choose a subset of profiles and filter the alternative graphs from their perspectives.

Finally, viewing prioritisation from a third perspective, the consultant considered certain tactics to be more important than others. The consultant suggested that the SURVEY tactic, which may involve very little interaction, is in principle important to support properly. As part of prioritising the functionality, and the appropriate users, the consultant would want to filter all the graphs from the perspective of certain tactics. Further, the consultant suggested that it would support communication with clients if the system noted any particularly important tactics, if the underlying theory had classified them as such. This discussion of more or less important tactics is discussed as part of the future work planned for this research. Finally, as part of this discussion of important tactics, the consultant considered it important for the system to explain, if asked, which tactics were important to each user profile. Currently, this mapping within the framework is not made visible in the Sii interface.

Relationships and communication

In support of the four contexts of UEM usage presented by Furniss et al. (2008), the consultant brought the issue of relationships and communication to the forefront of discussion many times throughout the interview. In particular, for using Sii to support communication, the consultant felt that significant framing would be required before presenting such raw analyses to clients. For such a communication, the consultant envisioned four phases of framing: 1) the goal of performing such an analysis, 2) a description of what using the method is trying to achieve, 3) a contextual overlay over what the analyses will reveal, and then 4) a description of what the analyses are saying in those contexts.

In respect to the filtering and prioritisation, using a project-context overlay for the method would allow the consultant to frame the analysis in terms of a) what was important for the client, b) what the client would be preoccupied by, or c) what was important to highlight from the many findings.

Summary

The perspective provided by this short case study was of a consultant-style Sii user and highlighted some unique insights, especially in terms of communication, for dealing with a high turn over of temporary colleagues, clients, and their clients. Building re-usable examples to take from project-to-project, was seen as a vital part representing the fourth context of usability evaluation described by Furniss et al. (2008): secondary activities. The key value, however, of re-using examples was in communicating these examples and principles to different styles of audience. A significant improvement to the Sii framework, would be to tailor the analyses produced to both the priorities of the project and the expertise of the audience.

4.3.1.3 Case Study 3: An Information Architect in a Web Development Company

Overview

This third case study describes the result of two hour-long interviews with an Information Architect (IA) at a marketing and web design company: Design Haus[8]. The company, with this IA as the technical lead, recently completed the redesign of the Royal Yachting Association (RYA) website[9]. The role of Design Haus was to evaluate and redesign the information architecture of the site using techniques such as wireframes and visual mock-ups. The development in this project, however, was handled by the software developers at the RYA. Maintained support is provided through additional consultancy as the designs are completed. Final testing was the performed in conjunction between the two parties before going live.

During the first interview, the Sii website was introduced and an analysis of the new and old[10] implementations of the RYA online shop[11] discussed. After the interview, several design recommendations were provided to the IA for review, along with a second analysis that demonstrated the benefits that these changes would bring to the support provided for searchers[12]. Further, to aid communication, the keyword search provided in both the old and new designs were compared in a third analysis to the keyword search provided by Google[13]. These analyses are available online, and included in Appendix B, but the focus here is on the discussion surrounding the analyses and the use of the framework within the company.

Summary of Analyses and Results

In summary of the analyses, the new design, as of 2009, provided a much broader support for search tactics and users across the board, but still missed out on supporting 6 tactics altogether. In particular, the re-design made better use of faceted classifications and visualisation of results. Further, the addition of breadcrumbs and numbers, indicating the number of related results for each facet value, increased support for many tactics and search profiles. The comparison to the Google's keyword search, however, revealed that the design could be further improved by elements such as operators and spelling corrections. For users, while learning and selecting (or buying in the case of online

[8] http://www.dhaus.com
[9] http://www.rya.org.uk
[10] http://web.archive.org/web/20071008231541/www.rya.org.uk/Shop/
[11] http://mspace.fm/sii/project.php?pid=000011
[12] http://mspace.fm/sii/project.php?pid=000012
[13] http://mspace.fm/sii/project.php?pid=000013

retailers) were fairly balanced, more support was provided for users who did not know what they wanted to buy.

Several design changes, as listed in Table 4.5 were recommended, which particularly aimed at improving all round support and balancing the support for users who knew what they were trying to buy. The expected benefits were visualised in the third analyses[13]. The design changes were classified by the IA under two categories: novelty and likely use. First, the IA classified the design changes as either: a) a bug, b) lost in communication, c) a good idea, d) not sure, e) a bad idea, or f) purposefully avoided. The first two of these classifications assumes that the design idea was planned, but had not materialised, with the first being known and the second being realised by the analyses. Options c) to e) assume that the idea had not been directly discussed, where the options vary on their merit. Finally, f) represents the case where a design idea was previously considered and rejected. Further, the likelihood of each being implemented was discussed.

#	Design Idea	Bug	Lost in Comm.	Good Idea	Possible Idea	Bad Idea	Avoided	To Be Added?
1.	Make keyword and browsing result views the same.	x						Yes.
2.	Use term indexes for keyword search, not exact string matching.						x	No. Cost of further server software.
3.	Highlight keywords in search results.		x					Yes.
4.	Add spelling suggestions.		x					Yes. Especially important in jargon-filled yachting domain.
5.	Add query expansion suggestions.				x			Maybe. Needs testing, as domain may be too narrow.
6.	Add query operators (e.g. -term).					x		Unlikely. Shop is so small. Not much benefit.
7.	Allow multiple selections in facets (i.e. see books and CDs).				x			Maybe. Not much benefit in small domain. Maybe instead do group by. Not much penalty for viewing one then the other.
8.	Remove any selection from breadcrumb, not just last selection.			x				No. To much developer effort for small interaction benefit.
9.	Turn metadata in result list into active filtering links.		x					Yes.
10.	Vary metadata in result list, so it says subject, rather than product type, if user selected a product type.			x				No. Each item is related to many subjects, so too much metadata to show in results list.
11.	Show search term when viewing keyword search result			x				Yes.
12.	Show similar results when viewing shopping basket			x				Yes, but the plan is to encourage membership (a special product) at basket time instead.
13.	Make metadata in shopping basket a selectable filtering link.		x					Yes.
	Totals	1	4	4	2	1	1	7 Yes, 2 Maybe, 4 No.

Table 4.5: RYA design recommendations as classified by the information architect on the project.

Participant's Perspective of Sii

"I definately take [Sii] very seriously."

The IA here saw Sii as a tool that would provide visibility to areas for possible improvement. From the classifications above, two conclusions were drawn. First, Sii provided a rigourous checking method for many features, which were intended for the design but had been lost during the course of the year long project. One known bug was found, but further ideas were added to the bug list because of the systematic process provided by Sii. The IA noted that, especially for smaller projects, the designs proposed were based on experience, knowledge, intuition, and informal use of Nielsen's heuristics (Nielsen and Molich, 1990). The value provided by Sii, in the case of these miss-communicated ideas, was that it complemented their otherwise less rigourous methods. *"[Sii] provides a formal process to something that less formal"*.

The second conclusion drawn from the findings in Table 4.5, which further supports the idea of a checking process, is that not all good ideas will be implemented. Three key reasons were provided for not implementing ideas, some of which were even considered as good ideas. The first reason for not implementing a feature was the cost of additional server software or hardware. The design ideas were often considered in the light of the current back-end technology being used by the website. Another design idea (#8 in Table 4.5) was that the challenge of implementation was deemed greater than the benefit provided. The final reason provided for not implementing a feature was over-crowding the interface and adding too much complexity. Sometimes this reason was based on the size of the domain and shop, such as #6, and sometimes this was based on the visual effect caused. Upon further discussion, the domain size reason was used when an idea was a good idea for a larger domain size, but the visual rejections would usually hold true for larger domains (like #10).

The other value described by the IA, relating to the rigour of the process, was that it forced the evaluator to consider different user perspectives. In this example, the IA valued knowing that it could be improved for users who were learning to use the site. *"I like that you can see that learning can be improved. [Sii] prompts you to think about improving [learning]. That applies to all [of Belkin's dimensions]"*.

Use in Context

The IA saw two phases, in particular, where Sii would provide *checking* value: the design of wireframes [14] and, and before the implementation goes live. Despite being a rigourous and short to apply, the IA expected that it would not be used on smaller projects, not because it wouldn't provide value, but because clients would not have paid for any evaluation. The IA did not expect that the use of Sii would slow down any

[14] A form of paper prototyping used in industry.

normal processes within the company, but would instead enhance the existing design processes used.

The use of Sii for communication was also discussed. Like the previous case study, the IA commented on the use of complex language in the interface. For communicating with his boss, who understand that usability is important, terms like 'BIBBLE' from Bates' tactics, would provide very little common ground for discussion. For communication between the company and clients, and even between staff, the IA suggested that the language should be much more business oriented, regarding improvement of online sales figures as the focus. *"Its difficult to find the time to go over [the definitions]... [Sii] needs to be in business facing language... [Sii] needs to be not jargon."*

Like in the previous study, the structure of communication with colleagues and with clients was discussed. In further emphasising the point of business facing language, the overview was: *"Talk to them - give a good introduction - use plain English"*. The process was further broken down into a) the problem, b) a reference, and c) some suggestions for improvement. Notably this process involves the reference to another system which both the design team and the clients believe to be a good example. This suggestion may have been in light of the comparison previously provided with Google during the first interview.

Summary

The IA involved in this case study saw great value in the analyses provided by Sii, and in particular for providing a rigourous method to an otherwise informal process. In particular, the analyses in this case highlighted a number of bugs and features that were intended for the design but had not materialised during the project. The project was, in this case, already completed, and so Sii could have provided these discoveries before going live and avoided the need to make changes after deployment. This case study also provided insight into when the Sii analyses would be approved, but not implemented. Budgetary, development effort, and visual constraints all had impact on whether re-designs would be implemented, and in relation to the size and scope of the online store. In particular, the IA noted that several ideas would unnecessarily crowd the interface, given the scope of the RYA shop, and that maintaining simplicity was important. The first extension discussed in Chapter 5 discusses these issues in more detail. Notably, the IA was keen to try the complexity-focused extension when ready.

Sii was mainly seen by this IA as a solo-use tool, rather than as a tool to discuss and communicate design requirements. Where the previous two examples were with individuals either working in or near academia, this case study focused more closely on need in industry, where business and sales oriented analyses were more important. This provides part of the motivation for studying the use of language within the framework in the future work section (Section 6.2).

As a final point of summary, the IA was keen to use Sii in future projects. Typically, within a year long project, the wireframe stage would happen between 1 month and 2 months in, and Sii would then be valuable but for around a week. Notably, the company was not involved in any projects that were at this right stage. Further, the IA expected that catching projects in any similar companies at exactly the right designing stage would *"be like catching a butterfly"*. Catching projects at this right stage is discussed further in the future work (Section 6.2). Another previously deployed site, however, will be analysed and discussed in the near future, as the company prepares for a follow-on contract.

4.3.1.4 Summary of Case Studies

Three case studies were discussed above, regarding: 1) an academic project, 2) a consultant working with multiple clients, and 3) an Information Architect working within a web design company. Each of these case studies provided unique insights into the value of Sii and the constraints of use in context. All three presented strong evidence for Sii's suitability for regular use. Two consistent themes, however, were discussed: trade-offs and language.

Trading off the benefits of functionality against the complexity added to an interface was noted, in particular, by the academic and the information architect. This was further backed up by the trade off against the effort and budgetary constraints discussed by both the consultant and the architect. Several design suggestions were rejected by the information architect in particular that would have made search result lists untidy or overloaded with metadata. These findings provide both development requirements for future work (Section 6.2), but also motivation to analyse the complexity costs of adding features. Section 5.1 addresses this need directly.

The second common theme was in regard to Sii's use of language. In particular, the industry motivated information architect noted that learning the technicalities of the method was a potential barrier for using Sii to communicate ideas to others. Informally, however, the consultant noted that the Sii framework would require careful introduction to clients regarding its purpose and then its findings.

4.3.2 Pilot Studies of Usage

To further inform the understanding of how the Sii might be used in practice, several pilot studies were performed, which studied the practical issues of applying the framework. Each of the pilot studies addressed stages of using the framework, and the aim of performing them was to identify areas of the Sii procedure that require focus during future work. The studies are considered as preliminary pilot evaluations because they were performed in simulated contexts by computer-science graduate students and research assistants. While computer scientists may be representative of the population that would use the framework, Gray and Salzman (1998) have previously discussed the limitations of trying to generalise from studies of graduate students, using UEMs in fictional scenarios, to the experiences of practitioners. Although the pilot studies below provide insight into the nuances of applying the framework, they may not be indicative of carefully thought through, attentively applied usage by evaluators working on projects with real stake-holders. Chapter 6 discusses the future work planned in studying real usage of the Sii framework.

Initially, the pilot studies included three participants whose first language was not English. It quickly became clear that the learning curve for understanding the Sii procedure was significantly longer when crossing a language barrier. Even with additional time provided to explain framework and requirements of the three pilot studies, the foreign-language participants produced notably smaller and less consistent contributions. Consequently, the results of these participants were excluded from the pilot studies, and the preliminary evaluations proceeded having removed any variation created by language barriers.

4.3.2.1 Pilot Study 1: Identifying Interface Features

After choosing the interfaces that are being compared or analysed using the Sii framework, the second preparation task is to identify the features that are contributing to the design. The aim of the first pilot study was to see if participants could accurately identify appropriate interface features of a familiar interface. The hypothesis was open, in that no prior expectations were held as to the resulting consistency.

Method

Seven participants were set the task of identifying the contributing features of the Google Web search interface. Google was chosen for its familiarity, so that participants could focus on learning about the task requirements. Before beginning the task, participants were presented with an example analysis of the interface features presented in study of three faceted browsers described in Section 3.2.3. Participants were told that they did not have to choose from this list, and that the list of features in Google may be entirely different. Participants were then provided with a computer that had Google pre-loaded, and with a user logged into a Google account so that the personalisation features were active. The task explicitly excluded, however, the alternative searching modes, such as Google Shopping and Video search, and the additional search options, such as the Wonder Wheel[15], shown in Figure 2.15 on Page 31, that were introduced after the beginning of the study. Participants were then given an unconstrained amount

[15]http://www.google.com/support/websearch/bin/answer.py?hl=en&answer=142143

of time[16] to identify and name, by writing on paper, the features provided by Google. Participants were then debriefed with a short interview, which included a discussion of the task's challenges.

Results and Discussion

The number of features identified by participants ranged from 11 to 15. In total, however, 24 features were identified, as shown in Figure 4.12. The graph shows that only three features were identified by every participant, although 4 of these 7 participants took the basic keyword search and results lists for granted and did not explicitly include them in their original list. These participants made it clear during the debrief, without prompting, that the two features were considered as given with the Google search. In the context of using Sii, rather than simply listing features on paper, participants may have been more inclined to include them explicitly in the feature list.

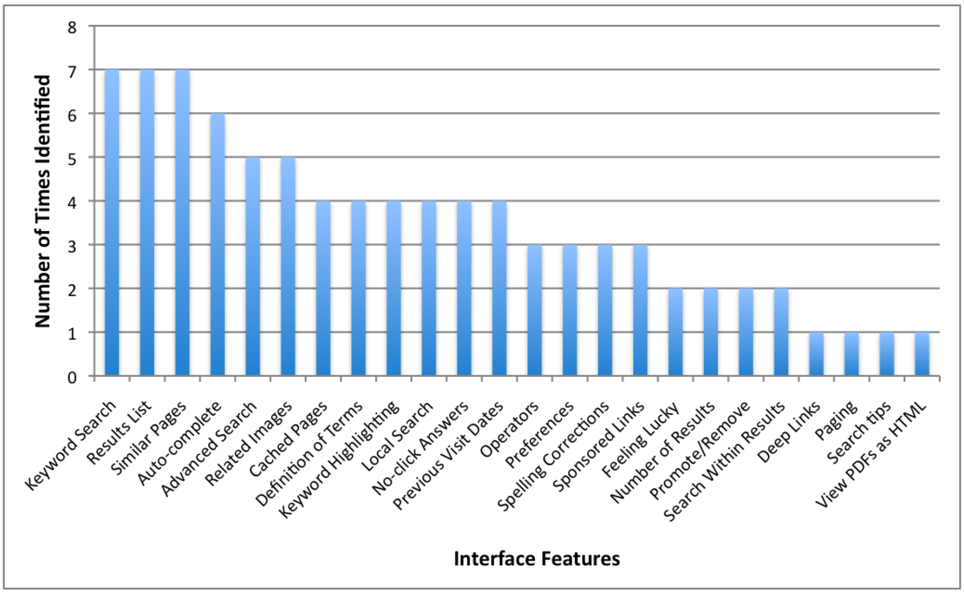


FIGURE 4.12: Graph showing the features of the Google Web Search interface identified by 10 participants.

Notably, the more frequently identified features are those that are more visually prominent in the Google interface, such as the Similar Pages link provided with every result, and the Auto-Complete provided as the user enters a search query. Some of the least frequently identified features included the ability to view PDFs in HTML format, and viewing the Search Tips that Google provides a link to from the footer of each page.

Another notable theme in the results is variation in granularity chosen by participants, as discussed in Chapter 3. Some considered paging to be a separate feature to the way that results were presented. Similarly, some participants chose the bold highlighting of query terms in each result view to be a separate feature to the representation of results. Finally, from the discussion of the debrief, some participants considered the special operators,

[16]Most participants spent between 5 and 10 minutes on the task.

such as using the minus sign to remove results relating to a chosen query term, as part of keyword search, while others decided this to be a separate feature. This variation is notable in the examples discussed so far in this thesis. The initial comparison of three faceted browsers in Section 3.2.3 considered elements such as changing selections and adding selections in the facets to be seperate features, where as the analysis of three faceted browsers discussed in Section 4.2.2 did not. Often the choice of features used in a Sii analysis will depend upon the differences between the designs being compared. In the first analysis of faceted browsers, it was the intricacies of the implementations that differentiated the three browsers, where as the second example compared browsers that differed at a higher level.

Finally, some participants, during the debrief, discussed whether some features should or should not be considered as contributing to the search process. From Figure 4.12, we can see that some participants considered Sponsored Links to be a contributing feature, where as others described such links and adverts as simply an annoyance that would confound the search process. This element of the discussions shows that some reasoning and decision should occur to decide which of the identifiable features in an interface should be considered as supporting search.

Summary

This pilot study has shown first that users cannot objectively decide what constitutes a feature of a search interface and second that there will likely be variation in the features chosen by different evaluators. It is also clear, however, that not all identifiable features should necessarily be included in a list of features that support search. In respect to the results of this pilot study, some future work will likely focus on improving the guidance provided to practitioners in choosing features to compare during the use of Sii.

4.3.2.2 Pilot Study 2: the Procedure of Applying the Framework

The aim of the second pilot study was to analyse how consistently evaluators would estimate the counts of how many *moves* it would take to achieve a certain *tactic* with a feature. Essentially, this study was of the application process described by Figure 3.2 in Section 3.1.2. Like the previous pilot study, no specific hypothesis was held over whether the data entry would be consistent across the participating evaluators. Instead, the aim was to investigate what issues would be experienced by users trying to apply the framework, when using it for the first time.

Method

Seven participants took part in the pilot study, which began with a 30 minute explanation of what the framework was designed to do, and how it should be used, using the comparison of faceted browsers from Section 3.2.3 as an example. The participants were then asked to evaluate a sample of the features identified within Google for supporting a sample of the *tactics*. In total, participants were given 45 minutes to evaluate the support for 15 random tactics by the following user interface features: the display of results, the auto-complete, the spelling corrections, the minus-sign operator for excluding results relating to a certain term, and the 'similar pages' links. User's were provided with both the full definition of the 15 tactics, and the short definitions provided on the Sii website. Participants were provided with a table printed on paper for writing down the number of moves required to achieve each of the 15 tactics with the 5 interface features. The session was concluded with a group discussion of the task and its challenges.

Results and Discussion

Of the potential 75 judgements that could have been made in the 45 minutes, the highest achieved was 64. The lowest achieved of those who entered valid data, was less than 10. Two participants, however, entered answers such as 'yes' and 'no'. Of the four that provided at least 20 judgements, the average number of answers submitted was 42. There were 25 cells in the results table that all four participants made judgements for. The average correlation of these 25 points between participants, however, was 51%.

Instead of focussing on the correlation achieved by those able to begin using the framework, the overriding conclusion that can be drawn from this pilot study is that, for some participants, one hour and 15 minutes was not enough time to begin effectively performing Sii's application procedure. Although it was originally intended that the task be performed from reference material alone, after the initial description of the framework, it quickly became clear that additional support, by answering questions, was required for any significant number of judgements to be made. Even the participant who submitted the most judgements, was only able to do so by proactively asking for help. Those who produced no desired results asked very few questions. Although there are very few elements of the procedure to learn, the first conclusion drawn, therefore, is that the learning curve should be studied and the language used to describe the procedure addressed.

From the group discussion that followed, two key challenges experienced during the learning curve were identified. The first of these challenges was in maintaining a delineation between the features being assessed and the other features of the Google interface. It was hard, for exmaple, for some participants to decide how auto-completion, independently of the keyword search box, supported tactics. This challenge may be easier if preceded by the analysis and description of the parts involved in the preparation tasks. Had the participants chosen and defined the features of Google themselves before the study, then this task may have been easier. Notably, in the previous pilot study, some participants found it hard to decide if features contributed to the support for search.

The delineation of features within a search interface clearly presents a challenge for assessors when using the Sii framework, and it may not become clear to them why they are doing so until they begin applying the framework.

The second key theme within the group discussion was that learning the tactic definitions was a major hurdle. The majority of questions asked during the task were in relation to the definition of these tactics. Many asked for examples of how the tactic applied to other parts of Google, or to parts of the comparison described before beginning the task. Several participants noted that it was easier to begin the analysis using the shorter definitions, despite having lengthier academic definitions available. The group agreed, however, that these short definitions would be best supported by a continuing example. In support for the arguments against using computer science students and researchers rather than a sample of the expected user base of practitioners working on search interfaces, the consultant in the second case study described above was able to accurately estimate the definition of the majority of Bates' tactics from their name alone. It may be that the learning curve for the expected user base will be much faster than those involved in these pilot studies.

Summary

Although this pilot study was designed to investigate how consistently multiple evaluators, working independently, could assess the same interface using the Sii framework, the findings were more oriented towards supporting first-time users through the learning curve. The results provided some indication both of a) how long the learning curve might take, and b) what modifications to the language used within the framework might reduce the learning curve. Although many of the participants struggled to perform any analysis in the first hour's experience with the framework, it is encouraging that techniques which are considered relatively simple, like Heuristic Evaluations are often taught in half-day seminars (Law et al., 2009). Further in a comparison of the SUE method to Heuristic Evaluation, participants were taught each method for 2 or 3 hours (De Angeli et al., 2000). It is also encouraging that relatively low expertise is required to begin using the framework within a short period of time, which is considered another trait of "easy" techniques like Heuristic Evaluations.

The Sii website already uses the short definitions of the 32 tactics, and also provides several example analyses. As part of the future development of the system, described in Chapter 6, an example will be carefully integrated into the application procedure.

4.3.2.3 Pilot Study 3: Interpreting the Graphs

The third pilot study was designed to investigate the final stage of using the Sii framework: interpreting the graphs for producing potential design improvements. Like the previous two pilot studies, the aim was to assess the consistency of interpretations produced by different evaluators when viewing a single set of analysis graphs. Also, like the previous two pilot studies, no assumptions were made about the potential consistency, or what would improve it, but the study was designed to investigate what factors affect the interpretation of Sii's output graphs G1-G3.

Method

11 participants were involved in the third pilot study, where masters-level students worked together in a group of 7, and graduated researchers participated individually. A relatively simple example, the Backward Highlighting evaluation described above, was used as the common analysis across all participants. After describing the point of the framework and the three graphs it produces, the two variations of the the Backward Highlighting technique were demonstrated to participants. Participants were then given up to half an hour to discuss the graphs, what the graphs told them about the designs, and how Backward Highlighting might be redesigned. Finally, a short debrief was performed to discuss the challenges experienced in performing the task.

Results and Discussion

All four individual participants, and the group together, were able to quickly draw accurate conclusions from the graphs. Example statements, which demonstrated that participants understood what each graph was designed to visualise, included "WEIGH must be better supported because the highlights are grouped together" and "The support seems to tail off for users who know what they are doing".

All participants but one were able to translate the analyses in the graphs into potential design changes, but the group of student participants produced the single largest list, including six design changes. In fact, the discussion held within the group provided the most considered, as well as the largest number of, design suggestions. This finding provides some evidence that the results of the Sii analysis should be discussed in a group. Heuristic Evaluations are often discussed by multiple independent evaluators, and Cognitive Walkthroughs are usually performed and discussed as a group.

Of the design changes suggested by the remaining three individual participants, who suggested around three each, all but one were discussed by the group of student participants. This finding indicates that discussing the analyses as a group might also provide the most complete set of potential design changes, as well as the most considered design suggestions.

The majority of re-design suggestions were prompted by missing or weak support for certain tactics. The group, and all three individuals focussed on the poor support for the CUT and BLOCK tactics, and most then suggested the same design change for each tactic. To support the CUT tactic, in most significantly cutting down the number of results, it was suggested that the highlights be varied in color, or hue, to indicate number of related results. The group, and one participant, noted that there was already a heavy use of color in the interface, and that simply ordering the highlights by weighting, or using a bar-graph style visualisation might be more appropriate. To support the BLOCK tactic, in excluding results related to a certain term, the majority of participants suggested that some method of rejecting a highlight should be provided, either through a [x] button or a right-click menu.

Reflecting on the task

The majority of users initially indicated that the graphs were daunting. Many later attributed this feeling to the number of tactics displayed in Graph G2. In reflection, the majority also felt confused over how to make use of the user profile graph, G3, as it was not possible to tell which tactics to support better in order to support the user types better. This indicates that a) it is important to see, as suggested by consultant in Case Study 2 above, which tactics relate to each user profile, and b) that design revisions typically relate to the discussion of support for tactics.

The majority of participants, including those that voiced opinion within the student group, suggested that participating in this example analysis was enough to go away and interpret similar graphs on their own. Several participants noted that they would still require the definitions of tactics and user types for reference, two suggesting that maybe no amount of experience would allow them to memorise all 32 tactic definitions. In this regard, a pop-up link is now provided by each graph in the online Sii framework for the timely display of definitions.

Finally, the group of students discussed the use of this method in comparison to performing user studies. Some suggested that the method was a welcome break from working with potential users, while others said that the lack of involvement with users meant that it would be very difficult to choose between potential design revisions. One student participant, who happens to be the same participant who provided the most judgements in Pilot Study 2, suggested that the Sii framework could be used well in conjunction with user studies. In particular, the Sii method would provide the richest insight into designs and potential changes, but user studies would support designers in choosing which design directions to take.

Summary

Aside from identifying a few spelling mistakes on the online framework, this pilot study showed very promising results for the value of using Sii. In particular, it was valuable to see the difference between a group of assessors working together compared to individual contributions. This finding suggests that regardless of how many people apply the framework, the results should be presented and discussed as a group. Further, it is encouraging to see that the 6 design suggestions were generated by separate participants when using the same graphs.

4.4 Summary of Validation

This chapter has validated the Sii framework, as far as possible during the three year period of doctoral research in the United Kingdom, using three approaches: a) internal validation of the way the framework was generated, b) external validation of the results it produces, and c) contextual validation of the practical elements of using the framework. It may take, by following the precedence of other usability evaluation methods like the Cognitive Walkthrough, 5-10 years before we can estimate the impact that this method will have on industry and academic. For now, however, this research has focused on ensuring that the framework has been properly built, provides accurate results, and can be easily integrated within current working practices.

Section 4.1 used two methods for validating the way that the framework was built. The section began with a discussion of the models chosen and the available alternatives. Factors such as citation count, academic reuse, and appropriateness of scope were used to compare the available options. Second, Section 4.1 presents the method used for validating the mapping that joins the models used within the framework. The mapping constitutes part of the contribution of this doctoral work, and so the final mapping was reviewed by independent judges. One product of this review was a revised mapping that was agreed upon by all judges.

Section 4.2 validated the accuracy of the whole framework, including the results produced by the revised mapping, to show that Sii provides results that correlate closely with the finding of empirical user studies. The majority of the results found in the academic evaluations could be seen within the three graphs produced by the Sii framework. Further, the framework provided insight into the potential reasons behind surprising results of the user studies, and their remaining open questions. Two more example Sii analyses available in this thesis. First, Chapter 5 describes an extension of the framework that allows it to analyse collaborative search interfaces, where groups of people actively collaborate on a search problem. This example further speaks to the validity of results. Finally, another example is included in the appendices of the documents.

Section 4.3 studied the practicalities of the Sii framework from two perspectives. First, short case studies of practitioners were presented to discuss contexts of usability evaluations such as fitting in with working practices, communication of ideas, and relationships with clients. Furniss et al. (2008) provided evidence that contextual issues such as these can be important in the uptake of new usability evaluation methods. Further, a series of small pilot studies were performed to identify the potential challenges that first time users of the framework might experience.

Despite successfully answering the majority of the research questions[17], and providing strong evidence for the validity of the framework at this early stage, there is still plenty

[17] the remaining research questions are addressed in Chapter 5

of research that can be performed on the Sii method. Chapter 6 describes the planned future work in more detail, but in particular the validity of the Sii framework will be supported by case studies of practitioners working on their own projects that have real budgetary constraints and stakeholders. This future work is preceded, however, by the investigation into potential extensions to the Sii framework that might further support practitioners using the Sii framework.

Chapter 5

Conditional Extensions to the Sii Framework

> *Increasingly, people seem to misinterpret complexity as sophistication, which is baffling - the incomprehensible should cause suspicion rather than admiration.*

Niklaus Wirth, Designer of Pascal and Turing Award Winner

> *A general principle for all user interface design is to go through all of your design elements and remove them one at a time. If the design works as well without a certain design element, kill it.*

Jakob Nielsen, Principal of the Norman Nielsen Group, from his book: Designing Web Usability

The previous two chapters have described and validated the Sii framework, which can be used to assess the functionality of an Information Seeking Interface (ISI) for how it supports a range of known search tactics and from the perspectives of multiple searcher profiles. The end of Chapter 4, in particular, was focussed on understanding how the framework works in practice. As has been demonstrated by the development of several other Usability Evaluation Methods (UEMs), and discussed in Chapter 4, it can take 5-10 years to understand the impact that a new framework, like Sii, will have. Notably, though, during the early stages, the creators of new methods have typically tried and tested potential variations, modifications, and extensions. The GOMS method, for example, includes several variations that focused on aspects such as parallel actions, and cognitive processes. John (1990) discusses the extensions developed for the GOMS method in more detail. Similarly, several variations of Heuristic Evaluations have been developed for, for example, mobile and pervasive techology (Mankoff et al., 2003).

As the on-going work into actual use of the Sii framework, discussed further in Chapter 6, begins to bring back results, this chapter addresses some of potential limitations that have been noted so far. Section 5.1 responds to the notion that the Sii framework consistently rewards the addition or improvement of search functionality without reference to the complexities that they might add to the interface. Several directions for extending the inspections provided by the Sii framework are considered, and the potential for using Cognitive Load Theory (CLT) is investigated in more detail.

Section 5.2 provides a detailed investigation into a recently popular theme in Information Seeking (IS) called Collaborative Information Seeking (CIS), where people actively search together in identifiable and small groups. Most of the existing IS theory assumes that individuals are searching independently of any other users. CIS, however, notes that sometimes small groups, such as family members or colleagues, share a search goal, like planning a group trip, but are forced to work independently. The theory underlying the Sii framework was largely informed by the individual contexts of searching, and so Section 5.2 revisits the models in terms of a shared seeking context. Further, the section applies a modified version of the framework, using the revisited theory, to a popular CIS interface called SearchTogether, showing that Sii can still be accurately applied to such conditions.

The chapter concludes with a discussion of other potential extensions, and how they should be built in order to integrate with the Sii framework. Notably, quantifiable theory, such as the bounded models used so far in the framework by Bates (1979b,a) and Belkin et al. (1993), is usually required in order to build reliable predictions.

5.1 Managing Interface Complexity

Applying the Sii framework to IS interfaces (ISIs) may help identify certain tactics and searcher profiles that are under-supported. Typically there are two means to correct any such weaknesses: re-designing the existing features or including additional functionality. The former of these is usually the ideal, over simply adding new functionality to a system, which can lead to scope creep and introduce unnecessary features that over-crowd an interface. One concern of the framework, as discussed in the first case study described in Section 4.3, is that the framework will always reward the inclusion of additional functionality, while not accounting for the affect it may have on the complexity of the interface. One result not modelled by the framework, for example, was that study participants performed some simple search tasks more efficiently with a baseline keyword search interface, instead of the experimental conditions designs to improve support for search Diriye (2008). Further, in the third case study, the information architect decided not that some potential design revisions would not be suitable, as they would over-crowd or over-complicate the user interface.

While many methods have focused on enabling and supporting user behaviour, others have also focused on the simplicity and intuitiveness of interface design. Notably, Cognitive Walkthroughs focus on the learnability of a system from a novice user's perspective. Further, the checks in Heuristic Evaluations include: use of natural everyday language, and a careful use of white-space. In fact, the notion of simplicity appears in several design philosophies, including the Laws of Simplicity (Maeda, 2006), The Beauty of Simplicity (Karvonen, 2000), The Design of Everyday Things by Norman (2002), and Designing Web Usability: The Practice of Simplicity (Nielsen, 1999).

Examples of Simplicity

The iPod is perhaps one of the clearer examples of where a simple clean design, rather than significantly new functionality, has led to Apple's current majority in the mp3 player market[1]. Similarly, of the search engines available online, Google has been the most popular since December 1999[2]. Like the iPod, Google purposely has the least cluttered interface, compared to engines like Yahoo! (Tischler, 2005). Google's front page, which almost only provides a search box, guides users directly to the main service it provides. Microsoft's search engine, MSN Search at the time, changed to a similar front page in July 2004. Similarly, Ask Jeeves[3] made a slow transition to become notably similar to Google around 2002. Yahoo! still provides a much busier interface containing the latest news, sport, entertainment, adverts, and links to a diverse set of related yahoo services, such as Flickr[4].

It might also be noted that Google has not added faceted search, a feature involved in many of the interfaces discussed in this thesis, to all of their search services. Potentially, the addition of such a feature could support users during web search, but might also clutter or over-crowd their famously clean design. Instead, Google has added faceted search selectively to their services, most notably appearing in the product shopping service, as shown in Figure 5.1. It is possible that Google has identified certain conditions where the benefits of faceted search outweigh the costs to adding complexity to their design.

Balancing Functionality and Design

Interface designers are faced with the challenge of trying to estimate which design options are better for the user, trading increased functionality against simplicity. While, as mentioned above, some HCI techniques can be used to help designers maintain simple and intuitive interaction, this first extension builds a measure of complexity directly into

[1]AppleInsider reports, sourced from Credit Suisse, that the iPod began dominating the mp3 player market in 2004 (http://www.appleinsider.com/articles/06/05/24/ipod_how_big_can_it_get.html - viewed 23-MAY-2009).

[2]Google's press release citing a quarterly release from the NPD group, describing a survey placing Google first overall among 13 search engines for user satisfaction and loyalty. Google held the largest index, on and off, from August 2000 (http://searchenginewatch.com/2156481 - viewed 23-MAY-2009).

[3]http://www.ask.com

[4]http://flickr.com

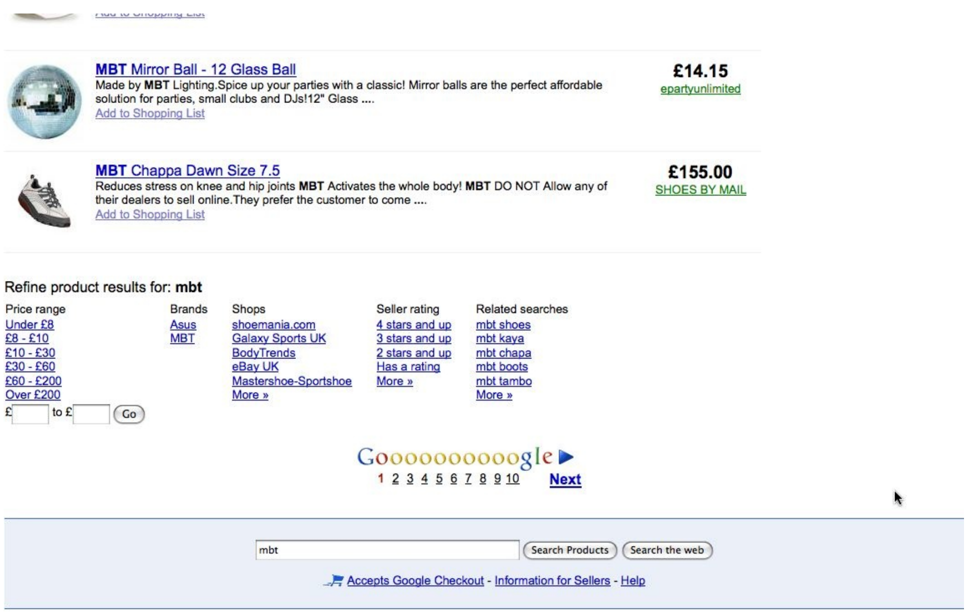


FIGURE 5.1: Google provides facets at the bottom of their shopping service only, allowing users to refine by facets that include: price, brand, and retailer.

the inspection framework. The aim is to support designers in simultaneously analysing search interfaces for the amount of functional support they provide and then how complex they are (Wilson and schraefel, 2008b). First, related research concerned with reducing complexity is reviewed. Cognitive Load Theory, which aims at identifying and producing means of reducing times of increased load in working memory of the brain, is investigated in more detail. An extension to the framework is then built, which operationalises Cognitive Load Theory. The extension is then applied to three studies considered in the preceding chapters of the thesis.

5.1.1 Related Work in Managing Interface Complexity

The majority of the ‘simple’ design philosophies, mentioned above, promote the use of careful user-centred practices in order to improve simplicity. Using techniques such as personas and scenarios and task analysis, for example, can help designs provide timely support to users as they achieve certain tasks.

Flow during interaction

One theory in promoting such timely careful design is Flow Theory (Csikszentmihalyi, 1990), which analysed the factors that affect situations where users feel immersed in their activities, including distorted perception of time and loss of self-conciousness. Systems and tools, therefore, have to fit suitably with their intended activities, to maintain a sense of flow, rather than interrupting it. Ghani and Deshpande (1994) studied the using optimal flow theory in human-computer interaction design, discovering that a user’s own perception of their control over software was key.

In using the searcher profiles, the subsequent work by Belkin et al. (1995) also suggested that the flow of interaction with an interface is important. In their work, Belkin and colleagues created 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), and the others discussed in Section 2.1, define several stages that users typically follow when completing an information seeking task.

Instead of the Sii 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. As discussed in Chapter 2, however, the information seeking process is not particularly linear and so it may be hard to use flow directly. Further, the work by Ghani and Deshpande (1994) could be used to support the Sii framework, as increased functionality might support control.

Engaging and Disengaging

Another similar 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). After performing a detailed literature review, including Flow Theory, O'Brien and Toms (2008) identify 10 factors of engaging in a task with an interface. Further, they noted that these factors varied according to different contexts. Gaming, for example, should be challenging, but shopping should not. After performing a set of interviews, O'Brien and Toms (2008) produced a model of engagement that included attributes that affect the point of engagement, the period of engagement, and the reasons for disengaging. Reasons for disengaging with a system included usability, challenge, and interruption. Later, O'Brien et al. (2008) performed a detailed survey of online shoppers and were able to, using factor analysis, reduce the number of key factors in engaged interaction down to 6: attention, usability, aesthetics, educability, novelty, and involvement.

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, would provide strong features in a way that would encourage engagement and discourage disengagement. The output of the research so far has been a subjective questionnaire about experience, used to judge how engaging a system was when it was being used. Unfortunately, the theory of engagement here is not well-enough defined or quantifiable in a way that can be used in an inspection method.

Workload and Effort

Another stream of research has been into assess workload and effort. NASA designed a Task Load Index scale, referred to as the NASA-TLX, to assess the mental workload experienced by operators while achieving tasks (Hart and Staveland, 1988). Like the Engagement scale produced, the NASA TLX is performed by asking participants of a study to introspect their previous experience and provide subjective ratings on 6 scales: mental demand, physical demand, temporal demand, performance, effort, and frustration. Answers are multiplied by weightings and summed to produce an overall score. The authors note that such a subjective measure is quicker and easier to apply and the earlier SWAT method, which stands for Subjective Workload Assessment Technique (Reid et al., 1982). SWAT requires participants to card sort the definitions of three points within three scales of: time pressure, effort, and stress, which are then converted to a 100 point scale.

More recently, another subjective method, called the Workload Profile (Tsang and Velazquez, 1996), provided yet another subjective rating form, but based on the models of human working memory. Rubio et al. (2004) compared Workload Profiles with the NASA-TLX and SWAT scales, and demonstrated that the Workload Profiles performed best. Unfortunately for the Sii framework, these scales are both subjective and retrospective, where as, for use in an inspection method, theory is needed that is predictive and, ideally, more objective. One such potential theory, which is also based on human working memory, is Cognitive Load Theory (CLT).

Put simply, the notion of CLT is that the complexity of a learning task *and any learning material* both affect the users ability to gain the knowledge they seek (Chandler and Sweller, 1991; Chandler and Sweller; Paas et al., 2003a,b; Mayer and Moreno, 2003). Further, learners have a limit, although it is considered that effort can change this limit (Paas et al., 2003b), and learn, the load imposed by the task and the learning material must fall within this limit. Finally, additional space within this limit is used to commit information to long-term memory. The benefit of this style of approach is that it a) it focuses on how the design of learning material affects ability to learn, and b) that empirical tests have been used to demonstrate 9 variables in design to reduce the load imposed by learning materials. Consequently, CLT can be used within the Sii framework, because it can be used to inspect designs for how they fare against these 9 factors in order to predict the load imposed by learning material, which in the case of modern information seeking, is the search interface.

The next section describes CLT in more detail, including its basis in models of working memory, and the 9 ways that interface design can reduce the load imposed by user interfaces. For the Sii framework, operationalising such a theory means that designers could measure the cognitive load added by new functionality, and decide if the new feature is worth the change in load. Designers would be supported, therefore, in the ability to trade off functionality and complexity and choose designs that are the most supportive and the least confusing.

5.1.2 Cognitive Load Theory Defined

CLT is based upon a model of human working memory, which is also described after the initial description of cognitive load theory. Assuming that working memory has a limit, CLT notes that there are three factors to the state of someone's cognitive load at any one time, as shown in Figure 5.2 from (Paas et al., 2003b).

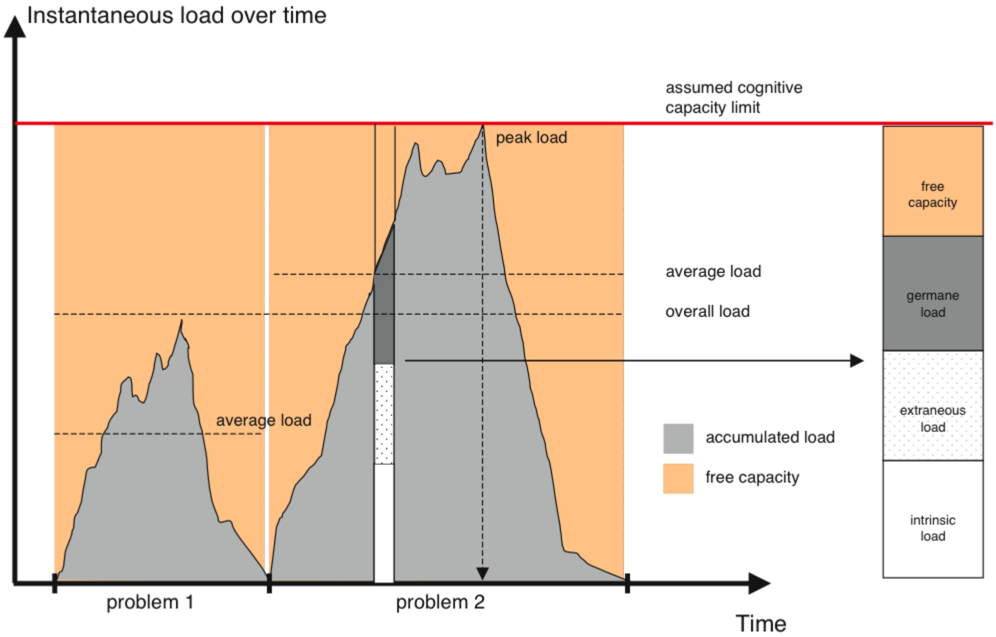


FIGURE 5.2: Cognitive Load, which varies over time, occurs within limited capacity, and is made up of intrinsic, extraneous, and germane load.

The complexity of a learning task is called *intrinsic* load, and, during a task, is considered to be more of a constant variable. Second, the load imposed by materials given to support a learning task is called *extraneous* load. Badly designed learning materials, therefore, further impede users in achieving their task, especially if they cause the cumulative load to breach the assumed capacity limit. Further, it is noted that for users to commit facts to long-term memory, *germane* load is required. Consequently, if the cumulative intrinsic and extraneous loads are too high, then there may not be enough space to apply germane load and commit what is being learned to long-term memory.

Learning materials should aim to support users no matter how much intrinsic load their task requires. If a problem is too big for working memory, then learning material should support users in breaking the task down into steps, each with lower intrinsic load. Further, in designing learning materials, they themselves should impose as small an extraneous load as possible on users. If the extraneous load is high, then only tasks with a low intrinsic load may be achieved. Ultimately, however, both need to be reduced to make space in the overall cognitive load, for germane load. According to CLT, although space for germane load can be produced by minimizing intrinsic and extraneous load, the design of learning materials, as well as the effort committed by learners, can effect whether or not the space is used for germane load.

Applying Cognitive Load Theory to Design

So far, CLT has been mainly designed to understand how instruction manuals, for example, can be better designed to teach people to use machinery or computers (Chandler and Sweller, 1991). In these scenarios, the task has been to learn how to use a computer and the learning materials have been provided in book form. Learning, however, is often the same task held by exploratory search users, except that the material they have to support them in achieving their goal is an ISI. Ultimately, the user is still aiming to learn something, and has resources to help them do it, and so part of operationalising this theory is in proposing that CLT can be applied to understand the complexity of search software. Such a proposition is supported by Mu (2004), who, states 'cognitive loads are closely related to the complexity of a task, the system used to operate the task, and the operators characteristics, which makes no indication that 'the system need be instructional paper documents. Further, others have considered how CLT might help interface designers convey search result relevance (Hu et al., 1999) and explain why users rarely provide relevance feedback during search (Back and Oppenheim, 2001).

Nine factors that affect cognitive load have been identified and summarised by Mayer and Moreno (2003). Each of these factors, and their impact on cognitive load, have been identified through empirical methods such as secondary task analysis, where reduced performance in a monitoring task indicates increased cognitive load of the main learning task. In order to understand these nine factors, though, models of working memory need to be quickly reviewed. The aim here is not to discuss, critique, or validate models of working memory, but to explain the key elements in order to further discussion of CLT.

5.1.2.1 Working Memory

As part of explaining the 9 ways of reducing cognitive load, Mayer and Moreno (2003) reviewed a model of working memory that accounts for processing various forms of multimedia, originally from Mayer (2001) and shown in Figure 5.3. The diagram is based on the widely accepted model of working memory produced by Baddeley and Hitch (1974), which suggests that working memory is made up of two separate auditory and visual processors, called the Phonological Loop and Visuo-spatial Sketchpad respectively, controlled by a 'Central Executive' processor. The notion that visual and auditory information were processed separately, however, was also noted in Dual-coding theory, produced by Paivio (1969). Mayer suggests that as visual and auditory information arrives from sensory memory, it is processed separately by working memory and integrated with knowledge from long-term memory.

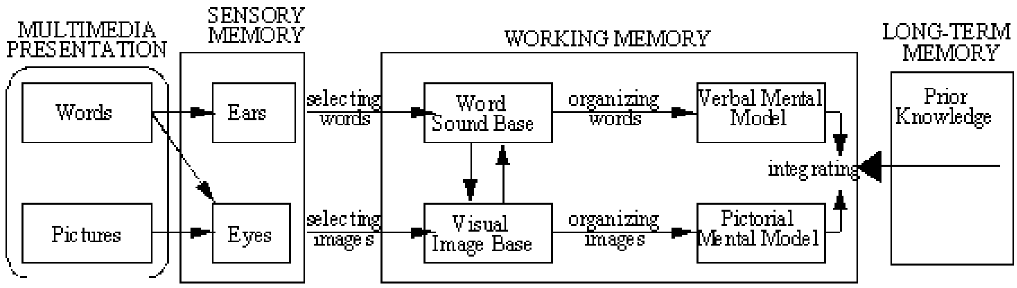


FIGURE 5.3: Working memory takes audio and visual input from sensory memory, processes it an integrates the results with knowledge from long-term memory.

The integration point with long-term memory is further in-line with the revised model of working memory, which includes an episodic buffer (Baddeley, 2000). Although research has challenged these elements of the working memory model, there is also significant empirical evidence for each of them. The basis of Mayer's model, in terms of reducing cognitive load, is that designers should make sure that learning materials are designed to work with and not against this process. If, for example, duplicate information is found during integration, then cognitive load is increased. In evidence of this, recent work by Kirsh (2009) showed that providing people with a visual cue while playing games of tic-tac-toe in their heads, can impede rather than support participants. Yet, when playing an unfamiliar and modified version of the game, with a larger 4-by-4 grid, a visual cue supported users in playing the game faster. Mayer's model would imply that the 3-by-3 grid, as a visual cue, was considered duplicate information by working memory, as it was already established in long-term memory. Consequently, the duplication of information impeded rather than supported users in playing the game.

5.1.2.2 Reducing Cognitive Load

Given this model of working memory, Mayer and Moreno (2003) provided the following 9 ways of reducing cognitive load during learning, in relation to nine empirically demonstrated effects. Readers should refer to Mayer and Moreno (2003) who provide many references for each noted effect.

1. Off-loading refers to the reduction of cognitive load, by distributing learning into the different modalities of working memory. This method is based on the demonstrated *Modality effect*, which has shown that audio narration can help reduce high volumes of text. mSpace, for example, has tried to support this method by providing audio preview cues so that users may take advantage of the auditory channel when making decisions about musical domains (schraefel et al., 2003).

2. ***Segmenting*** refers to the reduction of extraneous load by breaking up learning into stages. This method is base on the demonstrated *Segmentation effect*, where lessons were broken into learner-controlled segments rather than a continuous unit. Web search naturally happens in stages, as users submit keywords, choose results, make judgements, and repeat. Notably, the learner is expected to control the progression through stages.

3. ***Pretraining*** refers to the preparation for learning by learning key characteristics and terminology before starting. This method is based on the demonstrated *Pretraining effect* where students were taught terminology of mechanical components before learning to use them. In information seeking, providing facets or overviews help may help learners to understand the key terminology before browsing results.

4. ***Weeding*** refers to the removal of unnecessary extraneous material. This method is related to the *Coherence effect*, which showed that learning improved by streamlining the teaching material. In the second pilot study in Section 4.3.2, for example, participants noted that short concise descriptions were easier to use that full academic definitions when learning to use the Sii framework. The simplicity of Google's interface, as discussed above, for example, may be explained by this *Coherence effect*.

5. ***Signalling*** is based on the *Signalling effect* and suggests that cues for how to process or work through learning material are important. Rather that weeding out unnecessary material, this method focusses on highlighting important material.

6. ***Aligning*** refers to the process of carefully co-locating related information. This method is based upon the *Spatial Contiguity Effect*, which refers to occasions when a user has to mentally integrate information from multiple sources, such as text and a diagram. Chandler and Sweller approach this problem by making sure that the text necessary to understand a diagram is embedded within the diagram. Otherwise, the system places unnecessary extraneous load on users, as they have to remember textual information while interpreting the diagram, or visa versa. An example here, from mSpace, may be that previous choices-made are highlighted and left in place, rather than displayed as a separate list of choices in a separate location (Wilson et al., 2008). Consequently, users can see both their decision and alternatives in place. Conversely, it may be better to have all your choices in one breadcrumb-style place, rather than having to find them in multiple facets.

7. ***Eliminating Redundancy*** refers to the *Redundancy effect* where the same information is displayed in multiple places, so that the user is potentially required to a) read information they have already read and b) recognize what is new or has already been seen. Chandler and Sweller further their previous diagram and text example, by removing text that simply states what is clearly demonstrated by the diagram.

8. Synchronizing refers to timely display of related information, if not persistent. This method is based upon the *Temporal contiguity effect*, where learning was improved by presenting narration and diagrams at the same time, rather than in sequence. Where off-loading recommended that some visual information be made auditory, for example, this method suggests that mixed auditory and visual information be presented at the same time if related. In the evaluation of three faceted browsers by Capra et al. (2007), discussed in Section 4.2.2, participants felt disorientated by switching from the two faceted browsers to the original website when viewing individual results.

9. Individualizing refers to the training of users in spatial learning. More focused on supporting learners when *Synchronizing* is not possible, the aim is to reduce the cognitive load required to keep diagrams in working memory while narration or additional text describes the diagram.

These methods work in different ways to support different forms of overload. Method 1, for example, helps reduce cognitive load by sharing the processing of core information across modalities. Method 2 and method 3 reduce cognitive load when both channels are overloaded with important information by spacing out the pace of learning. Methods 4 and 5 reduce cognitive load by removing or reducing unnecessary material and focusing on core information required for learning. Notably these two methods may provide the majority of the counter measure in the Sii framework, which pushes to support users in any required search tactic as quickly as possible. Methods 6 and 7 reduce cognitive load caused by poorly organised presentation. Finally, methods 8 and 9 reduce cognitive load imposed by the temporality of learning. The next subsection describes how such methods might be integrated into the Sii framework, before applying it to example analyses presented so far in this thesis.

5.1.3 Integrating Cognitive Load Theory within the Sii Framework

The first stage for integrating the methods above into an inspection UEM, is to identify what aspects of a user interface contribute to increasing cognitive load in each effect. Of the 9 methods, only 6 can be reduced through improving the learning material, or in this case the Information Seeking Interface (ISI). Unfortunately, it is not possible to ascertain from an ISI what constitutes pretraining (#3) for searchers. It may be possible in future work to identify, however, that certain user profiles could benefit from pretraining support. It is also not possible to determine if an interface supports segmenting (#2), as one cannot tell what needs to be segmented and into what size without knowledge of a larger task. Further, it is not possible to determine the development of spatial skills for the individualisation method (#9).

There are 6 remaining load-reducing methods that can be potentially operationalised and integrated into the Sii framework. Before operationalising them, however, the assumptions relating to each method must be specified. Any future changes should be based on these assumptions, as they influence the measures taken. These assumptions are:

1. *Off-loading* is required when there are (a) multiple features (b) providing core information on (c) a single modality channel.

4. *Weeding* is required when features are (a) unnecessary information for (b) a search tactic.

5. *Signalling* is required when a feature provides (a) core information (b) without any signalling.

6. *Aligning* is required when (a) multiple features are (b) providing core information in screen locations that are (c) far apart.

7. *Eliminating Redundancy* is required when the same (a) information is (b) in different places.

8. *Synchronization* is required when (a) multiple features are (b) providing core information on (c) different pages.

From this list of assumptions, there are a number of things that need to be recorded during inspection:

- Which modality a feature works in
  - eyes or ears
- Whether a feature includes signalling
  - yes or no
- Whether a feature provides core or extra information (in relation to each tactic)
  - core or extra (regarding each tactic)
- Where features are in relation to other features
  - within, next to, near (approximately the same quadrant of the screen), far (approximately a different quadrant of the screen), or different page
- Whether features provide the same or different information to other features
  - same or different

So far, this CLT inspection process has not been added to the online service, while still in the early stages of development. Procedurally, however, evaluators would be expected to make the recordings for these five aspects after identifying the features of each interface (preparation task 2) and before starting the data entry process (Loops L1-L3). During this extra task, the evaluator would have to determine a) the modality of each feature, and then b) whether each feature contains some form of signalling. Then, in respect to each other feature, the evaluator would have to identify a) how co-located they are and b) whether they present the same or different information.

From this data entry, and the subsequent application of the data entry process, the necessary values can be identified, including the following additional inferred values. First, it is simple to calculate whether the modality of each feature is the same as or different from the modality of the other features. Second, from the data entry process, we can assume that if a tactic can be achieved with a feature, then the feature includes core information for that tactic. Consequently, with respect to each tactic, we can infer from values greater than 0 that that feature contains core information, rather than superfluous information.

Further to these entered and inferred values, the extension weights the effect of each measure according to the empirically demonstrated impact values provided by Mayer and Moreno (2003). The *Coherence effect*, for example, reduced using the weeding method, has an effect size of 0.9, where as the signalling effect size is only 0.74. Mayer and Moreno (2003) also notes the number of studies that have contributed to the calculation of each effect size, ranging from 1 to 8 studies.

5.1.4 Applying the Cognitive Load Extension to Examples

In relation to the three graphs (G1, G2, and G3) produced by Sii, this extension produces three graphs from similar perspectives, herein called G1clt, G2clt, and G3clt. Below, these graphs are presented and discussed for several of the examples above. One potential limitation, however, is that as the measures depend heavily on how features relate to other features (for aspects such as similarity of content and distance between features), the extension provides limited information about single features on their own. Consequently, little can be learned about the cognitive load of keyword search, an example of feature support described in Chapter 3, on its own.

5.1.4.1 Analysing Backward Highlighting

Like the keyword search example, Chapter 3 and Chapter 4 evaluated the backward highlighting implementations individually. Both techniques, however, provide coloured highlights over text. Consequently, alone, they both impose the same amount of cognitive load, where they are considered as core or supplementary depending on the tactics being performed. The difference between the two implementations, described in Section 3.2.2, is that bucket highlighting duplicates highlighted information in the facets, and puts them together above the facets. Comparatively, backward highlighting only highlights the data within the facets. Consequently, when analysed in relation to the facets, we see the differences shown in Graphs G1clt, G2clt, and G3clt below.

Results

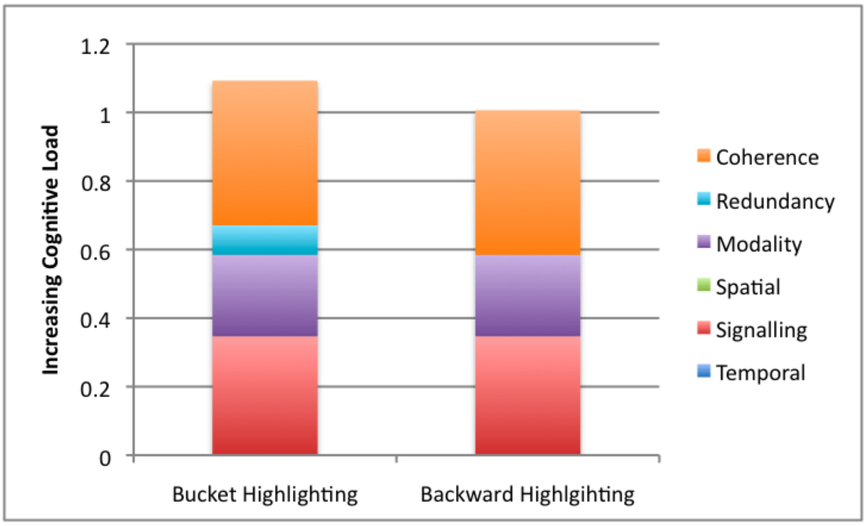
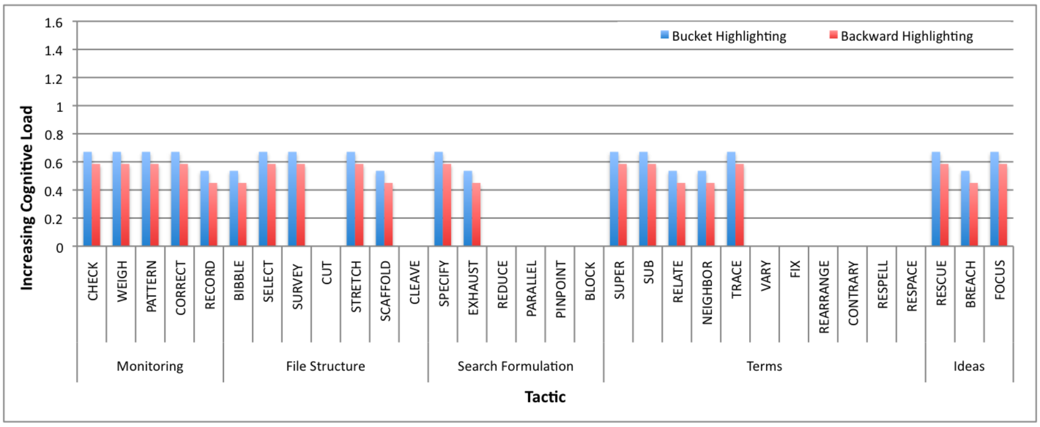


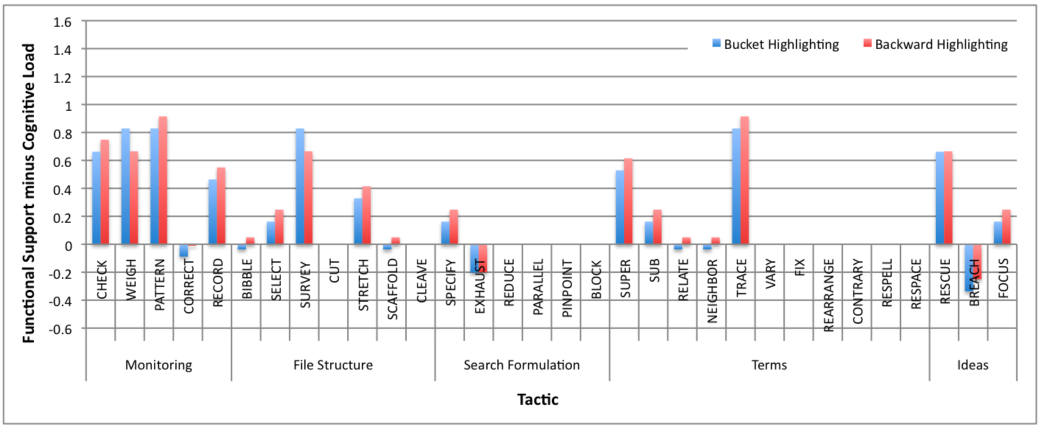
FIGURE 5.4: Graph G1clt, showing the cognitive load imposed by the two Backward Highlighting designs (Wilson et al., 2008) discussed in Section 3.2.2, where taller bars represent increased load.

From Graph G1clt, in Figure 5.4, as expected from the CLT theory used, Bucket Highlighting, creates additional redundancy load while not affecting the other forms of load. As they both contribute to the same tactics, they are both considered as providing core or supplementary information for each tactic. Consequently, they both contribute the same amount to the coherence effect. To reduce this coherence load, backward highlighting would have to provide core information for a wider range of tactics. Some of the design suggestions provided in Pilot Study 3 of Section 4.3.2 would increase the range of tactics the highlights support and thus reduce incoherence. Both features have the same modality effect, as they both work in the visual modality. Neither implementation contribute any spatial contiguity effect, as they are both within or directly next to the facets that they relate to. Both implementations perform as signalling for the facets, so do not impose any cognitive load through lack of signalling. Finally, both are temporally aligned with the information they highlight, so there is no temporal contiguity effect.

Figure 5.5 shows the Cognitive Load analysis of both backward highlighting techniques combined with the facets that they are highlight. Consequently, Figure 5.5 (a) shows the cognitive load imposed by each technique combined with the cognitive load of the facets. Figure 5.5 (b) shows the effect of taking the cognitive load away from the G2 functional analysis, of the highlighting techniques combined with the functional support

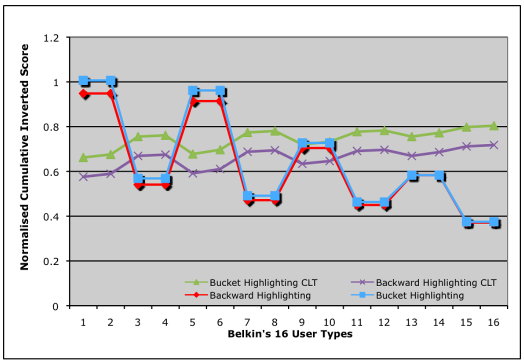


(a) Graph G2clt: Cognitive Load imposed



(b) Graph G2mix: Functional support minus Cognitive Load

FIGURE 5.5: Graphs G2clt and G2mix, showing the cognitive load imposed, and functional support provided by the two Backward Highlighting designs (Wilson et al., 2008), for 32 search *tactics* (Bates, 1979b,a), where taller bars represent increased load and stronger support, respectively.



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

(a) Graph G3clt: Functional support and Cognitive Load.

(b) User Profiles

FIGURE 5.6: Graph G3clt showing the functional support provided for, and cognitive load imposed by the two Backward Highlighting designs (Wilson et al., 2008), for 16 searcher profiles (Belkin et al., 1993, re-shown to the side), where peaks represent stronger support and higher load, respectively.

of the facets[5]. Although a rather crude technique, at this stage, simply penalising the functional support by taking away the amount of Cognitive Load does provide some seemingly accurate findings.

First, Graph G2clt (Figure 5.5 (a)) shows that, because the difference between the two implementations is in Bucket Highlighting's duplication of information, there is a consistent increase in Cognitive Load across all tactics supported by the techniques. It has already been noted above that both highlighting techniques cover the same range of tactics. Those tactics that are supported by both the highlights and the facets experience some modality effect, as their both features pass through the same processing channel. Those tactics that are only supported by one of the facets or highlights receive a slightly reduced cognitive load, caused instead by the coherence effect of having additional information.

Graph G2mix (Figure 5.5 (b)) shows where the two implementations vary. Notably, the increased cognitive load of the Bucket Highlighting technique means that there are only 2 tactics where the increased functional support outweigh the costs of cognitive load imposed on the searcher: WEIGH and SURVEY. SURVEY is the tactic that the functionality most directly aims to support, by placing all the highlights together, making it easier to view them all at once. Similarly, WEIGH is supported by the collection of highlights, as users can more easily to decide whether it is worth continuing with all the potential additional selections visible at once.

Graph G3clt (Figure 5.6) indicates that there are two main user groups that are particularly supported by the combination of mSpace facets and either of the highlighting techniques. These two groups (ISS 1-2 and 5-6), where the functional support outweights the cognitive load, are those who depend on recognising both metadata and information, particularly on the left-hand side where users are scanning for items because they do not know if one exists. A third group (ISS 9-10) also slightly experiences the benefits of highlights, which are those who recognising results, but knows that a particular result exists. This result is more prominent during the backward highlighting technique, where the cognitive load is lower.

Summary

The analysis of the backward highlighting technique, using the cognitive load extension, seems to provide results that correlate closely with the user study findings. The integration of the two measures in Graph G2mix, in particular, indicate that even a crude subtraction of cognitive load form functional support provides quite an accurate analysis compared to the findings of the user study described in Section 4.2.1. The next section analyses a more complicated example.

[5]This support is similar, in shape, to the more complete analysis of mSpace presented in Figure 3.11 on Page 70

5.1.4.2 Analysing the Academic Case Study

One Sii analysis that provides key motivation for including a cognitive load extension is the first case study of an academic research project, described in Section 4.3. The case study provided some empirical results indicating that users performed known-item searches with a basic keyword search implementation better than when using designs that included interactive query expansion features.

Results

Figure 5.7 shows the functional support provided by each feature, broken down by the types of cognitive load being imposed. Alone, the keyword search in the baseline interface only contributes some coherence load, as it does not support all the features. Notably, however, the coherence load of the keyword search is smallest in the baseline condition, as the term suggestions, and their definitions, only support a few tactics. No temporal, redundancy, or signalling loads were discovered. Although it might be considered that their suggested terms and their definitions might provide duplicate information, they are sufficiently close that, according to the assumptions listed above, they they appear as one source. Aside from the additional coherence load created by the terms and their definitions, spatial contiguity and the modality effect were both identified. The modality effect was recorded because of the additional text entering the visual mode. The slight spatial load is imposed as the searcher is required to switch between results and the side bar when reviewing term suggestions.

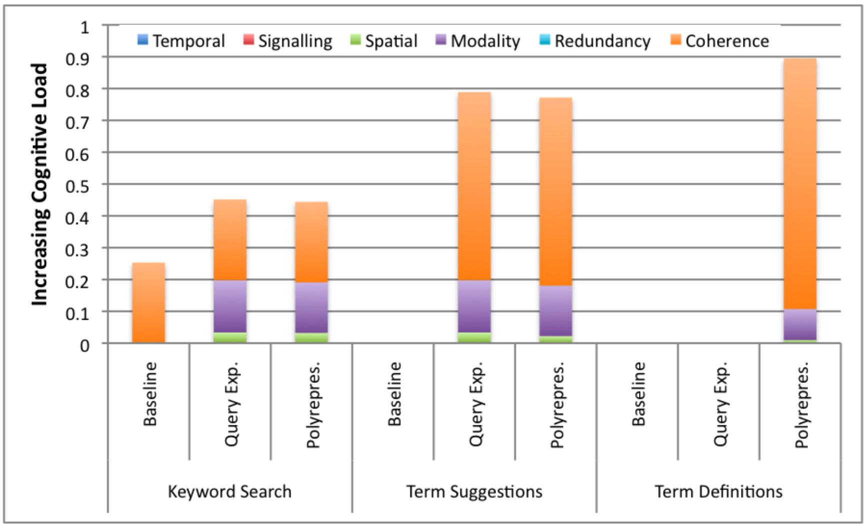
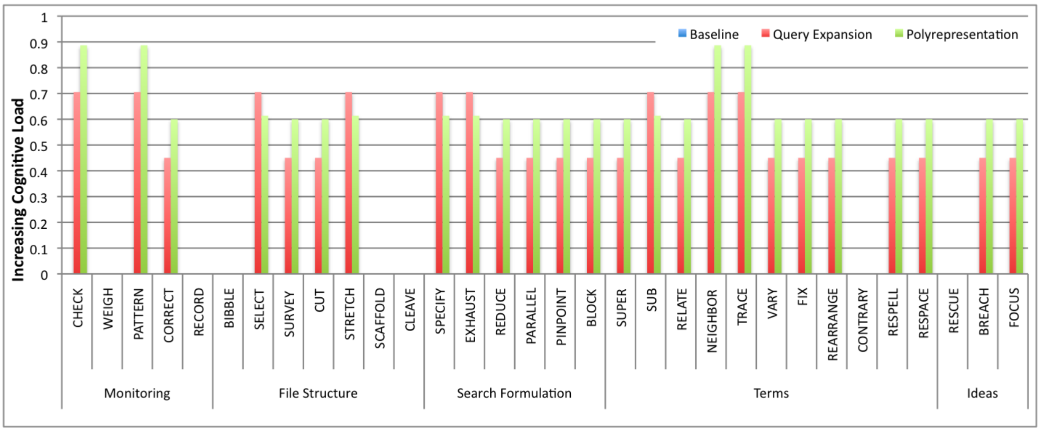
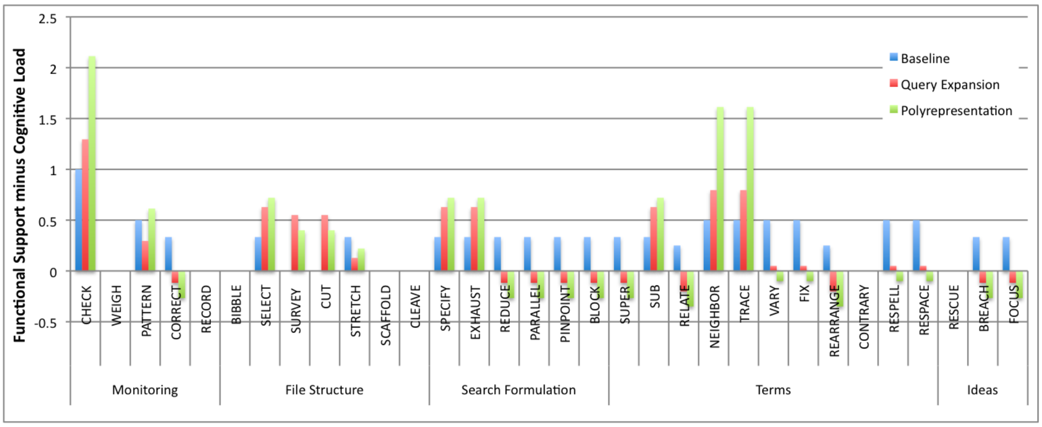


FIGURE 5.7: Graph G1clt, showing the cognitive load imposed by features of the Interactive Query Expansion designs studied by Diriye et al. (2009), where taller bars represent increased load.

As expected, the extra cognitive load imposed by the additional features shows, in Figure 5.8 (a), increased load on the tactics that the term suggestions are designed to support, such as PATTERN, and TRACE. It should be noted that G2clt (Figure 5.8)

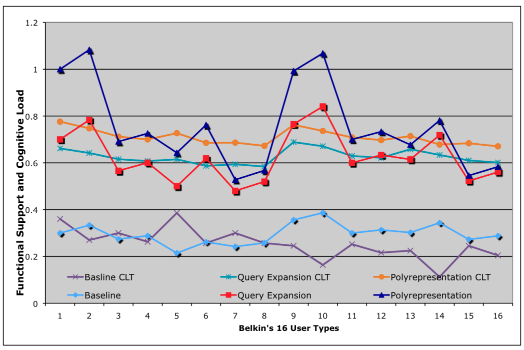


(a) Graph G2clt: Cognitive Load imposed



(b) Graph G2mix: Functional support minus Cognitive Load

FIGURE 5.8: Graphs G2clt and G2mix, showing the cognitive load imposed, and functional support provided by the Interactive Query Expansion designs studied by Diriye (2008), for 32 search *tactics* (Bates, 1979b,a), where taller bars represent increased load and stronger support, respectively.



(a) Graph G3clt: Functional support and Cognitive Load.

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

(b) User Profiles

FIGURE 5.9: Graph G3clt showing the functional support provided for, and cognitive load imposed by the Interactive Query Expansion designs studied by Diriye et al. (2009), for 16 searcher profiles (Belkin et al., 1993, re-shown to the side), where peaks represent stronger support and increase load respectively.

shows the cognitive loads for the tactics that are achievable with the functionality. The cognitive load noted for baseline condition in G1clt (Figure 5.7) is in the implementation's inability to support gaps shown in Figure 5.8. The remaining question is as to whether the additional functional support provided by the term suggestions is worth the extra cognitive load imposed. Figure 5.8 (b) shows the crude subtraction of cognitive load from the functional support, as discussed in the previous example. The graphs suggest that there are many tactics where the extra functionality outweights the cognitive load imposed, especially in those tactics that were noted as imposing a high cognitive load themselves. From the number of taller blue lines also shown, however, it is clear that there are also many tactics where the baseline keyword search design provides better overall support. Notably, as also shown in G3clt (Figure 5.9), these tactics are those important to searchers who know what they are looking for and how to describe it.

Summary

Diriye and colleagues recorded empirical evidence that the participants performed known-item searches more efficiently with the baseline interface, which had no term suggestions or definitions. The authors posited that the support for each user profile needed to be normalised by the support for all searcher profiles, because the different designs were trying to support different forms of search. The suggested normalisation has shown mixed results so far, but requires additional research in the future. Instead, this cognitive load extension indicates that the additional functionality imposes more cognitive load than support for known-item searcher profiles. Graph G3clt, shown in Figure 5.9, shows both the functional support for each interface design, and the cognitive load they impose, for each search profile. As intended, the recognising and learning profiles receive much more functional support from the additional term suggestions and definitions. Notably, however, and in line with the empirical findings presented by Diriye (2008), the baseline interface provides greater support, than cognitive load, in the right hand side of the graph, where searchers know what they are looking for.

5.1.4.3 Analysing 3 Faceted Browsers

In this section, the three complex faceted browsers, studied by Capra et al. (2007) and discussed in Section 4.2.2, are analysed. The hope is that a CLT analysis can explain why their study revealed that participants performed equally well, if not slightly better, with the carefully customised Bureau of Labor Statistics website compared to the academic and industry faceted browsers: RB++ and Endeca respectively. The Endeca interface, although highly customisable, was studied in an un-configured, off-the-shelf state. Their study used three tasks: a simple search task, a multi-faceted search task, and a learning task. Further, they asked participants to choose their most and least favourite features of each interface. The functional analysis in Section 4.2.2 shows that most of these results could be identified. Below, the cognitive load extension is applied in order to identify the remaining results.

Results

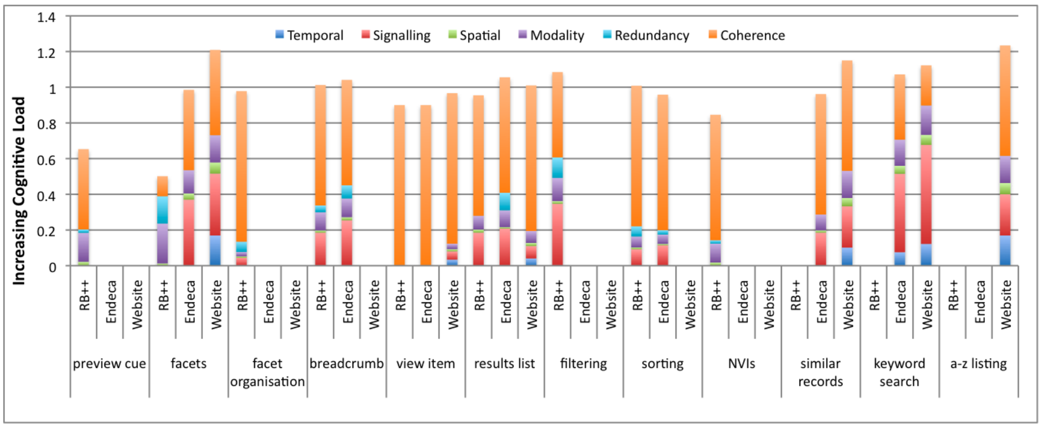
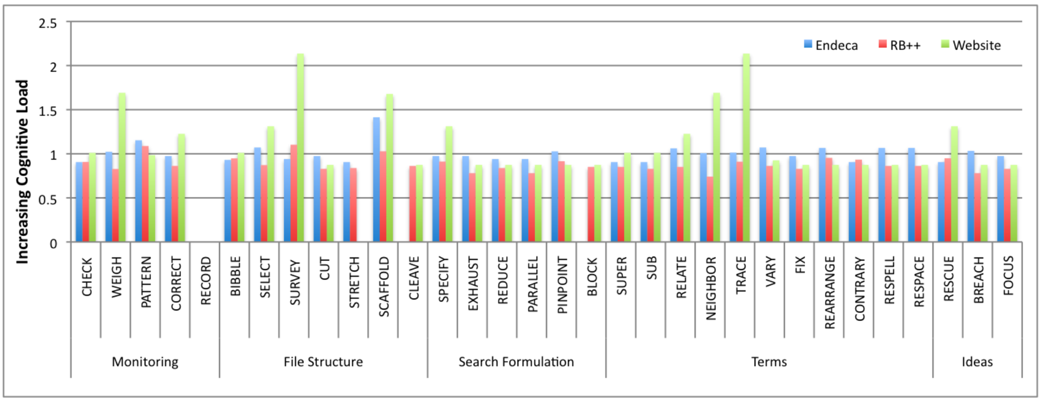


Figure 5.10: Graph G1clt showing the cognitive load imposed by features of the three interfaces studied by Capra et al. (2007), where taller bars represent increased load.

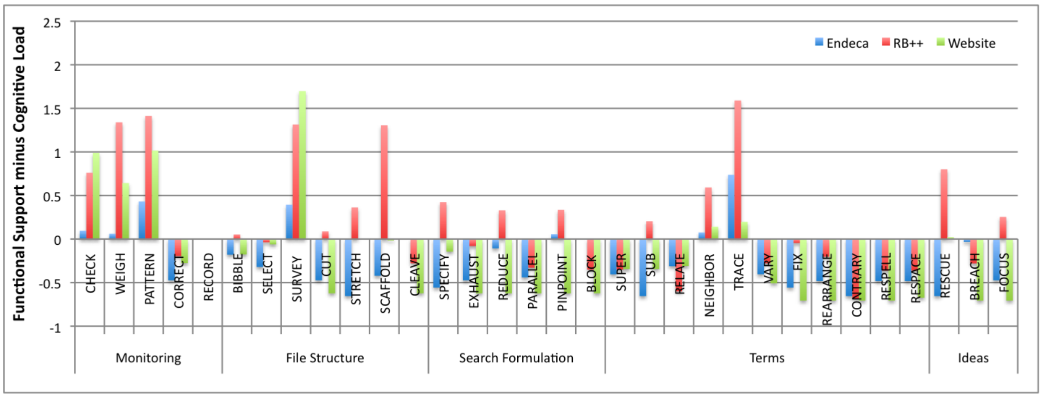
Two of the subjective findings from Capra et al. (2007) were not identified in the Sii analysis described in Section 4.2.2. First, participants reported that the way search results are shown was one of their least favourite aspects of the un-configured Endeca browser. According to Graph G1clt, shown in Figure 5.10, Endeca's results view imposes the highest cognitive load compared to the other two browsers. Further, Endeca's Keyword Search and Results view both impose the highest cognitive load of the features it provides. These two features were most commonly rated as the least favourite parts of the Endeca interface by participants.

The second result that was not explained by the Sii analysis in Section 4.2.2 was that, although the facets in RB++ were considered to be the most powerful, they were described as poorly organised. According to G1clt, the facets in RB++ impose the smallest cognitive load, but this load includes a much larger amount of redundancy and modality effect. One hypothesis is this split opinion of the RB++ facets is caused by a) the modality, b) the redundancy, or c) both the modality and redundancy loads imposed. Notably, of the three results views evaluated in Figure 5.10, the Endeca version has the largest modality and redundancy loads too. It may be that, in close competition, these two loads are subconsciously important to users.

One concern, which may need refinement in future work, is that the temporal disconnect experienced by some participants when jumping between the faceted browsers and the original website to view individual results, is not displayed by this cognitive load extension. It may be that grades of temporal contiguity are required, which differentiate jumping between pages within one website and switching between whole browsers. Notably, however, as users are able to do very little when viewing individual results,

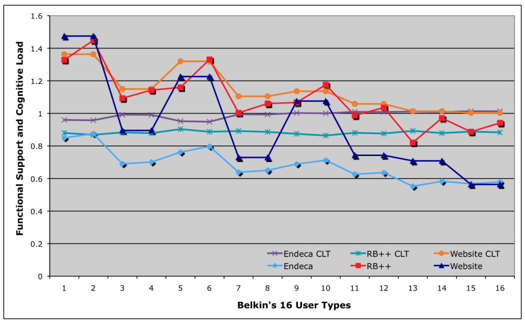


(a) Graph G2clt: Cognitive Load imposed



(b) Graph G2mix: Functional support minus Cognitive Load

FIGURE 5.11: Graphs G2clt and G2mix showing the cognitive load imposed, and support provided by the three interfaces studied by Capra et al. (2007), for 32 search *tactics* Bates (1979b,a), where taller bars represent increased load and stronger support respectively.



(a) Graph G3clt: Functional support and Cognitive Load

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

(b) User Profiles

FIGURE 5.12: Graph G3clt showing the functional support provided and cognitive load imposed by the three interfaces studied by Capra et al. (2007), for 16 searcher profiles (Belkin et al., 1993, re-shown to the side), where peaks represent stronger support and increased load respectively.

without returning to the browser, no loads are imposed for the other 5 cognitive load effects.

Graph G2clt, shown in Figure 5.11 (a), shows that no one browser imposes a higher or lower cognitive load consistently across all features. Endeca, however, typically imposes a higher load for the term tactics towards the right of the graph. Notably, the original website imposes much larger amounts of cognitive load for 5 tactics, which correlate roughly to the increased functional support shown in Figure 4.6 on Page 99. Graph G2mix, shown in Figure 5.11 (b), however, shows the results of subtracting the cognitive load from the functional support provided by the three browsers for each tactic. First, this subtracted analysis indicates that there are very few tactics where the functional support of the un-configured Endeca interface outweighed the estimated cognitive load. Another seemingly accurate analysis from G2mix is that the SURVEY tactic is most supported by the original website. This result was seen in Section 4.2.2 and still holds with this cognitive load analysis. Contrary to the empirical evidence provided by Capra et al. (2007), however, RB++ seems to provide more support for many of the tactics after subtracting any cognitive load. This more complex example may be highlighting the limitations of crudely subtracting cognitive load from functional support, as they are generated with different analytical measures.

Graph G3clt, shown in Figure 5.12 also provides mixed results. First, however, user profile 13, the profile representing two of the tasks in the user study, is the only profile where the cognitive load of the RB++ browser is higher than the support. In fact, for this user profile, the every browser provided less functional support than cognitive load. Further, user profiles 1 and 2, which represents the third task set in the study, is the only case where the Bureau of Labor Statistics website provided more support than its cognitive load. RB++ also provides more functional support in this and 2 other user profile groups (5 and 6, 9 and 10). It is not clear, from this analysis, however, why the Endeca browser did not perform significantly worse in the user study by Capra et al. (2007). This empirically insignificant different found by Capra and colleagues remains, therefore, a motivating case study for future research into both the Sii analysis and this cognitive load extension.

Results

The results of this example application of the cognitive load extension have, again, provided results that correlate with the findings of empirical studies. This example, however, has also highlighted some of the limitations of the extension. First, this more complex example seems to show that the currently crude method of subtracting the cognitive load measure from the functional support measure may be too simplistic. Graphs G2mix and G3clt indicate that the cognitive load of the three browsers almost consistently exceeds the functional support they provide, which seems unlikely. Second, the temporal disconnect experienced by participants when jumping between the two browsers and the baseline website is not captured by the model at all.

5.1.5 Conclusions and Future Work

This section has presented the early stages of an investigation into a complementary measure for use within the Sii framework. The Sii framework rewards ISIs that support a range of tactics and user profiles in as few actions as possible. Command line interfaces, however, support users in performing many tasks, but are hard to use unless users have learned all the commands. This complementary extension, therefore, rewards ISIs for minimising the amount of cognitive load being imposed upon the searchers. So far the results have been promising, providing some analyses that correlate well with experiences reported from empirical user studies. Some study findings, particularly from the 3rd example, which would seem to be related to cognitive load theory, however, were not identified by Sii's new extension.

Like the original framework, this extension should be carefully studied for an extended period of time. Further, given that the human brain is an inherently harder object to study than the behaviour outwardly exhibited by searchers, the underlying models of cognitive load theory are less well defined. Notably, unlike the numerous empirical studies of search behaviour, there are relatively few studies that have taken direct cognitive measures of human searchers. Consequently, it is harder to establish whether the results of this cognitive load extension are accurate. Recent research indicates, however, that we may soon be able to quantifiably measure the cognitive load being imposed by user interfaces. Hirshfield et al. (2009), for example, demonstrated that functional near-infrared imaging (fNIR) could identify the different loads experienced by participants performing memory-based tasks using two interface conditions that provide notably different support for the task.

There are many variables in this extension that could be studied during future research. The effect sizes identified by Mayer and Moreno (2003) have been used as weights over the contributions of each effect. The use of these weightings, and their actual values, should be studied to see if they have a significant effect on the accuracy of results. Further, given the lack of user studies performed from the perspective of cognitive load, empirical evaluations need to be performed in order to further determine if the extension is accurate. In the applications of the extension above, a potential mixed measure was initially proposed. Currently, the cognitive load measure is crudely taken away from the support provided by the ISIs. Both the method and value of integrating the two measures, however, have shown strong potential, and should also be studied in future research.

In summary, the successful creation of such a complementary measure to the Sii framework, combined with some seemingly accurate results, shows that the investigation into cognitive load theory is promising. Further work is required, however, in order to empirically demonstrate that the extension is accurate. Like the original framework, case studies investigating the value of the extension for evaluators, will also be required. Notably, given the challenge of studying cognitive aspect of human learning, this extension alone should receive the same amount of study, if not more, than the Sii framework.

5.2 Collaborative Search

The previous extension above was designed to produce a counter measure to integrate into the Sii framework to trade-off decisions regarding functionality and complexity. The aim of this extension to the framework, however, is to enable the inspection of a special case of ISIs that are not accommodated by the theory used to develop the framework. Consequently, this extension does not affect or increase the number of analyses produced, but re-frames Sii so that it can be applied in an unusual circumstance (Wilson and schraefel, 2008a, 2009a)[6].

Although the majority of Information Retrieval (IR) and Information Seeking (IS) systems have been designed for solitary use, recent research has shown that we collaborate on search activities with our colleagues, family, and friends, by asking for guidance, sharing links, and even dividing up tasks (Järvelin and Ingwersen, 2004; Morris, 2008; Twidale et al., 1997). Consequently, several novel search interfaces have been developed recently to support users in collaborating on shared search tasks (Amershi and Morris, 2008; Morris and Horvitz, 2007b,a; Smeaton et al., 2006). Along with the challenge of designing new collaborative search interfaces, however, comes the challenge of evaluating them. Like all evaluations, methods for assessing CIS interfaces will be grounded by a) how we model the nature of collaborative search, and b) how we model successful or efficient CIS behaviour. Consequently, this part of the chapter first reconsiders how much of the underlying existing and often well established, solo-focused, IR and IS theory still applies to Collaborative Information Seeking (CIS) contexts.

The Sii framework, however, is built upon the solo-focused IS theory, and so part of the aim for this extension is to a) show that the extension can be applied to CIS interfaces; b) show how it can identify additional requirements for CIS interfaces; c) show how it can model different dynamics within collaborative teams, such as experts searching with novices; and d) provide the additional means required for CIS researchers to apply the framework to collaborative search software. Further, performing this re-assessment of IS theory begins, with two specific examples, the inevitable process of reconsidering the models and default assumptions held by the IS community, in the light of CIS activities.

[6]The most up-to-date description of this work is described by Wilson and schraefel (2009a)

In the following sections, the history of IS research is first reviewed, especially highlighting where CIS has been addressed, in order to summarise what is already known about collaborative searching and to inform the re-assessment of the Sii framework. Once complete, the models of search tactics (Bates, 1979b,a) are re-framed from the perspective of CIS contexts. Then, in Section 5.2.3, an evaluation is presented of a recent freely available CIS interface: SearchTogether (Morris and Horvitz, 2007b), using the CIS extension to the framework. The results of the evaluation are then further validated by correlating its findings with known usability issues identified by the designers of SearchTogether.

5.2.1 Reasons for Analysing Collaborative Search

Although forms of implicit collaboration, such as recommender systems (Resnick and Varian, 1997) and even Googles PageRank (Brin and Page, 1998), have been well researched, investigation into interfaces for explicit, synchronous and asynchronous, collaborative information seeking has only recently received a flurry of interest. This is surprising given that such collaboration during search has been identified many times in the history of Information Seeking (IS) research, discussed in more detail by Hansen and Järvelin (2005), and that there has been around 20 years of research into Computer Supported Collaborative Work (CSCW). The recent focus on CIS research, however, is a union of these two areas that extends our ideas of IS research with a subset of the tasks being investigated by CSCW.

5.2.1.1 Collaboration in Information Seeking Focussed Literature

Much of the early work into IS was researched within Information and Library sciences before personal computers, and certainly the World Wide Web, were widely available. The dominance of primarily solitary keyword searching interfaces on personal computers and the web has, as discussed further by schraefel (2009), overshadowed our understanding of alternative models and searching scenarios. Consequently, some of our understanding of CIS can be learned from a time when IS was usually performed in physical encironments and in conjunction with librarians. Several models, for example, were based on dialogues or conversations between typical searcher and librarian roles: Conversation for Action (Winograd and Flores, 1986), for example, and the Conversational Roles Model (COR) (Sitter and Stein, 1992; Stein and Thiel, 1993). The result of these searcher/librarian models, however, has usually been to design search interfaces to act as the librarian. Belkin et al. (1995), for example, created 16 typical search scripts (including transition points between scripts) to influence the design of a dynamic dialogue-based search interface. The focus of CIS, however, is on dialogue between two searchers during search, rather than the dialogue between a searcher and a librarian, or a user and a system.

Focusing more directly on dialogues between two users, several IS models highlight the socio-organizational contexts in which searching takes place (Järvelin and Ingwersen, 2004; Kuhlthau, 1991; Marchionini, 1995; Wilson, 1981). Typically, though, these models have focused on the affect that social contexts have on individual search behaviour. Hansen and Järvelin (2005), however, studied the socio-organizational contexts of a Swedish patent office, empirically demonstrating that active collaboration can occur throughout the typical search-process stages: problem identification, planning, seeking, and completion. Allen (1977) studied the socio-organizational settings of engineers and scientists, showing that in many cases colleagues were also used as sources of information and/or guidance. O'Day and Jeffries (1993) showed that the results of seeking activities are usually shared or distributed within an organisation. Further work by Talja (2002), categorised such sharing as one of: strategic, paradigmatic, directive, or social distributions. More detailed surveys of collaboration in the information seeking domain have been provided by Hansen and Järvelin (2005) and Prekop (2002).

5.2.1.2 Further Defining Collaborative Information Seeking

Given the relatively small amount of direct CIS investigation so far, some initial efforts have focused on identifying the specific requirements for collaborative search software. An example is the survey performed by Morris (2008), which revealed that around 95% of people take part in collaborative searches, with the majority performing these either a) once a week or b) once a month. The most common tasks for collaborative search included: travel planning, online shopping, and literature searching. 80% of these searches were typically performed in a pair. 22% indicated that they were co-located, 12% occurred in separate locations, and the remaining majority reported that they had taken part in both co-located and remote collaborative searches. Of these collaborative searchers, only 18% indicated that they had divided a task among the participants, with up to 87% searching together.

Another initial strand of CIS research has been to better define what counts as collaboration during search. Shah (2008) presented an onion model of CIS indicating that collaboration is made up of several encompassing layers of interaction, including communication and corroboration. One take away from this onion model is the suggestion that collaboration goes beyond users simply working in group, to searchers working together in the support of mutual interest and gain.

Golovchinsky and colleagues (Golovchinsky et al., 2009; Pickens and Golovchinsky, 2007) have formalised an understanding of CIS research, by identifying the facets that define CIS: a) explicit versus implicit collaboration, b) depth of mediation (server to interface), c) concurrency, and d) location. Explicit CIS is in-line with the activities surveyed by Morris, in which groups of searchers actively work together to achieve a shared task. Implicit CIS, however, represents the times when a users search is affected by other similar searchers. Collaborative filtering (Resnick and Varian, 1997), and to some extent Googles PageRank (Brin and Page, 1998), use the experience of the masses to support or improve new searches. CIS research is typically concerned with explicit collaboration, where implicit systems, such as collaborative filtering and ranking algorithms, have been studied in great detail already. Depth of mediation is defined by whether a search system controls the collaboration (Golovchinsky et al., 2008; Pickens et al., 2008), or whether the user interface allows users to communicate and work together (Amershi and Morris, 2008; Morris, 2008). Concurrency determines whether users are searching synchronously at the same time (Amershi and Morris, 2008), or asynchronously at different times (Morris and Horvitz, 2007a). Finally, searchers can either search together in one environment (Amershi and Morris, 2008; Morris et al., 2008) or in distributed environments (Morris and Horvitz, 2007b).

5.2.1.3 Designing Collaborative Information Seeking Interfaces

Recent efforts have produced some early designs of explicit collaborative search software that, in turn, are also producing new insights into additional requirements for collaboration during information seeking tasks. S3, standing for Storable, Shareable Search (Morris and Horvitz, 2007a), was designed to support explicit asynchronous search, mediated by the user interface, and for either co-located or distributed groups, by recording peoples searching activities, making them persistent over time, and providing them to others in a team. CoSearch (Amershi and Morris, 2008) is designed to support explicit, co-located, synchronous CIS, by allowing groups of searchers to use mobile devices to interact with queries being performed on one machine. These external devices could be used to suggest queries into a queue, and to share the load of parsing pages of results. Another approach to co-located CIS has been to design larger devices that support multiple simultaneous users. The Fischlar-DiamondTouch system (Smeaton et al., 2006), for example, provides separate and shared spaces on a single table-top display, to allow users to share results when searching for videos. SearchTogether (Morris and Horvitz, 2007b) is designed to support explicit, distributed CIS, which provides means of communicating with, recommending pages to, and monitoring the activity of other searchers. SearchTogether users can be synchronous or asynchronous, as search summaries are kept to support users in joining or re-joining a search.

The CIS interfaces discussed so far have all been mediated at the interface level. Research by Pickens et al (Golovchinsky et al., 2008; Pickens et al., 2008) mediates search at the system level, by distributing results automatically using the findings of one group member to suggest searches to another.

5.2.2 How Collaborative Search can be Analysed with the Sii Framework

The key points that can be drawn from the existing work described above are that a) there has been a recent flurry of search systems that directly support CIS, b) the communitys understanding of CIS is becoming more formal but is on-going, and c) the notion of CIS has been identified numerous times implicitly in the history of IS research. Although this latter point is encouraging, the majority of IS research assumes the user is acting solitarily, especially within the Information Retrieval community that has focused on improving performance time and accuracy of document retrieval systems. It remains in question, however, as to how much of IS research still applies. Potentially all IS work may apply to CIS, but perhaps requiring extension.

Below, the two established models from IS theory, used within the Sii framework, are assessed to see how they can be applied to CIS. These two models are presented and discussed in detail, for three reasons: 1) in order to understand their re-framing, we must first understand the original models; 2) this section can then be used independently as a reference for how these models apply to CIS behaviour; and 3) for CIS designers to apply the modified framework appropriately, the full extent of their re-framing must be detailed. Section 5.2.2.1 re-frames Bates model of search tactics, by providing a description of each of the 32 tactics and how they apply in a collaborative context. Section 5.2.2.2 re-frames the model of user profiles, provided by Belkin et al, from the perspective of different roles taken within groups of collaborative searchers.

5.2.2.1 Re-Framing Bates Model of Tactics for Collaborative Search

Bates, discussed in more detail in Chapter 2 identified 32 different tactics that people may carry out when searching for information across different technologies. Where these were originally designed to be self-serving tactics, they may have different implications for those who are part of a group or team. Each of these tactics is now discussed to identify the additional considerations that evaluators must maintain when applying the framework to collaborative search software.

The first five tactics are Monitoring Tactics.

- CHECK is to check that the current state of search is still related to the original reason for searching. In a group setting, the user may have to check both their current task, and the overall task of the group.
- WEIGH is to consider whether to continue or choose a different approach. In a group setting, users will require knowledge of what approaches have already been tried by other members of the team.

- PATTERN is to monitor ones actions for efficiency. In a group setting, users may benefit from comparing their own patterns to those of co-searchers.
- CORRECT involves watching for and correcting any errors during search. Although this may maintain as an individual activity, the many eyes of others may help identify errors a user has missed. Thus, in a group setting, it may be helpful to notice errors in other peoples work.
- RECORD is to record items for later return. The capture of context here may be even more important for others in the group who did not perform the original search.

The following 7 tactics relate to parsing result sets.

- BIBBLE is to check to see if other searchers have already carried out the current task. This may change vary little, except that those who may have already carried out the work may be others in the team, rather than unknown searchers from the past.
- SELECT is to select part of a task and address it as a set of sub-tasks. In a group setting, it may be beneficial to know that others have not already completed these sub-tasks, or to see if others could share the workload.
- SURVEY is to review the current available options. Again, it may be of value to know that others have not already completed some of current options.
- CUT is to take an action that has the largest affect on the overall task. This may not vary in collaborative search software, as other tactics from this group deal with preparing for the decision.
- STRETCH is similar to reusing something. It may be that a user can stretch the value of someone elses hard work to benefit their own. The actions of a known team of group may be much easier to visualize than trying to browse the previous actions of every other user in the history of the search service.
- SCAFFOLD is to design a different approach to find a certain result, having followed a dead end path. This may be much easier to do if the user can see and mimic the successful paths taken to similar targets by others.
- CLEAVE is a fairly solo activity in terms of applying a binary search technique to going through a structured list.

The following 6 tactics relate to formulating search plans, which has been shown as a core activity during collaborative search (Morris, 2008).

- SPECIFY is to apply a set of query terms that are known to produce the desired result. Searchers may benefit from knowledge from others in the group to do this, especially those who are not search-savvy.
- Being EXHAUSTive is also an activity that is easier with a team of searchers.
- To REDUCE is the opposite of EXHAUST, which allows un-expected but potentially valuable results to be found. This often involves parsing a larger amount of results, with many being unrelated or previously found and so shared human resources may help here too.
- PARALLEL is to broaden a search by using synonymous terms, for example. Like EXHAUST, this may be easier with shared group knowledge.
- To PINPOINT is the opposite of PARALLEL, and allows for searching to focus on specific synonyms.
- BLOCK relates, for example, to the use of NOT in a Boolean query. In a group, this action may help avoid overlap and may help searchers to discover results on a certain topic, but avoid results that relate to what a colleague is searching for.

The next 11 tactics relate to the specific terms used after having formulated a search plan: SUPER, SUB, RELATE, NEIGHBOR, TRACE, VARY, FIX, REARRANGE, CONTRARY, RESPELL and RESPACE.These are not discussed individually, but they are each mainly solo decisions. They could still benefit, however, from an awareness or reuse of other peoples search terms and phrases.

The final 3 tactics relate to changing ideas or mental concepts of the searcher and so tend to relate to the on-going learning that informs better searching behavior. Consequently, the three tactics are important for a team setting for keeping each other informed and sharing specific advances on a goal or problem.

- RESCUE is to rethink a problem, when the searcher realizes their ideas are inherently incorrect.
- BREACH is to extend ones boundaries of understanding given new information. An example may be realizing that diabetes is not solely related to genetics, but also to aspects such as diet.
- FOCUS, therefore, is the opposite of BREACH and relates to identifying that only a sub-part of a problem is actually relevant to the overall goal.

Many of the tactics above relate to the communication and sharing of findings from one searcher to the group. This concern appears to be important, as any member of the team may significantly alter the direction of the whole group.

5.2.2.2 Re-Framing Belkin et als Model to Searcher Roles Collaborative Groups

Searchers engaged in collaborative shared tasks often naturally fall into different roles (Golovchinsky et al., 2009; Morris and Teevan, 2008), where the resulting group dynamic is influenced by their individual skills. Belkin et al. (1993) identified 16 different types of searchers, shown in Figure 2.1, based on unique combinations produced by four binary dimensions: Method, Goal, Mode, and Resource. This model is described further in Chapter 2, but the figure is shown again here (Figure 5.13) for easy reference during this re-framing. The main assumption made in this collaborative re-framing of Belkin's user model, is that different group dynamics can be represented by combinations of these profiles[7].

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

FIGURE 5.13: The 16 searcher profiles identified by Belkin et al. (1993), used with the Sii framework.

In a group setting, a searcher's role may be dependant on their existing knowledge and search experience (Golovchinsky et al., 2009), and the assumption held here is that these different roles map to particular searcher types, or to small sub-sets of searcher types. Golovchinsky and colleagues define several example roles: Search Expert, Domain Expert, Search Novice, and Domain Novice. Additionally, they describe a well-known pair of roles called Prospector and Miner, where a Prospector searches broadly for possible search paths, and a Miner investigates them in more detail. Teams of searchers may, however, have balanced peers searching together, such as two Domain Novices working together in the same way. Prekop (2002) also discusses some additional CIS roles, however these include more general information behaviour roles such as co-ordinating and managing team direction, which themselves do not directly involve information seeking.

[7] An improved version of this role analysis has been produced by Wilson and schraefel (2009a)

The easiest relationships to define using Belkins model are Prospector and Miner, Prospectors are scanning for possible leads (they do not know if a lead exists) in order to learn about the domain, and they will do this by recognizing good leads when they see them. The Prospector will also largely depend upon metadata of the searches. This means that a Prospector is primarily ISS2 (in Figure 5.13), but potentially ISS1, if they consider some example resources as well as metadata. They may also occasionally stretch to ISS3 and ISS4 if participants accumulate some good terminology during their search. Miners, however, are given leads to investigate further. A Miner, therefore, is primarily specifying to select results and primarily dealing with resources, rather than metadata. Miners are still scanning, rather than searching for known items. This puts Miners at primarily ISS7 and maybe ISS8. There is also a cross-over with Prospectors, however, when Miners are making sense of the leads provided to them, and so are temporarily learning from metadata (ISS4), and sometimes directly from resources (ISS3). Primarily, however, we are suggesting that Prospectors are ISS2 and Miners are ISS7.

The descriptions of Domain Experts, Domain Novices, Search Experts, and Search Novices, do not make any assumptions about the resource being sought. Also, we cannot make any assumptions about whether they know if a resource or metadata exists, although we can assume that either Domain Experts or Search Experts may know more about what they are looking for. We can assume, however, that a Domain Novice will primarily need to learn, whereas a domain expert is unlikely to need to learn. We can also assume that Search Experts know how to specify what they need in a search system, where as Search Novices will depend more heavily on recognizing results from less specific queries. Consequently, Domain Experts may include ISSs 5-8 or ISSs 13-16. Domain Novices, however, will be mainly limited to ISSs 1-4 only, but potentially 9-12 if they know of a particular resource to learn from. Search Experts may include ISSs 3-4, 7-8, 11-12, and 15-16. Search Novices, however, will be mainly limited to ISSs 1-2 and 5-6, but may potentially include 9-10 and 13-14 if they know of a resource to learn from. Clearly there may also be intersections as an experienced searcher may also be an expert in some domains, which would make them primarily ISSs 7-8 or 15-16.

The discussion of roles above re-frames Belkins ISSs for CIS research. This, and the re-framing of Bates tactics, are used to enhance the original models discuss the results of the example application of the Sii framework to a CIS interface in Section 5.2.3 below.

5.2.3 Analysing an Example Collaborative Search Interface

This section describes an analysis, using the Sii framework with the re-framed models described above, of a freely available CIS interface called SearchTogether (Morris and Horvitz, 2007b). SearchTogether makes a good clear example to inspect, as it designed for explicit, distributed, collaborative information seeking.

5.2.3.1 Method

The procedure for applying the framework, shown in Figure 3.2 on Page 58, remains unchanged. In this case, there is only one interface, shown in Figure 5.14, being evaluated: SearchTogether. Consequently, the value of Loop L1 is 1. The value of L2 is 16, and these features are listed in Table 5.1 and highlighted in Figure 5.14. As per usual, the value of Loop L3 is 32, but using the re-framed definitions described above. The process was applied using the online Sii framework, where the full analysis can be viewed using interactive graphs[8].

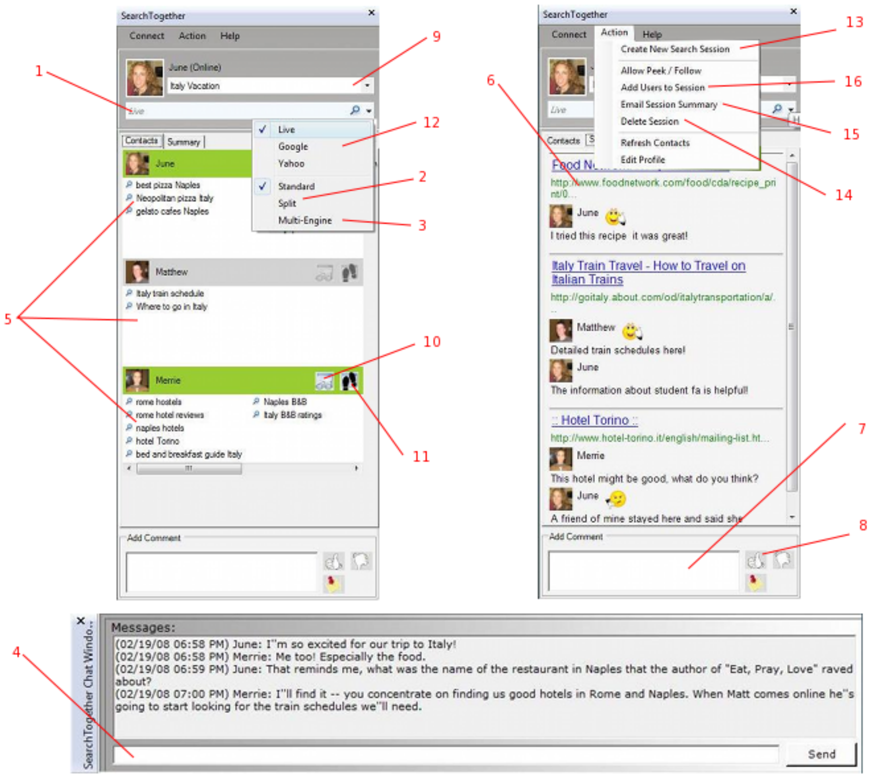


FIGURE 5.14: Design of the collaborative information seeking interface: Search Together, where callout identifiers are described in Table 5.1

5.2.3.2 Results

Graph G2, shown in Figure 5.16, shows how each of the 32 known search tactics are supported by SearchTogether. Notably, the 5 tallest bars are for the CHECK, PATTERN, STRETCH, SCAFFOLD, and RESCUE tactics, which together relate to some of the key expected benefits of CIS, in the form of monitoring other searchers. PATTERN, for example, represents the tactic of looking for search patterns that find good results. One of the foremost elements of the SearchTogether interface is the visualisation of the queries used by collaborating searchers. The interaction with this query history is very

[8] http://mspace.fm/sii/project.php?pid=000010

#	Interface Element	Description
1	Keyword Search	Standard keyword search functionality, provided by the search engines.
2	Split Search	The ability to split searches across people taking part in the search session.
3	Multi-Engine Search	Splitting the search, but by search engine.
4	Chat Function	A messaging function for people in the search session.
5	Query Histories	Per-user query histories, which can be double-clicked to see the results again.
6	Summary Panel	A panel that displays the search summary of recommended and commented URLs.
7	Comment on URLs	The ability to add a comment to a page being viewed - displayed in the search summary.
8	Thumbs Up/Down	The ability to rate a page positively or negatively - displayed in the search summary.
9	Change Search Session	The ability to switch between search sessions.
10	Peaking	The ability to see what someone else is looking at.
11	Following	The ability to peak, but have the pages update as the person you are peaking at follows links.
12	Change Default Search Engine	The ability to change to a different search engine.
13	Create New Session	The ability to create a new search session.
14	Delete Search Session	Remove a search session.
15	Email Session Summary	Email a search session summary to someone.
16	Add friend to Session	The ability to add someone new to the search session.

Table 5.1: The 16 interface elements identified in the evaluation of SearchTogether, where the callout identifiers match those included in Figure 5.14.

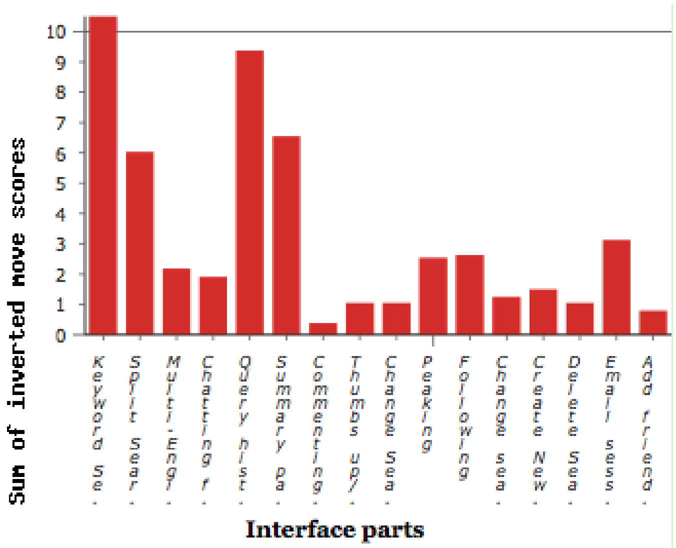


Figure 5.15: Graph G1 showing the support provided by the features of SearchTogether (Morris and Horvitz, 2007b), where taller bars represent stronger support.

simple. It can support numerous tactics, including CHECK, PATTERN, SCAFFOLD, and BREACH, without requiring any physical moves. Further, it can be used to drive new queries directly, by clicking on the listed terms, with only one mental (choosing) and one physical move (clicking). The prominence of the query histories in the interface and their simple interaction model mean that they provide almost as much total support for search as the basic keyword search function (see Figure 5.15).

The next most well supported tactic is BIBBLE. Of the 32 tactics, BIBBLE inherently depends on other searchers, and is defined by identifying whether anyone else has already searched for a term. A question would first be as to why this is not the most supported

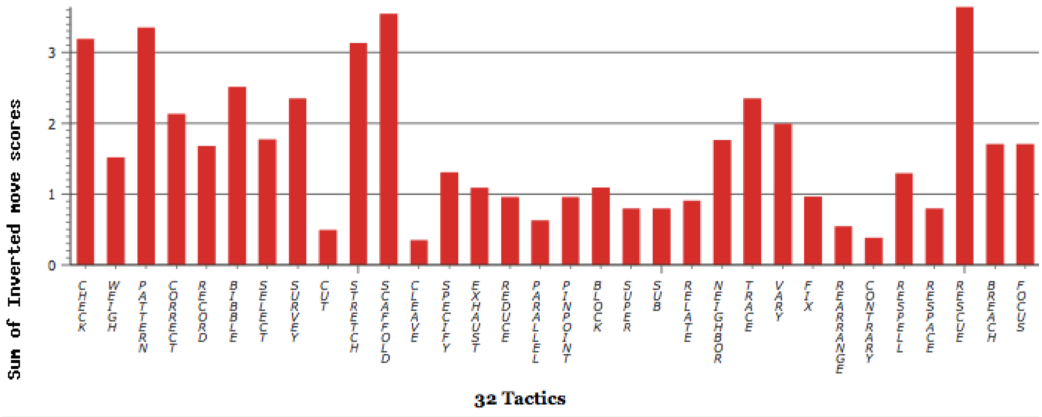
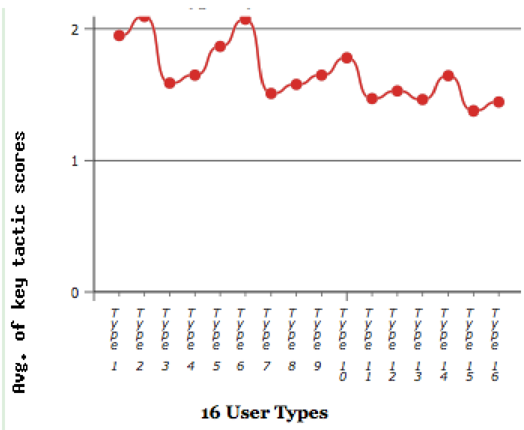
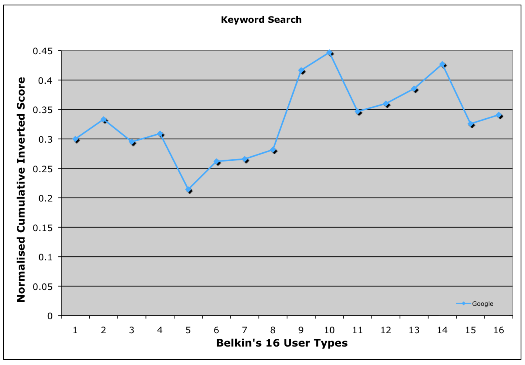


FIGURE 5.16: Graph G2 showing the support provided for 32 search *tactics* (Bates, 1979b,a), by the features of SearchTogether (Morris and Horvitz, 2007b), where taller bars represent stronger support.



(a) Search Together



(b) Keyword Search from Section 4.1.3

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

(c) User Profiles

FIGURE 5.17: Graph G3 showing the support provided for 16 searcher profiles (Belkin et al., 1993, re-shown below) by SearchTogether (Morris and Horvitz, 2007b)., where peaks represent stronger support. The analysis of Google's keyword search analysis (Section 4.1.3 is also re-shown to the side for comparison.

tactic in SearchTogether. The answer is that BIBBLE is supported by fewer elements of the interface, mainly the query history. If the summary panel, for example, displayed the queries used to find any recommended results, this would also support BIBBLE. Instead, these other views are better for sparking new ideas (RESCUE, SCAFFOLD, STRETCH), and so together receive a wider range of support throughout the interface.

Many of the remaining medially supported tactics, such as WEIGH, CORRECT, TRACE, NEIGHBOR, BREACH and FOCUS, are also supported by simply seeing other searchers actions, but in-directly. There is no specific functionality in SearchTogether, to help narrow the FOCUS of the search, except within the usual basic keyword interaction. However, it is easy for searchers to FOCUS their search by using terms from the query histories, for example.

Unusually[9], the RECORD function is fairly well supported. Users can easily add a result to the search summary (essentially keeping it on RECORD), by simply clicking on the Thumbs Up button (one physical move). With a larger number of moves, however, the user can also add a comment. These two interface elements, as shown in Figure 5.15, provide a relatively small contribution to the overall interface, but their simple interaction, especially the Thumbs Up/Down make the RECORD tactic one of the better-supported tactics in the system.

Improving Features

Several tactics are poorly supported by SearchTogether. It is hard for users to dramatically CUT down their results, or to explore them in any alternative ways than in the order they are delivered (CLEAVE). SearchTogether is ultimately built on top of the major search engine interfaces, which typically also struggle to support these tactics. While there are many new functions in SearchTogether, few of them support these tactics well, or the CONTRARY tactic. SearchTogether does attempt to support them, however, with the Split Search and the Multi-Engine Search. The interaction for these, especially for the mutli-engine search, takes much longer, as the user is required to explicitly allocate search engines to searchers, for example (see Graph G1 in Figure 5.16). These two searches would provide dramatically more support, potentially more than the basic keyword search, if they had separate buttons placed next to the search box. Multi-Engine search would also provide much greater support if it also automatically allocated each searcher to a different search engine. We can see an example of the minimal affect they currently have by comparing the EXHAUST and REDUCE tactics in Graph G1 (Figure 5.16), where the multi-engine search and split search should make it much easier for users to EXHAUST the potential results.

Looking further at Graph G2 (Figure 5.15), we see that the summary page provides the third highest amount of support across the SearchTogether interface. This has been

[9]in comparison to the other examples discussed in this thesis.

discussed to some extent already, as it allows users to spark new ideas for search by monitoring what has already been found. Where keyword search supports 23 of the tactics in a number of physical moves, the summary page supports 7 tactics, with 6 in only one move. Similarly, the query history view supports 5 tactics in only one move, but a further 7 tactics with only one extra physical move (clicking). The split search supports as many tactics as the basic keyword search, but in often twice as many moves. Similarly, the multi-engine search supports the same number of tactics as the basic keyword search, but often in 4 or 5 times the number of moves.

We see that some of the features only provide a relatively small amount of support, however without the Recommend and Thumbs Up/Down features, for example, we would see much less support provided by the summary panel. The chatting function, which provides direct communication with other searchers in the group, does support a wide range of tactics, however there is inherently a large number of steps involved in instant messaging, including waiting for responses and reading them, that dampens its strength compared to the simple interaction with the query histories.

Search Profiles

Using Figure 5.17 (a) (Graph G3) and (c) together, we can see that because the left hand side is higher, on average, than the right hand side, the design of SearchTogether provides a larger amount of support for users who are scanning, than for users who are able to specify what they need. This is supported mainly by the range of interface elements that support users in communicating results and sharing queries. A team of searchers, for example, would usually not be required for finding a single known resource (Type 15). The first and second quarters, and the third and fourth quarters, are typically balanced indicating that SearchTogether supports learners just as well as it supports searchers who are trying to find a particular resource or piece of metadata. The odd eighths of the graph (e.g. Types 1-2, 5-6, etc.) are significantly higher than their even counterparts. This indicates that SearchTogether is highly oriented to searchers who need to recognise important results when they find them. Much of the interface is recognising and learning from the queries used and discoveries made by other searchers. Finally, with the heavy emphasis on seeing other users queries, recommendations, and comments, the interface is highly geared up for searches dealing with metadata, rather than for actually finding specific web pages. Compared to the support for user types provided by just the keyword search function (shown in Figure 5.17 (b)), however, it is clear that users with known resources to find are largely unaffected by the design of SearchTogether (the pattern of Types 9-16 are almost identical in Figure 5.17 (a) and (b)), and the main effect is in the support provided to users who are scanning for potential resources that may help resolve their information needs. In fact, the most significant beneficiaries of the SearchTogether interface are Types 1-2 and 5-6, who are both scanning and having trouble specifying their needs.

The aim of the Sii framework is to identify potential issues with the types of tactics that different types of users can apply during search. Armed with such an analysis, like the one being presented here, designers can consider potential design changes, by aiming to address weakly supported tactics and user types. While clearly supporting users in learning from each others searches, the designers of SearchTogether, for example, may wish to try to support users in additional ways of manipulating, filtering, and re-distributing results amongst searchers (the CUT and CLEAVE tactics). Designers can quickly add new potential features to the analysis and see what affect it will have on the range of the support. Similarly, the strength of alternative designs to an existing feature can be directly compared both side-by-side in a graph similar to Figure 3.8.

Collaborative Searcher Roles

In terms of the CIS roles discussed in Section 5.2.2.2, we can see that SearchTogether is a great tool for Prospectors (primarily type 2, and sometimes type 1). The support is less oriented at Miners (primarily type 7 and maybe 8). In comparison to the support provided only by the keyword search function (Figure 5.17 (b)), however, the support for Miners (type 7) is still much higher. The balance of the odd and even quarters, in each half, means that Domain Experts and Novices are both well supported. The emphasis on the odd eighths of the graph, however, indicates that the support is particularly geared towards Search Novices. As mentioned already, the additional interface elements in Search Together are focused on learning from others, and so there is little additional functionality aimed at search experts.

As one of the concerns for evaluating collaborative search interfaces is that searchers can work together effectively, we can look for team dynamics that may be poorly supported, such as search experts working together. Design changes can be tested to see how example group dynamics are affected in the types of searchers shown in Figure 5.17. One finding from this analysis is that SearchTogether might try to support experts more directly by allowing them to explicitly coordinate less experienced co-searchers. Such experts may want to disseminate ideas to different users. Clearly, from the analysis above, this framework can provide insights into support for different types of collaborators during search. In the future, it might be interesting to analyse the desktop and mobile interfaces to the CoSearch system (Amershi and Morris, 2008), described briefly in Section 5.2.1.3. Such an analysis might tell us whether it is better to have Search Novices on the computer and experts with a portable device, or visa versa.

5.2.4 Validating the Analysis of Collaborative Search Interfaces

The remaining concern for this extension to the Sii framework is that it maintains the accuracy demonstrated in previous validations with solitary search interfaces (Chapter 4). In line with these previous validations, this section aims to correlate the results above

#	Key findings about SearchTogether produced by the analysis from our modified framework	Fact of the design	Empirically proven	Qualitatively reported	Not known, believed true	Not known	Not known, believed untrue
1	There are many ways for users to find good tactics taken used from other people.					X	
2	It is quite easy to see if someone has done a search before.			X			
3	Seeing queries associated with results in the summary page would further help users to know what searches have already been performed.			X			
4	Other existing features could be used to help users avoid repeating searches.				X		
5	The existing methods of keeping good resources are easy, but it could be improved.			X			
6	The only way to manipulate or reorder results is by performing a split-search.	X					
7	The interactions required to perform a multi-engine search could be cut down significantly.				X		
8	Providing separate single-interaction buttons for split- and multi-engine-searches would make them more accessible.				X		
9	The chat function is useful for many tactics, but slower than its alternatives (such as recommendations).			X			
10	The support provided by SearchTogether is more significant for users who are unclear about the sort of results they are looking for, but is less aimed at searchers who know what they are looking for.				X		
11	The support provided by SearchTogether is more significant for users who are unclear about their needs than those who can easily define their problem.						X
12	The majority of the interface is focused on using discovered results, rather than directly for searching.		X				
13	The three most useful features are the keyword search, the query histories, and the summary panel.			X			
14	SearchTogether might be improved by providing a feature that allows searchers to specifically suggest search terms into to-do lists for other users. I.e. enabling experts to help novices.			X			
15	SearchTogether could tabulate some results and provide filtering and sorting functions to manipulate them.				X		
16	A star rating, instead of a thumbs-up might allow grater expression and prioritization of the results in the summary.			X			
Totals		1	1	6	6	1	1

Table 5.2: A list of search-oriented usability statements classified by the designers of SearchTogether. Two columns representing empirical or qualitative disprove were not used and thus not included.

with the findings of user studies. Where in previous validations, however, results were directly compared with the findings described in publications, here results were communicated and discussed directly with the designers of SearchTogether (Morris, 2009). There are two advantages to these direct communications. First, as the version available online has been updated since the user study was published (Morris and Horvitz, 2007b), these communications provided insight into up-to-date known usability issues. Second, it provides the designers opportunity to comment on results discovered here that were not discussed in previous publications.

To validate the results, three aims were identified: to determine 1) whether the findings here are true; 2) whether any findings represent new insights into their designs; and 3) whether there are known search-oriented usability issues that we did not find. These three aims are discussed in turn below.

The Accuracy and Originality of the Results

To understand the first aim, a table of search-oriented usability statements was provided to SearchTogethers designers, containing both positive and negative statements produced by the extension. The designers were asked to then classify these results, as shown in Table 5.2, as: simply a fact of their design, empirically proven, qualitatively reported, not previously known but thought to be true, completely unknown, not previously known but thought to be untrue, disputed by qualitative reports, or disputed by empirical evidence.

First, as is clear from their absence in Table 5.2, the two latter untrue classifications, with either empirical or qualitative evidence, were not used by the designers of SearchTogether. Consequently, the first conclusion is that none of the results were known to be false. One statement (#12 in Table 5.2) was assumed to be incorrect. In correspondence, the designers felt that the support is approximately equal for searching and making use of good results. Consequently, faced with a novel insight that is not in line with their assumptions, the designers may now choose to keep watch for any future supporting or contradicting evidence. Similarly, one statement (#1) was not known either way. The extension here suggests that users can identify good search tactics taken by others, but the designers of SearchTogether have not explicitly examined the strategic improvements of novice users. Again, from the insight provided by the framework, the designers can keep a watch out for any evidence for and against this finding.

Of the 16 statements listed, one was listed as simply a fact of their design (#6). The process of performing this analysis, however, has highlighted this particular fact. This finding may foreground the issue to the designers and encourage them to explore additional functionality that integrates with some of their other previous work on grouping and organizing results (Morris et al., 2008). Another statement (#13) was deemed as empirically proven. In a previously reported study, which omits specific numbers, the three listed features were frequently chosen within the participants favourite features. A further six statements were listed as being reported during qualitative discussions with users. Statement #14, for example, is being explored in a separate companion-system for SearchTogether called CoSense (Paul and Morris, 2009). CoSense uses data from SearchTogether to provide an interface for making sense of an overall CIS session.

A further conclusion from these results is that the designers of SearchTogether have evidence from user studies that support a half of the statements produced by our analysis (8 out of 16). The associated conclusion, however, is that the remaining half of the statements represent novel insights, for which there is no contradictory evidence. Regardless of whether the designers believe them to be true (they only believed one to be untrue), they are now able to discuss, explore, and experiment with design alternatives that will provide additional evidence.

Missing or Unidentified Results

The combination of evidentially supported statements and still unproven insights provide strong support of the depth of analysis provided by the new extension, and additional evidence for the validity the Sii framework discussed in Chapter 4. In addition to reflecting on the results of this analysis, the third identified aim was to know if there were any additional known search-oriented usability issues that were not discovered. Ignoring implementation bugs, which are not the focus of the Sii framework, and sensemaking-based design considerations that led to a separate tool (Paul and Morris, 2009), the designers of SearchTogether highlighted four additional issues:

1. Users want to know if someone is peeking/following them, and if someone else in their group is peeking/following someone else.
2. It would be interesting to always see what URL a person is currently on, so you know if you want to take the time to peek or not.
3. Participants wanted to be able to edit and annotate the search summary pages.
4. Participants wanted a way to "push" a page to others (maybe a dedicated browser tab for each member of the group?)

The first of these four issues is more of a social monitoring issue, and involves a slight modification to the existing peek/follow tools. This suggested change is not a modification of the functional support for search, but instead affects the aesthetics of the user experience. Consequently, therefore, this change would not affect any search tactics, but instead simply provide awareness and perhaps comfort to the users.

Additional issues 2 and 3 appear to be valuable design directions. Providing an indicator of the current URL being viewed by each person, like the query histories, would support, in a single move, tactics including the CHECK, WEIGH, BIBBLE, SURVEY. Each of these tactics, however, is already well supported, and so the analysis did not highlight the need for an additional tool of this sort. Clearly, however, the current URL information will provide even more support for search awareness. Similarly, issue 3 is a modification to the already strong search summary feature. Again, as this feature is already prominent, the Sii extension's analysis did not highlight a need to further improve it. Although further improvements to strong areas of a design can be easily modelled within the Sii framework, such ideas will not be identified by designers searching for weaknesses in their interfaces.

Allowing users to directly push pages to other users (additional issue 4) is related to similar findings from the analysis above that led to usability statement 14 (in Table 5.2). Certain tactics, such as SELECT, FOCUS, CUT, and CLEAVE, are designed to break down searches or result lists. Pushing pages of results to other searchers may be a more manual alternative, therefore, to split or multi-engine searches. Alternatively, this tool could be used as a recommendation, or a pre-thumbs-up discussion. In this case, it can be perceived as an extension of the chat function, providing visual context to discussion. As part of discussion and communication, the tool may additionally support the already well supported CHECK, WEIGH, RESCUE, BREACH, and FOCUS tactics. Future work will be able to evaluate options for implementing these ideas as the designs become more concrete.

From analysing these additional undiscovered usability concerns, another conclusion is that, where within scope of the framework, they mainly extend already strong elements of the SearchTogether interface. While the designers of SearchTogether can easily evaluate these ideas, users of the Sii framework, and this extension, are unlikely to identify these types of issues while inspecting their interfaces for weaknesses. Consequently, this conclusion highlights that the framework has not missed any weaknesses in design, and will be able to analyse the benefits of adding such features to the interface when they have been designed.

5.2.5 Summary of the Collaborative Search Extension

This section has assessed an extension to Sii that enables the framework to inspect and analyse collaborative information seeking software. First, in Section 5.2.2, the theory used within the Sii framework was re-framed from the perspective of groups of searchers, explicitly collaborating on a shared task. Section 5.2.3 then described an application of the Sii framework, with the collaborative extension, to an example collaborative information-seeking interface: SearchTogether. The results and subsequent validation show that a) the framework can be just as easily applied to collaborative search interfaces as individual seeking software, using the new re-framed models; b) that the framework can still provide accurate an accurate analysis of a collaborative search interface; and c) can be used to suggest some redesigns to improve collaborative search interfaces.

The process of developing the necessary extension to the Sii framework, so that it can be applied to a special collaborative set of information seeking interfaces, has proved successful. While the previous extension described a new set of theories that provided a second set of measures and analyses, this extension affects the language of the main framework. Consequently, this extension does not produce a new set of analyses but allows the existing measures to be applied accurately in unusually but increasingly popular circumstances.

Much future work, however, remains for this extension too. While this work so far indicates that the the extension's revised models can produce accurate and insightful results, future work will concentrate on analysing its use in context of academic, and eventually, industrial development of collaborative ISIs. Notably, the language used here is a conversion from the language of an established research field to the terminology used by a new and developing community. Both the terminology and the models developed by this CIS community may still evolve dramatically. Like in the future work of the Sii framework, discussed in Chapter 6, it may take a number of years to identify the value and impact that this extension has on practitioners.

5.3 Developing Further Extensions in the Future

The early investigation into two extensions to the Sii framework have been discussed and initially tested. These two extensions, while motivated by the findings and case studies discussed in previous chapters, are not exhaustive of the possible extensions. Further, it is not unusual for similar models to go through such variations, as discussed at the beginning of this chapter. In a recent short paper describing the Sii framework (Wilson and schraefel, 2009c), a potential future investigation into mobile search was suggested. Mobile and ubiquitous computing is becoming more prominent (Greenfield, 2006), and so the special requirements of mobile search have received much focus. Although recent research has suggested that mobile search on increasingly powerful devices is becoming more similar to traditional search (Li et al., 2009), other research has focused on how we interact with mobile search (Karlson et al., 2006), the effects of location and movement while interacting with mobile search (Wilson et al., 2006a), and even shifts in the whole paradigm of mobile search (Jones et al., 2007). In particular, the notion of location-sensitive search is not captured by the theory currently used in the Sii framework or either of the two extensions described above. Such an extension to support the analysis of mobile search functionality could be a focus of future research.

Two key requirement for choosing appropriate theory for future extensions are that they are (a) well defined and have specific limits and boundaries, and (b) that they can be easily integrated within the current application procedure. These are both discussed below.

Section 5.1 above considered several options for evaluating the complexity of ISIs, including flow theory and the theory of engaging designs. These alternatives were not chosen because they typically contained retrospectively focused theory. The engagement research, for example, identified several factors of engagement that can be posed as questions to users about their experience. This retrospective view is similar to the NASA TLX scale described above, which depends on the subjective and retrospective views of users after experiencing a design. Cognitive Load Theory, however, states clearly that there are several factors of design that cause high levels of cognitive load. Further, these factors were defined and grounded in specific design circumstances. Cognitive Load Theory is like, therefore, the tactic and user profile models used by the Sii framework, and the group dynamic models used in the collaborative information seeking extension. If, in future work, the circumstances are identified that cause the factors of user engagement with interfaces, then the such a theory could be operationalised into an extension.

Another similarity between the two extensions described above is that they fit in closely with the existing procedure of the Sii framework. In fact, the collaborative information seeking extension does not alter Sii's procedure at all. The cognitive load extension, however, builds upon the second preparation task when using the Sii framework. Sii's users are already required to identify the features of the interfaces being evaluated by the framework. The cognitive load extension simply requires users to classify these features according to factors such as modality, and their similarity and distance to the other features. If not used as part of applying the existing framework, these extensions would instead be considered as separate usability evaluation methods.

Future extensions, including one for mobile search, will be designed according to these two requirements: for predictive and bounded theory, and integration into Sii's procedure.

5.4 Summary of Chapter

This chapter has made three contributions surrounding the initial research into future variations and extensions of the Sii framework for overcoming limitations noted in the research performed so far. Many usability evaluation methods, such as the cognitive walkthrough, heuristic evaluation, and GOMS, have been altered or extended for special circumstances.

First, an extension for measuring cognitive load was proposed and initially tested against the findings of the empirical studies discussed so far in the thesis. Specifically, some studies have indicated that participants have performed more efficiently on some search tasks using simpler interface designs. This extension was designed to measure the simplicity of a user interface, so that designers can predict that certain design options will not impose more cognitive load on users than the relative support provided for search. The results, while mostly promising, highlighted several areas for future research, including testing variations in the weightings used, modifying the assumptions made about temporal contiguity, and evaluating designs from studies that have specifically assessed cognitive load.

Second, an extension was proposed and tested that supports the evaluation of collaborative information seeking interfaces. The search theory, and consequently the language used in the Sii framework, assumes that searchers are acting as individuals. Instead, however, some recent research has studied how small groups of people collaborate over search, highlighting that many people watch over other peoples shoulders as they search, contributing ideas and suggestions. The recent focus, therefore, has been to support such scenarios, where people might be planning a holiday together, with more appropriate technology. It was noted, however, that solo-focused literature meant that the Sii framework did not easily describe the functionality supported for collaborative conditions. The theory used by the Sii framework was re-framed for the new extension and tested by studying a popular collaborative search interface: SearchTogether. The analysis provided promising results that were communicated and discussed with SearchTogether's designers.

Finally, this chapter described the key requirements and considerations for the development of future extensions. These requirements, for finding predictive, well defined theory that can be integrated into Sii's current application procedure, are discussed in the context of one idea identified for a future extension: mobile search. In particular, the extension would require predictive theory that captures the value and importance of location-sensitive search.

The research reported in this chapter has aimed to begin research into the some of the limitations discovered by studying the Sii framework so far. The results of building and testing the two extensions so far demonstrates that the Sii framework is both insightful and flexible. It is promising that it has been possible to overcome the limitations investigated so far, and research will continue into the development of these and other potential extensions in future work. Chapter 6 discusses the future work in more detail.

Chapter 6

Conclusions and Future Work

> *In Software Engineering: A Practitioners Approach, author Robert Pressman shows that for every dollar spent to resolve a problem during product design, $10 would be spent on the same problem during development, and multiple to $100 or more if the same problem had to be solved after the product's release.*
>
> **IBM, Cost Justifying Ease of Use, 2001**

Information Seeking is pervasive to both our lives and to our increasingly frequent experiences with computers. For online retailers and information resources, providing effective search functionality to users is imperative to improve throughput and sales. Tedeschi (1999) reported, for example, that by spending more than $1 million on improving the search functionality of the IBM website, sales support decreased 80% and sales improved 400%. The money invested by IBM was spent on 100 people working for 10 weeks on improving the quality of search provided. In a subsequent report, IBM also highlighted the evidence from Pressman (1992) stating that usability issues found during the design phases are an order of magnitude cheaper to resolve than during development, which is in turn an order of magnitude cheaper than after deployment. The aim of this work has been to identify search-oriented usability issues in the very early stages of design.

The chapters of this thesis report on a new Usability Evaluation Method (UEM), called Sii, which analyses the functional support provided by Information Seeking Interfaces (ISIs) for different search tactics and for different searcher profiles. Sii, which stands for the Search Interface Inspector, is an inspection framework that can be applied to even paper prototypes in a matter of hours and can estimate the support provided for different styles of search and from multiple user perspectives. So far, this work has researched, developed, tested, and revised the Sii framework, as well as having begun, as far as possible, to address some of the limitations found by creating conditional extensions. This chapter begins by summarising the research performed so far, before discussing the on-going work planned for the future. Much more can be done in terms of optimising the framework and improving the framework, while further demonstrating Sii's value for the information seeking evaluation. The chapter concludes with a discussion of the contributions made by the work.

6.1 Sii: a Summary

The research described in this thesis covers three main phases. After first reviewing related work into the information seeking, search interfaces, and evaluation methods, Chapter 3 discussed the design of the newly developed Sii framework. Chapter 4 then validated the Sii framework from a number of angles: the internal structure, the accuracy of results, and the framework's suitability for various practitioner contexts. Finally, based upon some of the limitations identified from the previous chapter, Chapter 5 discusses the initial investigation into overcoming these limitations. Each of these chapters is summarised in more detail below.

6.1.1 The Proposed Framework

The framework described in Chapter 3 has been designed to analyse ISIs in three ways. First, it analyses the features of search interfaces, such as keyword search boxes or browsing facets, to identify how they contribute to the overall support provided by an interface. Second, the interface is analysed for how it supports a set of 32 known search tactics. These tactics represent a range of behaviours, demonstrated in both physical and digital seeking environments, and include monitoring, query formulation, refinement, and idea tactics. The support for these tactics are then summarised into different user searching contexts, such as those who know what they are trying to find, and searchers who are intending to learn. The support for each of these searcher profiles is calculated by averaging the support provided for the tactics that they will likely need. To provide these analyses, the framework is built upon two models: a model of search tactics (Bates, 1979a,b) and a model of searcher contexts (Belkin et al., 1993). Further, a mapping was generated, which links key tactics to each searcher profile. This mapping, which has enabled the combined use of two IS models, represents one of the key contributions of the research presented in this thesis.

As an inspection method, grounded in information seeking theory, the Sii framework, and the analyses it provides, have five main advantages. First, the framework can be applied to even paper prototype designs, and guide design decisions before implementation begins. Second, the framework, which focusses on interface functionality alone, is both implementation and dataset agnostic. Consequently, any ISIs can be directly compared even if they provide different collections of documents. Third, where empirical evaluations of ISIs may tell us which design performs best for a given task, the framework's three analyses provide rich insight into the functionality provided and so can be used to explain the cause of any such findings. Fourth, the framework analyses ISIs from 16 different perspectives, where the tasks of empirical user studies typically cover 1 to 3 user profiles. This fourth benefit can provide a much more holistic view of an interface rather than a select view controlled by the user study conditions. Fifth, the framework can be applied in relatively little time compared to a user study. User studies, reported in the thesis above, have spent around 40 hours with users, with many more required for preparation and analysis. The application of the framework can take only a couple of hours, per interface, to both apply and analyse the results.

The Sii framework provides the first, to my knowledge, method that can be used to asses the support that different search interfaces provide for different searcher contexts, asking: 'Does this design support the right kinds of search?'. Many empirical measures have been produced for search-oriented user studies, but unlike the inspection methods provided by the HCI community, there are no methods that predict the suitability of search designs. Although other inspection methods have been produced in the HCI field, these often focus on the learnability or time-based efficiency of general software. For the HCI community, therefore, Sii provides a new method that evaluates their search interface designs using the expertise generated from the often distinct academic field of IS. The end of Chapter 3 compares Sii to other similar methods. Sii provides a similar, but less constrained functional analysis to GOMS, but a) with a specific search-oriented focus, b) a simple procedure similar to the Cognitive Walkthgouh, and c) requiring less expertise and time, like Heuristic Evaluation. In combining the benefits of these methods, Sii provides a fast and insightful analysis of functional support for search.

The framework, and the content of Chapter 3, provides three 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 convert established IS theory into an operationalised UEM for making predictive analyses of ISIs. The second contribution, as mentioned above, 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. Finally, a detailed analysis of three prevalent academic faceted browsers is presented, which provides some new insights into the support they each provide to different user types.

6.1.2 The Validation of Sii

The Sii framework was validated on three fronts in Chapter 4. Notably, these validations correlate closely the research questions listed in the Introduction (Page 3). The first research question, and its parts, relate to the way that a UEM can be built from IS theory. Although Chapter 3 described the structure, the first validation phase evaluated the way Sii was built. The second evaluation phase focused on the accuracy of results and Sii's ability to types of results produced. Additional validation, such as addressing false positives and false negatives, was further addressed by the example evaluated in Section 5.2. Finally, the third phase of validation addressed the third research question regarding the usability and suitability of the Sii framework for real usage scenarios. This third research question, however, is also further addressed by the research into extensions discussed in Chapter 5.

First, the structure of the framework was validated for both the IS theory used as building blocks, and the mapping used to join them together. The chosen IS models that were used to develop the framework were compared to their alternatives using three criteria: how established they are as theoretical models, their suitability in terms of size and scope, and, more objectively, their citation impact. The two models chosen, one for tactics and one for searcher profiles, were typically top in all three criteria, and are considered to be strong building blocks for the Sii framework. The mapping used to combine the two IS models was also validated, as it could not be produced by any objective metric that can be checked, but was built using the careful analysis of related literature. Consequently, the mapping required validation by achieving consensus between multiple independent judges. A combination of expert and novice participants reviewed a pre-prepared set of related literature and provided their own mappings between the two models. The collective consensus was used as a revised mapping, which had a 60% correlation with the original mapping. The variation between the six independently produced mappings, however, demonstrated that the process was non-trivial. The revised mapping provided more accurate analyses of ISIs.

The second validation was to show that the framework could produce analyses that correlated closely with the results of empirical studies. As well as demonstrating the accuracy of the three graphs, the validation process further validated 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, in much less time, and potentially at an earlier stage in design.

Finally, the third phase was to validate Sii's suitability for use by practitioners. Aside from demonstrating that the results are accurate, this validation process provided early insight into how the framework would fit into working practices, and whether different evaluators would produce the same analyses. First, a series of short case studies were presented that examined the use of the framework from different types of practitioners, including an academic project team and a search interface consultant working on multiple projects. Additional requirements for the Sii framework were produced that would aid secondary uses of the analyses, such as providing guided presentations of results to colleagues and the ability to prioritise design ideas when constrained by limited funding. Following the case studies, a set of pilot studies provided early insight into how recent graduates faired when using the framework for the first time. The pilot studies provided further insights into the way the method is presented online, and indicated that different independent evaluators could produce similar design improvements.

The contributions of this chapter were a) in clearly describing the decisions made while building the framework; (b) in providing further insight into the results of two academic studies; and (c) in demonstrating how the framework would be realistically used. The validation above stopped short, however, of reporting on the actual use of Sii by independent practitioners on projects with real constraints and stakeholders, which will begin to arrive as academics hear about Sii[1] and then reach the appropriate stage of projects to use it. The pilot studies tested the framework in simulated studies with students, but as the limitations of trying to generalise from simulated usage of UEMs are well known (Gray and Salzman, 1998), larger empirical tests of the Sii framework will be performed when the correct opportunities arise. The early pilot and case studies, however, have identified some of Sii's limitations, and the next section discusses some early extensions designed to overcome them.

6.1.3 Optional Extensions for Sii

Chapter 5 described two particular extensions to the framework. One concern highlighted by the case studies, was that some designs can provide less but more purposeful support for user types. Sii consistently rewards the addition of functionality, but Diriye (2008) discovered that a simpler basic search interface performed best for simple known-item search tasks. The first extension, therefore, investigated the notions of simplicity and complexity. An extension, which represents one of the key contributions from the chapter, was developed that operationalised Cognitive Load Theory (Chandler and Sweller; Paas et al., 2003a; Mayer and Moreno, 2003) in a similar manner to the way Sii was generated from IS theory. The extension, which estimates the mental load imposed on a user by the complexity of an interface, showed some promising results, but also some inaccuracies. Further work, described in the next section, is required, but another contribution of the extension was a cognitive assessment of three comparisons previously described above. Some new insights, over the novel findings produced by the Sii framework, were produced by analysing the new extension's results.

The second extension provides a conversion of the original framework into a version that can be used to inspect collaborative ISIs. The recent growth of collaboration IS (CIS) research would not be supported by the original framework, which was built on models of solo-focused IS behaviour. The first contribution of this extension was to re-frame some established IS models (those used in the Sii framework) from the perspective of searchers actively collaborating in small groups. Second, the re-framed models were used to evaluate a publicly available and popular collaborative ISI: SearchTogether.

[1]Apart from select academic workshops, Sii has only so just been published for wider audiences.

The final contribution of the chapter is in the discussion of how future extensions may be developed for contexts such as location-sensitive mobile search. Requirements are described for identifying usable theories and operationalising them in such a way that can be integrated into the Sii framework. The development of a mobile-search extension represents a small part of the future work that is planned for the Sii framework, and discussed below.

6.2 Sii: the Future

There is an abundance of future research that can still be performed on the Sii framework, and this section provides a map of how that work will likely be performed. Although Sii has been studied carefully over the last three years, the period of time associated with doctoral research is relatively small compared to the time it takes for UEMs to mature. The Cognitive Walkthrough, for example, was based on half a decade of research into the 'CE+ model', which in turn was based on research from the 1970's. After being first published (Lewis et al., 1990), the method then went through several variations over the following four years until a more mature version, now considered by many to be the standard procedure for applying a cognitive walkthrough, was published by Wharton et al. (1994). Then 6 years later, after the method had become popular, the streamlined Cognitive Walkthrough was presented by Spencer (2000), based upon the experience gained having used the framework many times within industrial projects. Similarly, other methods like SEEM and CASSM have been developed and studied for around 10 years (Blandford et al., 2008). Like the cognitive walkthrough, these methods were based on theory from another 10 years previously.

In comparison, while work into developing a method began three years ago, the first publications have only recently been released (Wilson et al., 2009b). If like the Cognitive Walkthrough, it may be another four years before a mature version of the framework is released based upon the continuing experience of applying and using the framework in longer, larger, and more realistic case studies. Case studies have become the de-facto method of studying UEMs, based upon the results produced by an international multi-institutional research effort over the last few years Law et al. (2009). Consequently, continuing and future research is aimed at developing and maturing the framework, and further establishing Sii in the HCI and IS fields of academia and industry.

The planned future work can be broken into three main agendas: 1) Developing, 2) Studying, and 3) Modelling. The first agenda, Developing, is primarily aimed at improving the online Sii framework, involving more pragmatic and programmatic aims. Studying, the second agenda, is perhaps the main research theme aimed at answering a number of remaining research questions, which are discussed below. Finally, the Modelling agenda is aimed at researching and, hopefully, further developing IS models and theories. Each of these research agendas have Short Term, Medium Term, and Long Term goals. Based upon the typical maturation span discussed above, these time spans are broken into 1, 4, and 7 year periods. The aim of achieving the long term goals, therefore, is in completing a roughly typical 10 year research period, including the doctoral research period, into the development and acceptance of the Sii framework. The goals in each of these research agendas are summarised in Table 6.1, and discussed in more detail in the following sections.

	Developing	**Studying**	**Modelling**
ST	- Interactive graph filters - In-line examples - Optional collaborative language	- Evaluate more designs - Engage with outside projects - Study Diriye's normalisation transform	- Cognitive Theory
MT	- Automatic analysis statements - Record move lists	- Find newly starting projects - Study learning curve - Investigate experimenter effect - Importance of tactics	- Track Engagement research - Investigate a mobile-search extension
LT	- Meaningful scales - Cognitive Load Extension	- What is Sii measuring? - Funded testing of Sii	- Track Information Seeking research - Compensating for reductionism

Table 6.1: Table listing the Short Term (ST), Medium Term (MT), and Long Term (LT) goals of three continuing research agendas into the Sii framework.

6.2.1 The Developing Agenda

This research agenda resembles, in a way, a software product development plan. The agenda contains short term minor improvements, medium term feature additions, and long-term development goals. The plan, however, both better supports practitioners in using Sii and enables continuing research in the two other research agendas discussed in the following sections.

Short Term.
Aside from small bug fixes and spelling corrections, the first planned additional feature is the ability to use the variables from one of Sii's three graphs as a filter over the remaining 2 graphs. Several participants noted that it would be hard to create re-designs that better support particular user profiles, without being able to specifically drill down to the tactics they depend upon. Further, however, the practitioner discussed in the second case study is regularly required to prioritise the features of a design plan according to the expected user base. For this practitioner, the filtering of and concentration on particular user profiles, features, and tactics was particularly important. The first task, therefore, is to design and implement functionality that supports this improved usage

scenario. Users will be able to click on bars, line markers, or x-axis labels. The result of such a click event, which can be captured by the javascript charting toolkit used[2], would be to temporarily filter the remaining graphs. If, for example, a user clicks on a particular searcher profile in Graph G3, Graphs G1 and G2 would filter to show the tactics that that user type depends on, and the features that contribute to that searcher profile. Further, the amount of each feature's support, since they may contribute to other tactics, would be shown, perhaps in context of the full support they provide. Representing the amount of change across multiple points within multiple graphs, will be part of this design challenge.

The second short term development goal, listed in Table 6.1, is to embellish the data entry process with both extended descriptions and familiar examples. Participants, particularly those with less expertise in IS, noted that the short descriptions of tactics were easiest to use while learning to use the framework. Further, however, participants regularly asked for examples for how tactics could be achieved with features of familiar search interfaces. Based on discussion with practitioners and participants, it would appear that these examples should be provided on demand, both optionally for all tactics, but also for each tactic individually. Further, extended definitions will also be made available on demand. Consequently, evaluators will have 3 complementary resources available while becoming familiar with the terminology used within the framework. One of aims of the Studying agenda, however, is to study the use of language in the framework to make it as accessible as possible for first time users.

The third short term goal for development is to build in a simple option that converts the IS language used in a Sii comparison to CIS language. If comparing designs of collaborative ISIs, users will be able to use this option to receive collaborative definitions throughout the application process. The short evaluation of the CIS extension showed promising results. Although the language used in the extension may require refinement in the future, the underlying process and data-entry are the same as the original framework. Further, by providing this optional switch of language, the framework would become available to this growing academic field, which has a fresh enthusiasm for evaluating the novel designs being produced.

Medium Term.
Within the next one to four years, one potentially valuable feature addition for the Sii website would be to automatically produce deductive statements about designs. The user profile graph, G3, could be automatically processed for differences in the halves, quarters, eighths, and sixteenths. These segments, as discussed in Chapter 3, relate to the dimensions that differentiate the user profiles. In doing so, statements could be automatically produced, such as 'Design A provides more support for searchers who can specify their need than those that cannot'. This development, however, should be carefully studied, as some evaluators may receive the statements and then neglect to

[2]http://docs.dojocampus.org/dojox/charting/

perform a more qualitative analysis of the graphs themselves. Further, research from the parallel agendas should indicate which sorts of statements will be valuable to users. Otherwise, it would be possible to make statements about each design being compared for each tactic, in comparison to every other tactic. For Graph G2 alone, 496 statements could be produced that directly compare any two tactics for one design. Analysing 3 designs, therefore, could produce 1488 statements. Comparing support provided by each design for every tactic would add another 96 statements. Further, an additional 32 statements could be made to say which design supports each tactic best, and another 32 to say which supports each tactic the least. Finally, another 6 statements could be made, stating which tactic is most and least supported by each design. From Graph G2 alone, therefore, a total of 1654 possible statements could be made. Research into adding automatic analyses, therefore, should focus on providing statements that highlight key distinctions between the designs involved in a comparison.

Another medium term feature development may be to extend the data entry process by requiring users to enter *a list* of moves required to perform a tactic with a feature, rather than simply entering the number of moves. So far, no participants have directly requested such a feature, but practitioners from the case studies regularly re-estimated their previously made counts. Capturing the moves at data-entry, however, would allow evaluators to review their analyses more easily, and also enable colleagues or other academic researchers to discuss them. It is not yet clear whether this would a) involve additional effort with little reward, or b) enable important governance over evaluations with an acceptable amount of extra work.

Long Term.
One key long term goals, although dependent on the progress in the Studying and Researching agendas, is to develop a more mature version of the Cognitive Load Extension, discussed in Section 5.1, available on the Sii website. Although it should not take a long time to implement, the extension requires further investigation such that it can be provided to Sii users to complement the current functional analysis.

The second long term goal identified in the Developing agenda, is to potentially develop more meaningful y-axis scales. Although the participants involved in the pilot and case studies did not query the meaning or importance of the scales used, the move counts entered into the system can become relatively confusing once inverted, summed, and averaged. This measure can be especially problematic if a design is analysed without another interface to compare against. Many other evaluation methods have clearly defined values and scales. The NASA TLX scale, for example, converts its entered data to a 100 point scale, so that it is easier for evaluators to judge the severity of results. Similarly, the GOMS method measures the efficiency of interface designs in time, where a better design supports task completion if fewer minutes and seconds.

Within Sii, the functional support per feature (depicted in Graph G1) could reach a maximum score of 32, if a feature supports all 32 tactics in a single move. This is very unlikely, however, and the highest score reached in the examples analysed in thesis was 12. The tactic support graph, however, can scale endlessly, where the maximum score possible would be equal to the infinite number of features supporting the tactic in a single move. The values in Graph G2, therefore, could be much higher when evaluating a whole interface. As individual features, however, are often designed to support certain types of tactics over others, these may support some tactics in a single move. Consequently, values nearing one, when analysing a single feature, are more likely than values nearing 8 when analysing interfaces with 8 features. Finally, Graph G3, in averaging the support provided by multiple tactic support values, also has a somewhat flexible upper limit. Again, although unlikely, this support could equal 8, if all 8 features support all the tactics that a user-profile depends on in one move. These chances, while higher when evaluating a single feature, are unlikely. Such scales may only be provided as the number of designs being analysed increase, when averages begin to identify what constitutes, for example, excellent support for a user profile.

6.2.2 The Studying Agenda

The main aim of the studying agenda is to perform empirical or qualitative research in order to answer a number of newly identified research questions.

Short Term.
The first of the short term Studying goals, is to continue evaluating designs as often as possible. A collection of analyses, including those presented in this thesis, are being collected on the Sii website[3]. Each example analysed provides more experience and will work towards achieving other aims, including establishing the meaningful scales discussed in the Developing agenda. While this approach develops my own understanding and experience of Sii, the second short term Studying goal, is to engage with practitioners in various related communities. One suggestion, from more informal discussions with academics and professionals, is to engage with the CODE4LIB community, which includes a large mailing list of digital library designers and developers. Such engagement might enable the investigation of additional case studies, or simply encourage others to use the Sii framework for their own gain. At the very least, such community engagement should provide valuable pragmatic feedback.

A third short term Studying goal is to investigate further the value of the normalisation transform suggested by Diriye and colleagues in the first case study discussed in Section 4.3. Although the results of the application to existing examples has provided mixed results, more specific investigations directly into the value of the transformation should be performed. In line with such a study, would be to test the following research question:

[3] http://mspace.fm/sii

is supporting as many tactics as possible better than supporting specific tactics more directly? The notion that more targeted support for specific tactics and searcher profiles, using a reduced set of features, may provide more evidence for the transform. More directly, however, targeted empirical studies can be performed to demonstrate that a transformed analysis is or is not more accurate than a standard analysis. If for example, certain designs are promoted for certain searcher-profiles by the transformation, than these user-profiles can be specifically targeted with study tasks.

Medium Term

The medium term of the Studying agenda contains the largest number of goals that answer research questions. There are several elements of the framework that can be optimised or improved, and so require further study.

The first medium-term aim is to try and collaborate with practitioners on working projects. Maintaining relationships with practitioners who have shown interest in the framework should enable the independent or assisted use of the Sii framework in real usage scenarios, rather than as a means to discuss hypotheticals about how a completed Sii analysis could have influenced a design process. As discussed in Chapter 4 design and development projects often involve multi-year timescales, and so finding projects at the right stage of pre-development design is crucial for evaluating Sii's use in real scenarios influenced by real stakeholders and budgetary constraints. While a number of practitioners have shown interest in the framework, they are mostly keen to use Sii when the next project begins. Further, while practitioners are interested in using a Sii analysis, it is a much larger commitment for them to make decisions based on a relatively new method. Larger, more committed uses of Sii are discussed in the longer term goals. The aim here, however, is to get practitioners to use the framework and decide for themselves on how and when to use Sii again in the future.

The second medium-term goal is to study the learning curve of Sii's first time users. Although the previous aim should enable the study of the learning curve, to some extent, it should also be possible to empirically study potential improvements and clarifications to the Sii website by measuring average learning time required, or by counting the number of questions asked, by first time users. Although the pilot study investigating first time Sii users, in Section 4.3, showed that a between an hour and two hours was not enough time for some masters students, the practitioner in the second case study was able to guess many of the definitions and terminology without being asked. It may be interesting, therefore, to also study the effect that expertise has on the learning curve, with regards to experience in a) usability evaluation and b) information seeking.

The third medium-term goal is to investigate the effect that different individual evaluators have on the outcome of an analysis (Hertzum and Jacobsen, 2003; Cockton and Woolrych, 2001). It was promising that the same range of re-design suggestions came from both a group and four individual pilot study participants in Section 4.3. The evaluator effect, however, can influence the use of Sii framework at all stages from preparation to interpretation. Again, it should be interesting to further assess the effects of a) expertise and b) variation in the instructional language used on the Sii website.

Finally, the fourth medium-term goal in the Studying agenda is to investigate whether some tactics are more important than others. At the moment, the framework highlights important tactics for each searcher-profile. The practitioner of the second case study suggested that some tactics may always be more important to support than others. Kim (2009), after revising the structure of user profiles (Belkin et al., 1993) for more web-specific cases, began to simply count the instances that tactics were performed while using a typical search engine. Although this provides a simple quantitative measure for prioritising tactics, one of the main positions upheld by the Sii framework is that interface designs influence a user's ability to perform tactics. It may be that tactic instances are counted during observed or remote experience with a wide range of interfaces. Further comparing the number of times tactics occur in relation to support for each tactic determined by Sii may reveal whether frequency of tactic use is independent of, or influenced by, design.

Long Term.
The first long term goal of the Studying agenda is further investigate the types of results that Sii produces. Although clearly a functional analysis, the findings have correlated, in the examples analysed in this thesis, with task performance and user preference. In the evaluation performed by Capra et al. (2007) for example, discussed in Section 4.2.2, Graph G3 correlated closely with the empirical measures taken on the study tasks. Further, Graph G1, and some supporting results from G2, correlated closely with the most and least preferred elements of each design. It may be a valuable resource to study these different correlations more closely, especially when determining which designs to use after a Sii analysis.

The second long term goal, and perhaps the most significant active task to undertake for the Sii framework, is to gain the funded support to enable the design or re-design of a system from requirements gathering to full deployment. The value of such a large fully-funded project is to go beyond contributing to projects in their early stages and fully demonstrate, using before and after studies and Sii inspections, that designs can be improved through specific use of the Sii framework. Seeking such a scenario is important for a number of reasons. First, while Sii can be used during the design process of a short academic project, such projects rarely get deployed or receive wide-scale uptake from the public. Academic ISIs are usually developed as a means to empirically validate a hypothesis. Consequently, to really validate the value of Sii, a re-design should be validated both by short empirical studies and long-term usage. Second, while Sii can be used within other projects, their aims are unlikely to be to demonstrate that Sii is useful, but to achieve other goals set by the project funding. Unique funding is required, therefore, to enable the introduction of additional user studies focused at further validating the Sii framework. These user studies would not be aimed at testing the improved user interface, but at testing the value of the Sii framework. Third, funding would be required to monitor the on-going use of the software, perhaps through log analyses, to demonstrate that Sii has had a positive long term impact on the usability of the interface. Notably, this long term goal may be one of the key contributions to the maturation of the framework, as it would validate Sii's use in more critical and realistic contexts.

6.2.3 The Modelling Agenda

The Modelling agenda is focused on the continuing development of models and theories. The goals set here are largely directed but purposefully undefined, so that open-ended research can be performed to investigate a wide range of theory.

Short Term.
The main short term goal is to continue researching cognitive models and theories. The extension to the framework considered in Section 5.1 showed some promising results, but also some inconsistencies with expectations. Unlike my developing expertise in information seeking and human-computer interaction, I have relatively little experience with cognitive psychology. Consequently significant additional research is required that may lead to modifications that improve the cognitive load extension to the Sii framework.

Medium Term.
While the short term goal is focussed on developing the cognitive research, the first medium term goal is to track the progression of the other promising theories of complexity. In particular, the continuing work into engagement (O'Brien and Toms, 2008) may provide appropriately defined insight into what causes people to disengage with a user interface. Such developments in disengagement could complement, support, or even replace the cognitive load theory being used so far. Further, the use of fNIR technology to perform direct measures of cognitive load in the brain Hirshfield et al. (2009) also shows promising developments for validating the cognitive load extension over the next 1-4 years.

The second medium term research goal is focused on potential alternative extensions. So far a mobile, location-sensitive search extension has bee proposed, and investigation into the related theories and the generation of an initial extension will begin over the next 1-4 years.

Long Term.
Even within the period of this research, new or modified theories have been developed and released that shed new light on the way we search for information. These discoveries may affect the validity of the theory used in the framework. In the worst case, research

may disprove the theories used directly by Sii, which would require significant changes to the framework. The main focus of the long term goals for the Sii framework is to monitor the developments in information seeking theory, especially as the focus of research, according to the trends discussed by prominent academics like Järvelin and Ingwersen (2004), is moving towards interaction and user contexts.

A final goal of long term for the framework is to investigate what it means to be a reductionist. Theories and abstractions, by nature, are limited representations of certain elements of a phenomenon. The searcher profiles provided by Belkin et al. (1993), for example, are abstracted into 16 conditions using four elements of user contexts. As clearly demonstrated by the much larger model provided by Cool and Belkin (2002), there are many more aspects to information seeking contexts. Consequently, the nature of using theories to inspect designs, makes reductionist conclusions based on a limited view of users, and so they do not cater for the implications of behaviour not covered by the abstractions. Such nuanced or exceptional behaviour may be what makes a design popular or even significantly better than other designs. It is important, therefore, that even if Sii makes reductionist conclusions, that the framework's results are accepted and used appropriately. As part of the on-going research into information seeking, it will be important to investigate what it means for Sii to be a reductionist method, and try to understand what effect it may have on real users, evaluators, designers, and even project stake-holders.

6.3 Contributions of this Research

There are five key contributions made by this doctoral work, produced through the phases of design, development, validation, and extension of the Sii framework. These five key contributions, and their parts, are listed below.

1. **The framework.** The first and foremost contribution produced by the work is the Sii framework. In producing the framework, however, several smaller contributions were made:
   (a) A mapping to join two key information seeking models together.
   (b) The operationalisation of these models.
   (c) Several analyses providing novel insights into the design of information seeking interfaces.
2. **The Validation.** The second key contribution is the combination and variation used in validating the Sii framework from several angles. These validations provided novel insights into validity, accuracy, and practicality of using the Sii framework. Again, several smaller contributions were made during the validation process:

   (a) Novel insights into a previously published academic study.
   (b) Early indications into the consistency of use and the challenges therein.
   (c) Qualitative accounts of the perspectives of multiple and differing primary stakeholders of the Sii framework.

3. **The Complexity Extension.** The third contribution is in the promising investigation into using Cognitive Load Theory to mitigate one of the notable limitations of the Sii framework. Although in the early stages, the investigation showed that a cognitive model could be integrated into the Sii Framework, and more than sufficient evidence was provided that it could generate many accurate insights into the design of Information Seeking interfaces. Further, the investigation showed that the new measure could add a new dimension to the Sii framework, and allow evaluators to consider trade-offs between functionality and complexity during the design process.

4. **The First Generalisation.** The fourth contribution, proposed as an extension, or modification, to the language used by the framework, showed that the analysis approach could be generalised to support a specialised form of information seeking. The translation of the Sii framework to collaborative circumstances, and provides the first step towards showing that the technique can generalise to alternative models.

5. **The Sii Website**. The final key contribution of the doctoral work is the online tool that has been made available to both academic and industry practitioners. Producing the online tool makes the Sii framework accessible to anyone that wants to apply the Sii framework, without first having to re-create the analyses and procedures involved. Available for free online, the tool will also allow the community to discuss both the library of analyses being collected online, and the value of the framework.

6.4 Final Remarks

The Sii framework described in this thesis has been designed to provide a new type of analysis for information seeking interfaces to investigate a) the functional support provided by different interface features, b) the amount of functional support provided for different known search tactics, and subsequently c) the amount of functional support for different searcher profiles. The framework takes a holistic view of a search interface, by systematically and thoroughly analysing the whole interface and viewing it simultaneously from many user perspectives. The Sii framework also overcomes many barriers that may otherwise limit evaluation, comparison, and improvement of search interfaces, including: access to remote search systems, the influence of datasets, the evaluation of early search prototypes, and the identification of causal factors in empirical results.

The aim of the research has been to provide mutual benefit for both Information Seeking experts and Human-Computer Interaction experts. For Information Seeking experts, Sii provides a Human-Computer Interaction method to evaluate their designs. For Human-Computer Interaction experts, Sii enables them to utilise Information Seeking expertise when evaluating their novel interface designs. Further, the aim has been to support both groups with a method that is both fast and lightweight, so that practitioners can quickly analyse the strengths of weaknesses of anything from prototype designs to established systems, and do so from multiple user perspectives.

The results produced so far are promising and demonstrate that Sii can produce rich, insightful, and accurate analyses of information seeking interfaces. Both academic precedence and the future work described above, however, reveal that the findings so far represent only the first stages of developing, evaluating, and *understanding* new usability evaluation methods. Research into the framework is on-going and the next stages involve Sii being used in real and critical projects. Such work will continue to investigate the real impact that Sii will have. With this continuing research and development, however, Sii has the potential to enable search interface designers in making the right decisions that will a) reduce development costs, b) increase revenue for businesses, and c) improve the increasingly frequent searching experiences of users everywhere.

Appendix A

Full Definitions of the Search Tactics Defined by Bates

Tactics		Detailed Definition
Monitoring Tactics	CHECK	To review the original request and compare it to the current search topic to see that it is the same.
	WEIGH	To make a cost-benefit assessment, at one or more points of the search, of current or anticipated actions. Among other things, the searcher might consider whether any other approach would be more productive for the effort.
	PATTERN	Frequent experience with a type of question may lead to an habitual pattern of search. If, for example, a common request in an academic library is for addresses of researchers, then the librarian may soon develop a sequence of sources to search, arranged by their likely productivity. To PATTERN is to make oneself aware of a search pattern, examine it, and redesign it if not maximally efficient or if out of date.
	CORRECT	To watch for and correct spelling and factual errors in one's search topic. These may exist in the topic as presented originally by the user, or may slip into the searcher's thinking in translating a verbal request, or in remembering (without having in hand) a written request. In observing bibliographic searching done by several librarians, Carlson noted that the searchers would allow inaccuracies, particularly spelling errors, to slip into their search formulation. One librarian, for example, had a request on neuroglia, and searched instead on neuralgia, a very different concept. He noted several cases where a difficult technical term was not written down and the librarians would search for the remembered spelling, usually not find it, and then stop the search for that term.
	RECORD	To keep track of trails one has followed and of desirable trails not followed up or not completed. In complex searches it is sometimes necessary to return to the source of information or citations recorded earlier in the search. For example, after recording a number of citations from a periodical index, the searcher may then attempt to retrieve the articles cited and find a blind lead. The citation needs to be checked again in the original source. But unless the source, volume date, and subject term searched under were recorded, the searcher may have to go through the entries under a dozen terms or in several volumes to locate the desired citation. Similarly, if productive on-line and manual bibliographic search formulations are retained, later repeat effort may be saved.
	BIBBLE	To BIBBLE is to look for a bibliography already prepared, before launching oneself into the effort of preparing one. More generally, to BIBBLE is to check to see if the search work one plans has already been done in a usable form by someone else.

Continued on next page

Table A.1 – continued from previous page

Tactics		**Detailed Definition**
File Structure Tactics	SELECT	To break complex search queries down into subproblems and work on one problem at a time. This tactic is a well-established and productive technique in general problem solving. As each subproblem is solved, the parts can then be knit into a solution to the whole, larger problem.
	SURVEY	To review at each decision point of the search the available options before selecting. In Carlson's description of human searching behavior, he noted the following problem: There is almost no look-ahead in the human search procedures. All of the librarians studied exhibited to some extent this lack of look-ahead. They would often scan each entry as they came to it and then encounter a heading which would alter the search procedure. He concludes: Here the lesson is very clear: humans should scan over a reference document before making any detailed searches through it [5, p. 35]. Psychologically, this is a problem of going for closure too soon, that is, settling on a source or approach prematurely. In employing SURVEY, one resists that temptation and presumably achieves a more effective search.
	CUT	When selecting among several ways to search a given query, to CUT is to choose the option that cuts out, eliminates, the largest part of the search domain at once. In my opinion, this tactic is of fundamental significance in our field, and is relatively little known or discussed. Here are some examples: When looking up a book written by Smith and Brzustowicz, the search will be much briefer if one looks under Brzustowicz (assuming the file has entries under co-authors). In most files, there will be far fewer entries to scan under the latter name. Thus, in choosing to search under the latter name, with its few entries, one has cut out a larger part of the search domain than would be the case when searching under Smith, and has shortened the search accordingly.
	STRETCH	Naturally enough, one tends to think about information resources in terms of the uses for which they are intended. However, almost all reference sources can be used productively for some other purpose than intended. The internal organization of a file or reference book is designed around certain uses. Thus, access via certain record elements is provided, and access via other elements is not. But even though formal access is not provided, that other information is there in the source nonetheless. Introductions, which are outside the formal internal file organization of an information source, may also be informative in unexpected ways. Thus, to STRETCH is to use a source for other than its intended purposes. However, it should be kept in mind that to STRETCH effectively the searcher must first think differently, he/she must think about all the information that is in a source, not just about the ordinary uses of it.
	SCAFFOLD	Hodnett discusses the use of what he calls auxiliaries which are aids in problem solving which may or may not themselves be a part of the solution, but which make the solution possible. The technique of using auxiliaries is often employed in mathematics, where a seemingly irrelevant theorem is introduced, a theorem with little intrinsic interest, but one that enables the main theorem to be proved. Thus, to SCAFFOLD is to design an auxiliary, indirect route through the information files and resources to reach the desired information. For example, after unsuccessfully seeking information on an obscure poet, the searcher may find out who the poet's contemporaries were and research them in hopes of finding mention of the poet.
	CLEAVE	To employ binary searching in locating an item in an ordered file. (For those unfamiliar with this principle: In binary searching one first looks at a record in the middle of an ordered, e.g., alphabetized, file. One then determines the half of the file in which the desired record must lie. Then the middle record in that half of the file is looked at, and the quarter of the file in which the record must lie is determined. Then one looks at the middle record in the quarter section of the file, and so on until the desired record is discovered.

Continued on next page

Table A.1 – continued from previous page

Tactics		Detailed Definition
Search Formulation Tactics	SPECIFY	To search on terms that are as specific as the information desired. Specificity is one of the crucial concepts in systems of information access. Almost all systems of classification and indexing require that descriptions assigned to materials be as specific as the content of the materials and as the indexing system itself allows. Sears and Library of Congress subject headings use the rule of specific entry which requires entry of materials under the most specific terms that still encompass the content of the item; coordinate indexing, with its focus on concept indexing, brings about highly specific description, and so on.
	EXHAUST	To include most or all elements of the query in the initial search formulation, or to add one or more of the query elements to an already-prepared search formulation. Both this and the next tactic, REDUCE, are related to Lancaster's use of "exhaustivity". In searching, the more exhaustive a search is, the more of the elements of a complex request have been included in the search formulation.
	REDUCE	To minimize the number of elements of the query in the initial search formulation, or to subtract one or more of the query elements from an already-prepared search formulation. REDUCE is the opposite of EXHAUST. This tactic reduces the number of ANDed elements in the search formulation, making the search specification less stringent, and thus increases the number of documents likely to be returned on a search.
	PARALLEL	To make the search formulation broad (or broader) by including synonyms or otherwise conceptually parallel terms. PARALLEL and PINPOINT deal implicitly with elements in a query that are to be ORed together. Though these tactics are most readily applied in on-line Boolean searching, they may also be used in manual searching. For example, in the process of manually compiling a bibliography, one may look over catalog subject headings and terms in periodical indexes and expand the number of similar terms searched under (PARALLEL), either at the beginning of the search or after getting some experience with the type and quantity of materials under each term.
	PINPOINT	PINPOINT. To make the search formulation precise by minimizing (or reducing) the number of parallel terms, retaining the more perfectly descriptive terms. PINPOINT is the opposite of PARALLEL.
	BLOCK	To reject, in the search formulation, items containing or indexed by certain term(s), even if it means losing some document sections of relevance. This tactic deals implicitly with the Boolean AND NOT. The term NOT was not used, however, because the concept extends beyond the usual applications of Boolean searching.
Term Tactics	SUPER	To move upward hierarchically to a broader (superordinate) term. Searchers may be assisted by pointers in a thesaurus or may have to rely on their own knowledge to devise the term.
	SUB	To move downward hierarchically to a more specific (subordinate) term.
	RELATE	To move sideways hierarchically to a coordinate term.
	NEIGHBOR	To seek additional search terms by looking at neighboring terms, whether proximate alphabetically, by subject similarity, or otherwise. Coates pointed out many years ago that all manual (and we should add today, most automated) information organization systems do two fundamental things: locate and collocate. To use this tactic is to expand the search by examining the proximate entries, whatever they are. In on-line searching, one examines whatever proximate entries are made available by the on-line program one is using. Incidentally, the use of NEIGHBOR may be extended beyond term selection to resource selection as well. Since classification systems collocate books, it is easy to extend a search by examining related sources collocated on the shelves of the reference stacks.
	TRACE	To examine information already found in the search in order to find additional terms to be used in furthering the search. Two of the most common ways of doing this are to scan descriptor term lists in citations retrieved in on-line searching, and to scan on a catalog card the list of other headings that have been given to the document in question. These other headings on the catalog card are called the "tracings," hence the name for this tactic.

Continued on next page

Table A.1 – continued from previous page

Tactics		Detailed Definition
	VARY	To alter or substitute one's search terms in any of several ways. See remaining term tactics for some specific variations.
	FIX	To try alternative affixes, whether prefixes, suffixes, or infixes. Several may be done at once through truncation routines.
	REARRANGE	In any system where terms may contain more than one word, word order may make a difference in retrieval success. To REARRANGE is to reverse or rearrange the words in search terms in any or all reasonable orders.
	CONTRARY	To search for the term logically opposite from that describing the desired information. For example, one may want information on cooperation and, after an unsuccessful search, change the term to competition.
	RESPELL	To search under a different spelling. CORRECT dealt with maintaining correct spelling, among other things. But with RESPELL the concern is not with correctness, but with effectiveness. Particularly in current on-line search systems, there are a great many spelling variations that show up in the citations. One must expand the spelling variations to insure good recall. RESPELL is occasionally needed in manual systems too, where, for example, one needs to change from U.S. to British spelling to search successfully in a source.
	RESPACE	Spacing, particularly in hyphenated words, or words that appear with various spacings, can be critical in search success. To RESPACE is to try spacing variants. While spacing problems are most glaring in some automated search files, such problems can also be serious with manual files. The two fundamental variants in filing rulesword-by-word filing and letter-by-letter filingdiffer on how the blank space is to be treated in filing [29; 30, p. 339]. Both of these rules are in wide use. The searcher who is thinking in terms of one filing rule and enters a source that uses the other may miss the desired material.
File Structure Tactics	RESCUE	To check for possibly productive paths still untried, in an otherwise unproductive approach.
	BREACH	To breach the boundaries of one's region of search, to revise one's concept of the limits of the intellectual or physical territory in which one searches to respond to a query.
	FOCUS	To look at the query more narrowly, in one or both of two senses: (1) to move from the whole query to a part of it or (2) to move from a broader to a narrower conceptualization of the query.

Table A.1: The 32 Tactics using original definitions from Bates (1979b,a)

Appendix B

Extra Graphs

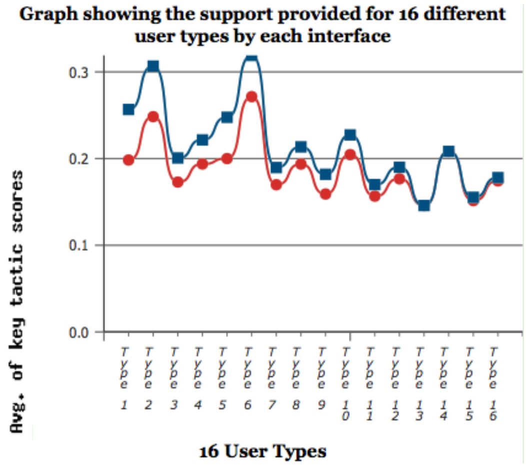
This Appendix includes a series of extra graphs for reference. While these graphs were not necessary for the discussion included in the main document, they are available here for inspection, should readers wish to review them in more detail. All graphs are also available online. All G3 graphs should be used in reference to the 16 user profiles included in Figure B.1.

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

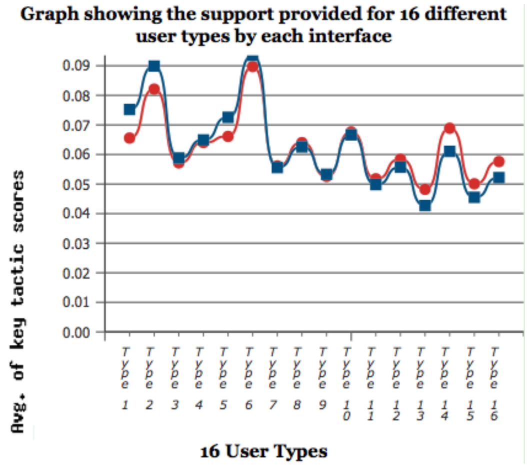
FIGURE B.1: The 16 searcher profiles identified by Belkin et al. (1993), used with the Sii framework.

B.1 Diriye's Normalisations

This section includes the the original G3 graphs from three examples discussed in the dissertation, along with the versions normalised by the method suggested by Diriye (2008), discussed in Section 4.3.



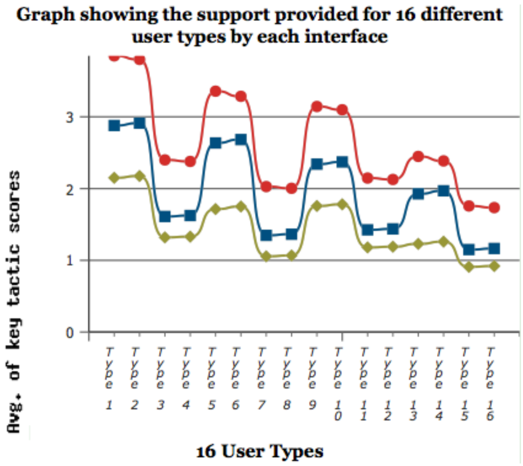
(a) Original



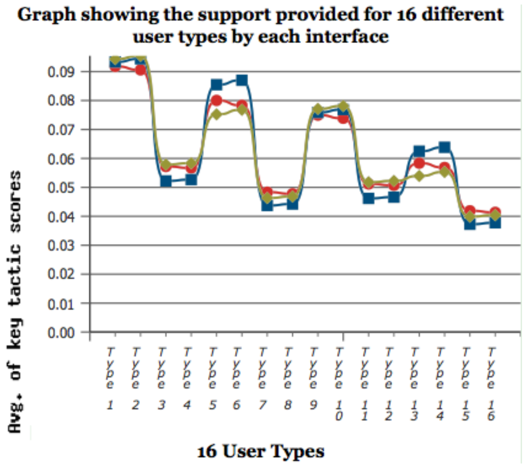
(b) Normalised

(c) Legend

Figure B.2: The original and normalised versions of Graph G3 from the Backward Highlighting example discussed in Section 4.2.1.



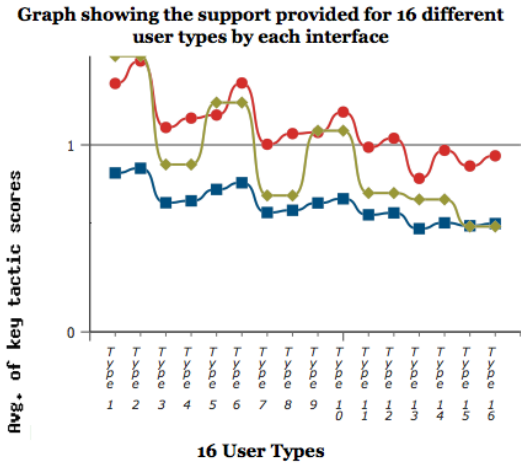
(a) Original



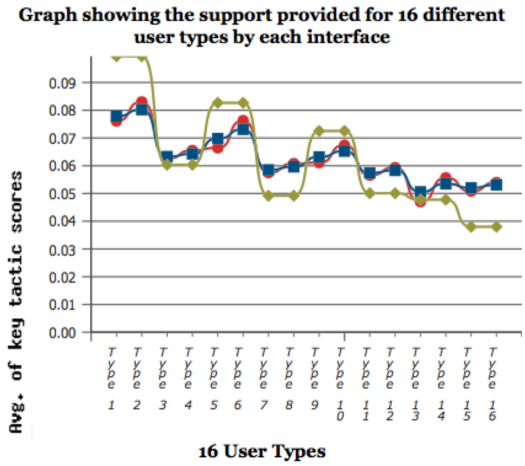
(b) Normalised

(c) Legend

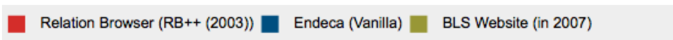
FIGURE B.3: The original and normalised versions of Graph G3 from the three faceted browsers analysed in Section 4.1.4.



(a) Original



(b) Normalised



(c) Legend

FIGURE B.4: The original and normalised versions of Graph G3 from the three browsers from the study by Capra et al. (2007), analysed in Section 4.2.2.

B.2 RYA Old and New Designs

These graphs compare the old RYA website, and the newly launched website. The analysis was studied together with an information architect at the design company, and the discussion that followed is described in Section 4.3.

FIGURE B.5: Legend for comparing the old and new designs of the RYA website.

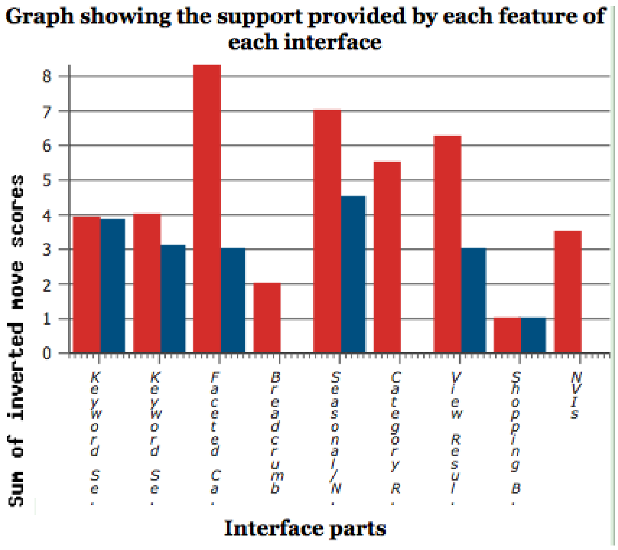


FIGURE B.6: Graph G1 showing the support provided by the features of the old and new designs for the RYA website, discussed in Section 4.3, where taller bars represent stronger support.

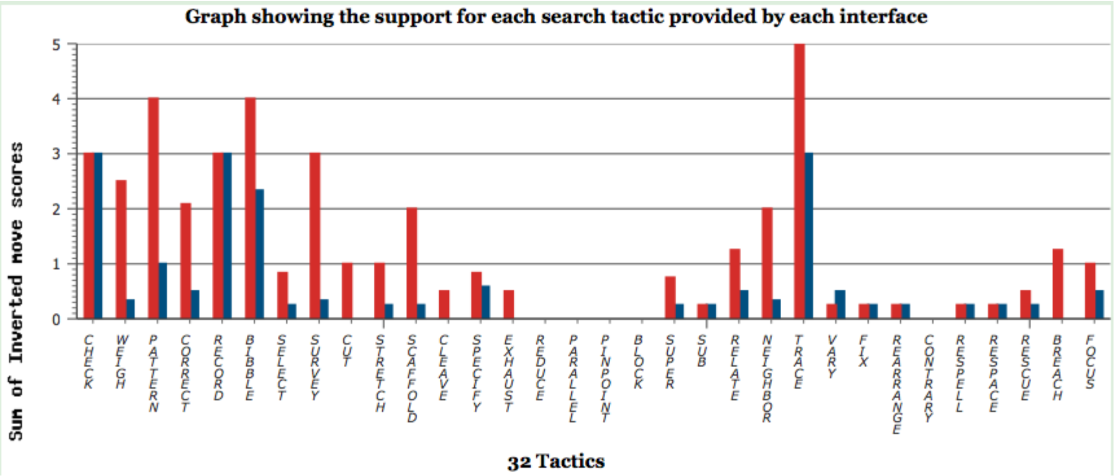


FIGURE B.7: Graph G2 showing the support provided for 32 search *tactics* (Bates, 1979b,a), provided by the old and new designs for the RYA website, discussed in Section 4.3, where taller bars represent stronger support.

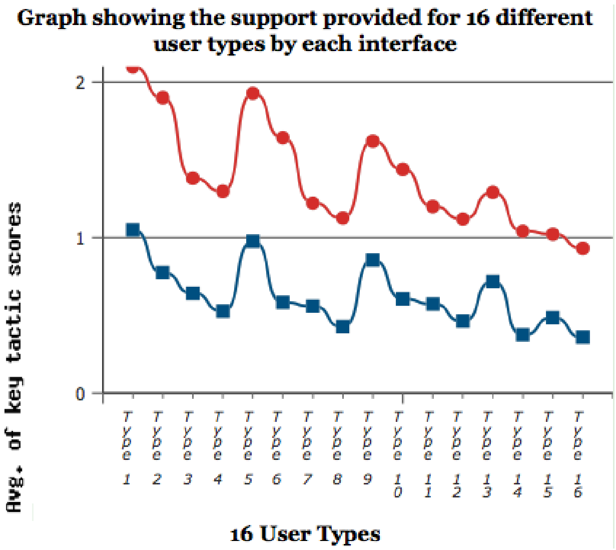


FIGURE B.8: Graph G3, showing the support provided for 16 searcher profiles (Belkin et al., 1993), by the old and new designs for the RYA website, discussed in Section 4.3, where peaks represent stronger support.

B.3 RYA Old, New, and Re-Design

These graphs were also created and discussed with the information architect of the company that redesigned the RYA website. The previous analysis was used to create design recommendations, and results of these suggested changes were added to the graphs for comparison. The graphs show that some significant improvements could still be made. These changes were discussed in more detail in third case study in Section 4.3.

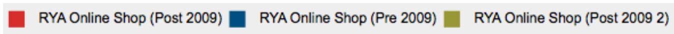


FIGURE B.9: Legend for comparing the old, new and re-designs of the RYA website.

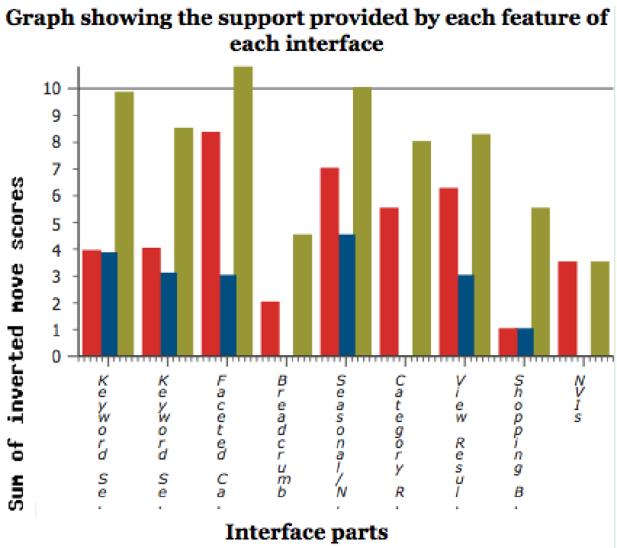


FIGURE B.10: Graph G1 showing the support provided by the features of the old, new, and re-designs for the RYA website, discussed in Section 4.3, where taller bars represent stronger support.

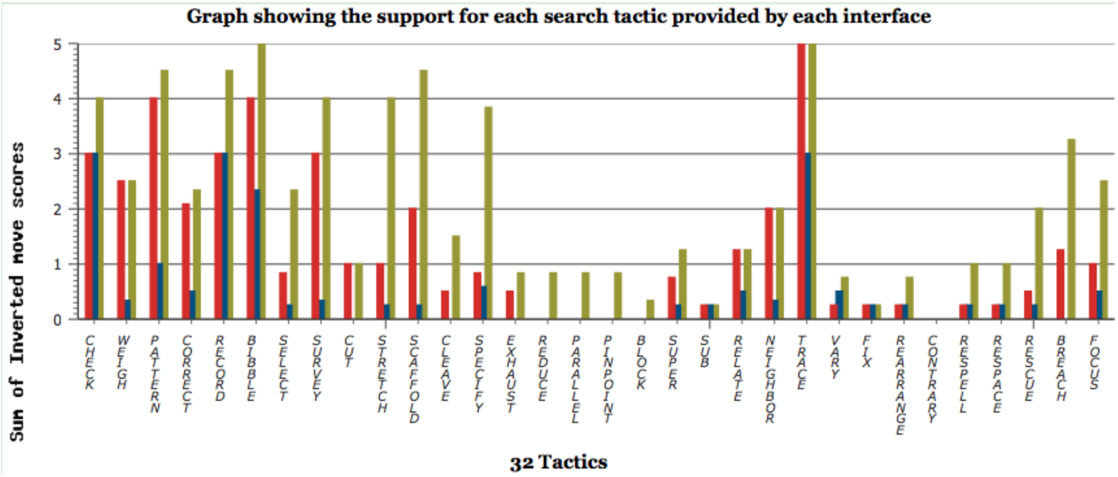


FIGURE B.11: Graph G2 showing the support provided for 32 search *tactics* (Bates, 1979b,a), provided by the old, new, and re-designs for the RYA website, discussed in Section 4.3, where taller bars represent stronger support.

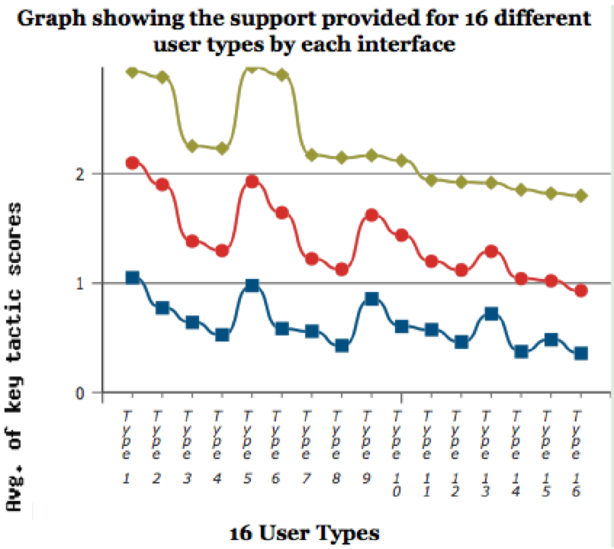


FIGURE B.12: Graph G3, showing the support provided for 16 searcher profiles (Belkin et al., 1993), by the old, new, and re-designs for the RYA website, discussed in Section 4.3, where peaks represent stronger support.

B.4 RYA Keyword Search Comparison

Finally, for additional comparison, the keyword search element of the old and new RYA websites were compared to the functionality provided by Google. The aim was to highlight the breadth of support that could be provided by keyword search, to demonstrate the value of improving the search interface.

FIGURE B.13: Legend for comparing the Keyword Search for the old and new designs of the RYA website compared to Google.

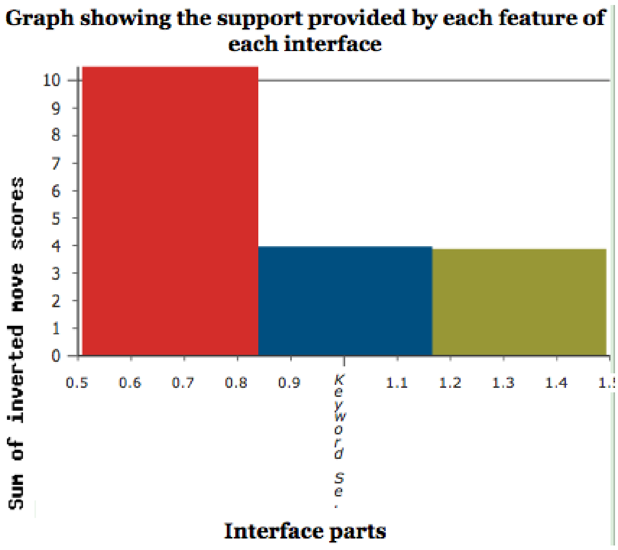


FIGURE B.14: Graph G1 showing the support provided by the keyword search of the old and new designs for the RYA website compared to Google, discussed in Section 4.3, where taller bars represent stronger support.

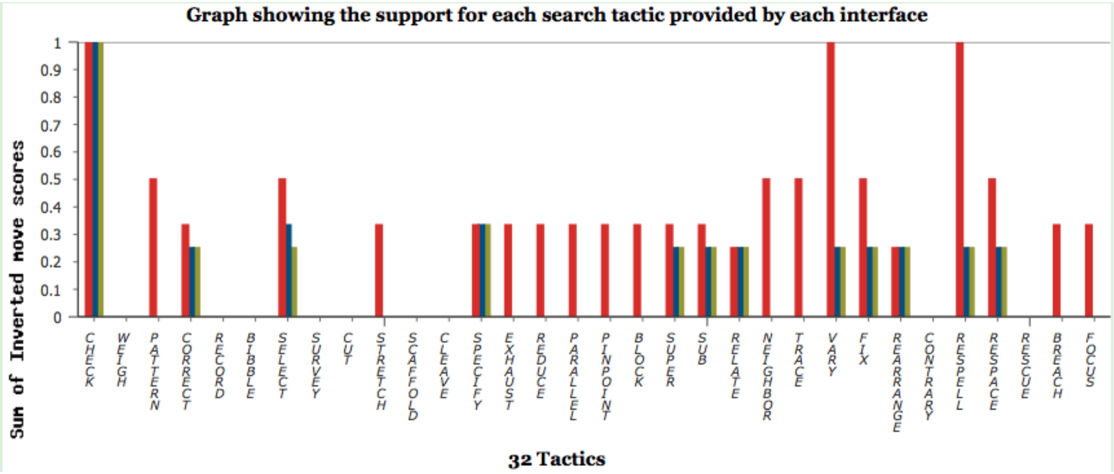


Figure B.15: Graph G2 showing the support provided for 32 search *tactics* (Bates, 1979b,a), provided by the keyword search of the old and new designs for the RYA website compared to Google, discussed in Section 4.3, where taller bars represent stronger support.

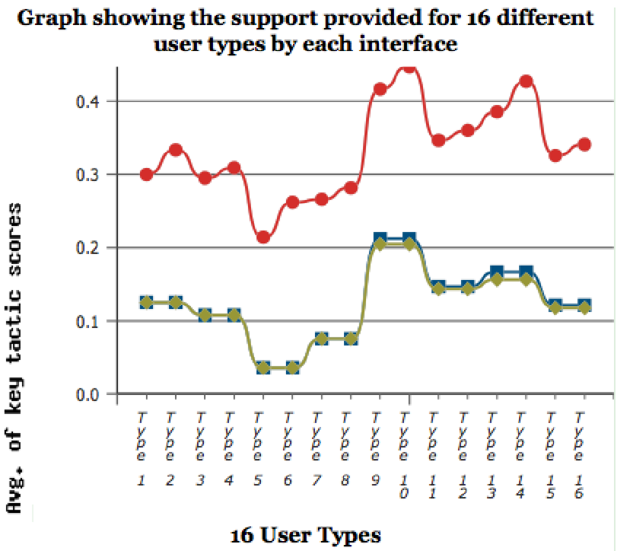


Figure B.16: Graph G3, showing the support provided for 16 searcher profiles (Belkin et al., 1993), by the keyword search of the old and new designs for the RYA website compared to Google, discussed in Section 4.3, where peaks represent stronger support.

Appendix C

Step-by-Step Guide to Using the Sii Website

The aim of this appendix is to provide a detailed step-by-step guide to applying the framework: from choosing something to analyse to assessing the benefits of re-design ideas. The Sii website[1] will be referenced appropriately throughout, as well as providing guiding screen-shots. The specific example discussed here is shown can also be found on the Sii website[2]. The high-level overview, as shown in the **Getting Started** guide on Sii's front page, shown in Figure C.1.

The focus of this guide is on these last two point, assuming that users register and begin a new project, describing carefully the steps and sub-steps involved in using the online tool. Before beginning, however, it is important to decide why an analysis is being performed in the first page. There are several usage scenarios, as described in detail in Section 3.2.4, which include, comparing design options, understanding prior art, and analysing potential design revisions. A *New Comparison* can then be started online, including a description of why the project has been performed.

The Tabulator has been chosen for the example carried throughout this guide. The Tabulator is a Semantic Web browser, designed to embody the interactions that have been envisioned by the future of the World Wide Web, by the Webs own creator: Tim Berners-Lee. Unlike the current web, made up primarily of web pages, the Semantic Web has been designed to be a Web of data that can be queried by either humans or computers in any combination as described by the data itself. That is, if the data on the Semantic Web has specific attributes and relationships, then either, and any number of them, can be used as constraints during search. While powerful, some users have reported that it is hard to use to find desired information. The reasons for the chosen example, therefore, listed below:

[1] http://mspace.fm/sii
[2] http://mspace.fm/sii/project.php?pid=15

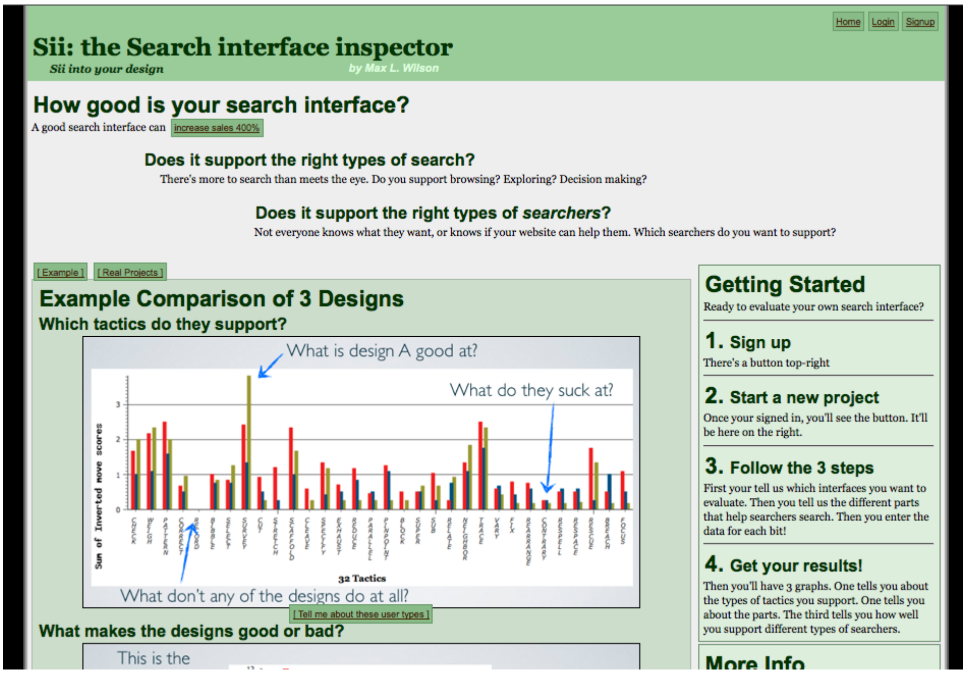


FIGURE C.1: The Sii website front page provides an example of how designs can be compared, some real examples, a getting started guide, and the ability to log in or register at the top right.

- To discover why some users have found it difficult to use
- To understanding the tabulator as prior art
- The tabulator provides a rich example for the guide

C.1 Applying the Framework

Figure C.2 shows the creation of a new project on the Sii website, including a motivation description in line with the reasons listed above.

After filling in this information, the user is taken to a project overview page, shown in Figure C.3, where the 3 steps can begin.

C.1.1 Step 1: Choose the designs

The first step of the frameworks application process is to identify the browsers being evaluated. In this example application, a single search interface is being evaluated: The Tabulator (Berners-Lee et al., 2006). Selecting the *Add Interface* button in Fiugre C.3 takes a Sii user to Figure C.4, where the details of an interface can be added. A version should be included or clarity, especially when comparing multiple versions or design

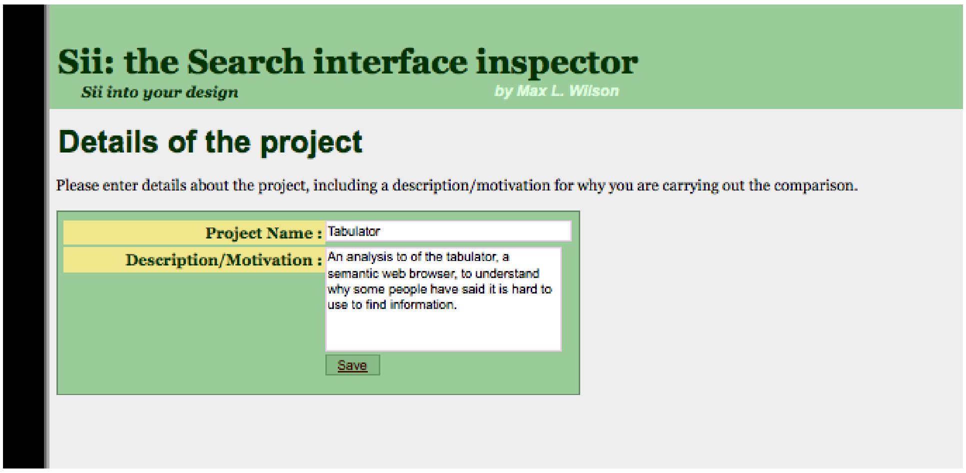


FIGURE C.2: Users should provide a title and motivation for a project, so that it is clear to others why the analysis is being performed.

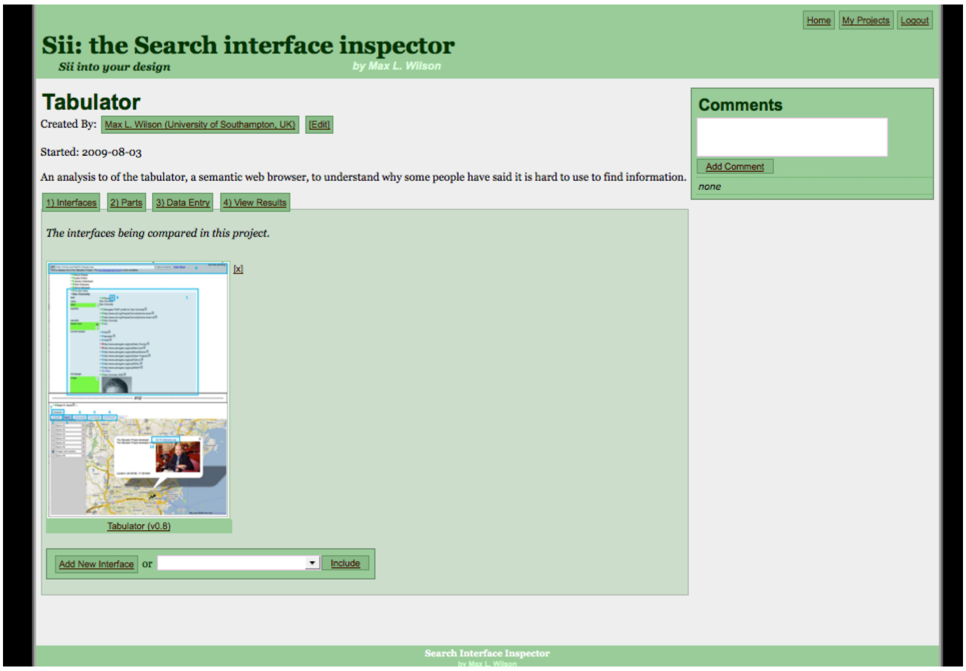


FIGURE C.3: The project overview page includes details about who created the project, and when. The three steps of applying the framework are listed in tabs underneath. The fourth tab shows the results.

revisions of a system. An image of the user interface should also be uploaded. This image is shown separately here in Figure C.5.

As stated above, the Tabulator is a semantic web browser, and so the aim of search on such a web is not to find pages that contain desired information, but to find the information itself. As a result of this difference, the Tabulator has also focused its design upon the ability to both find and analyze information. Consequently, the interface has two main halves, the search half, and below that, the analysis half. Step 2 breaks down the design into more detail.

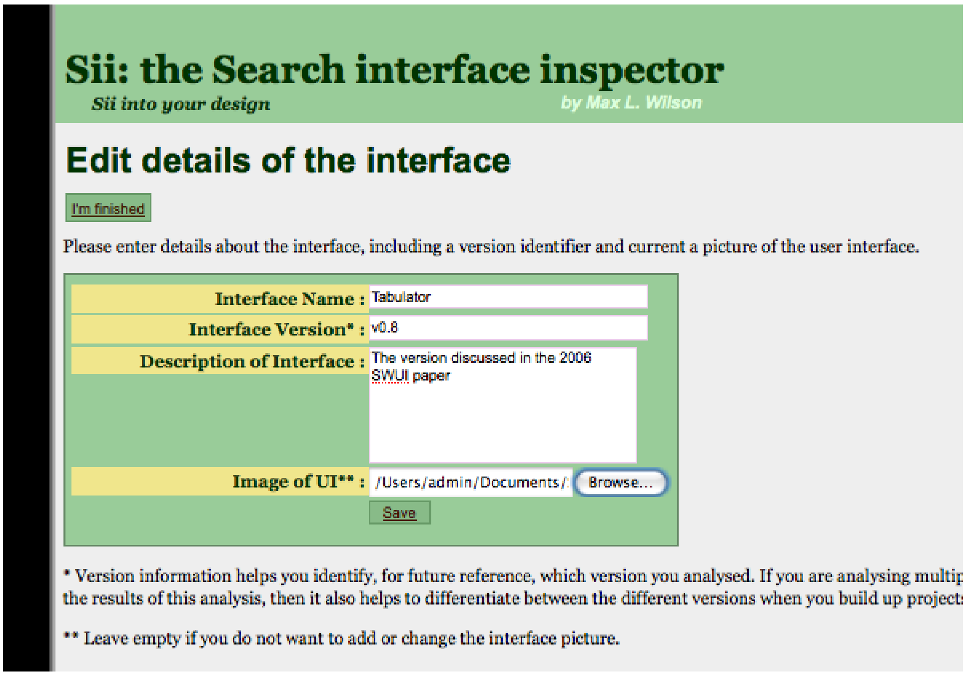


FIGURE C.4: The form for entering a new user interface into a Sii project.

C.1.2 Step 2: Identify the interface *parts* (or features)

There are 8 main features of the Tabulator interface, and two less obvious feature, which have been highlighted in Figure C.5 and inputted into the Sii website in Figure C.6. Each feature should be carefully described and users may find it helpful to think carefully about how each feature is distinct from the other features in the interface. These features are described below.

The foremost feature of the interface is the tree-based explorer (#2 in Figure C.6). Using this explorer, the user can expand any one of the root nodes initially listed to see all of the attribute types associated with it, and one or more of their values (long lists are cut off and replaced with a more button). The user can continue to navigate in this way as long as the values reached by expansion have further attributes to expand. As well as exploring in this way to find specific items of information, the user can also define a pattern and request, using the Find All button, to see all such values. To assert such a pattern, the user can select the attributes and/or values in the explorer, so that they are highlighted in green. Alt-select allows the user to select multiple attributes or values for more complicated examples, as shown in Figure C.5.

For example, a user might expand a developer team node to see all of its attributes, such as its office location and its developers, and expand the details of one team member, and highlight: the name, date of birth, current living location and picture. Pressing Find All will find these details for all the members of the team and pass them to the analysis features, described below. If, however, there is a team manager with these same details, he will also be found, as the user did not highlight developer as a constraint. The user

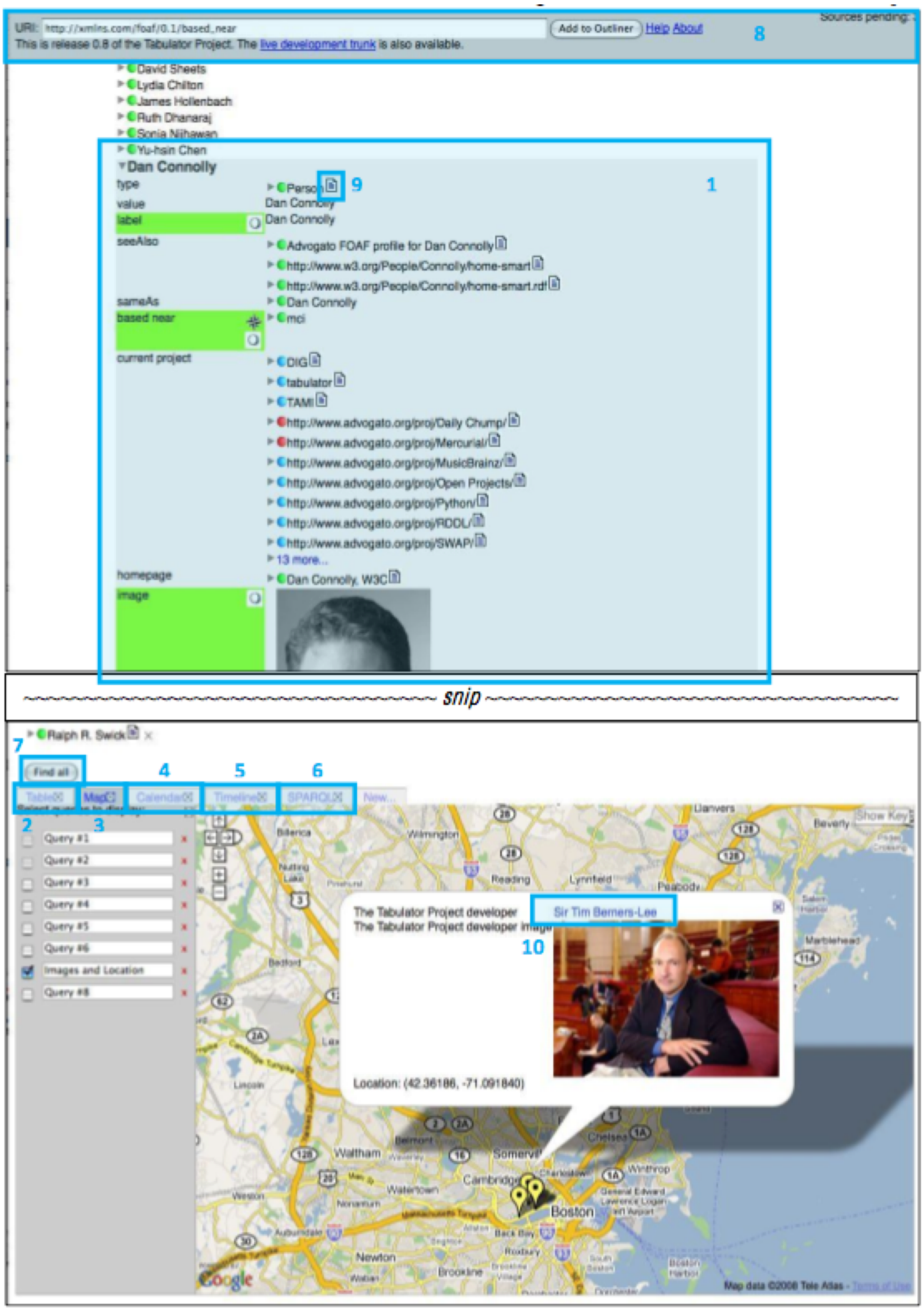


FIGURE C.5: The Tabulator browser interface, with features highlighted in blue.

FIGURE C.6: The features of each design should be listed together. New features can be added using the form at the bottom.

may add this constraint and select Find All to pass the new findings to the analysis modules, as a new result set. Further, the user may decide that they want to see the whole team, regardless if they are missing either their date of birth, or home town, and may mark them as optional with the radio button seen within the green highlight.

There are 5 analysis modules available (#4-8), that make up 5 separate features: the table view, the map view, the calendar view, the timeline view, and the SPARQL code view, which allows the user to directly edit a query in the SPARQL[3] language used to retrieve from the Semantic Web. The Find All button passes sets of results to these views to be displayed. In the team example above, the table view would show four columns, with the team members names, dates of birth, locations and pictures. As the query contains a location field, these can be displayed on the map view. Multiple result sets can be shown on the map view at once if required. Similarly, as the team member query above has a date field, the user can show their dates of birth in either the calendar or the timeline view, where result sets can be combined if required. The SPARQL viewer provides a query by example interface, allowing the user to edit the queries that produced existing result sets, and use them to create new queries, and thus new results sets.

[3] http://www.w3.org/TR/rdf-sparql-query/

The first unobvious feature of the interface is, in fact, the Find All button (#10), which serves to create results sets from the patterns defined in the explorer, and pass them to the analysis modules. This has been identified as a separate function as it is not required to explore or to analyse, but is required to move from exploring to analysing.

Another noticeable feature of the interface is the URI bar that is permanently visible at the top of the screen (#1). Primarily, the URI bar is used to display the complete URI of the last item selected within the Explorer. This allows the user to both check the provenance of an item selected, and copy and save it if necessary. The URI Bar may also be used to add certain parts of the Semantic Web to the browser, as a new root node on the interface. This can be achieved by pasting a URI into the URI Bar and pressing Add to Outliner, where Outliner is the name used for the explorer.

The penultimate feature to identify in the Tabulator is the RDF Popup button (#3). This allows the user to view the original source data, in the RDF format[4], of something found in the explorer. The final feature of the Tabulator to identify is that any item found in the analysis modules may be loaded as a new starting node in the explorer, by double clicking on it (#9). So in the team member example, the user may wish to start exploring again from one particular member, or one particular location or date.

C.1.3 Step 3: Data Entry

The final step of applying the framework requires the evaluator to count how many actions it takes for a user to carry out each known tactic, if the tactic is achievable with that feature. Tab 3, as shown in Figure C.7, provides the user with the opportunity to add data for every feature of each interface, should that interface include the feature. In this example, where only one interface is being evaluated, the list contains only features in the Tabulator interface. At each data entry point, the evaluator has to count how many moves it takes to achieve each of the 32 tactics with that feature of that interface.

Moves, as according to Bates (1979b,a), are segmented mental or physical actions. Familiar examples in keyword search would be: choosing a search term (mental), entering a search term (physical), pressing search (physical), scanning results (mental), choosing a result (mental), and opening it (physical). In the simple way just demonstrated, we seek to count the moves required to achieve each tactic with each of the 9 interface features. Repeated and circumstantial moves, such as scrolling, are not counted, as we cannot guarantee that they will be needed. It may not be possible to achieve some tactics with a feature of an interface, which should be left at 0.

Short of describing all 320 count estimations, they have been listed in a single table shown in Table C.9. Specific examples of how five of these counts have been calculated are listed below.

[4]http://www.w3.org/RDF/

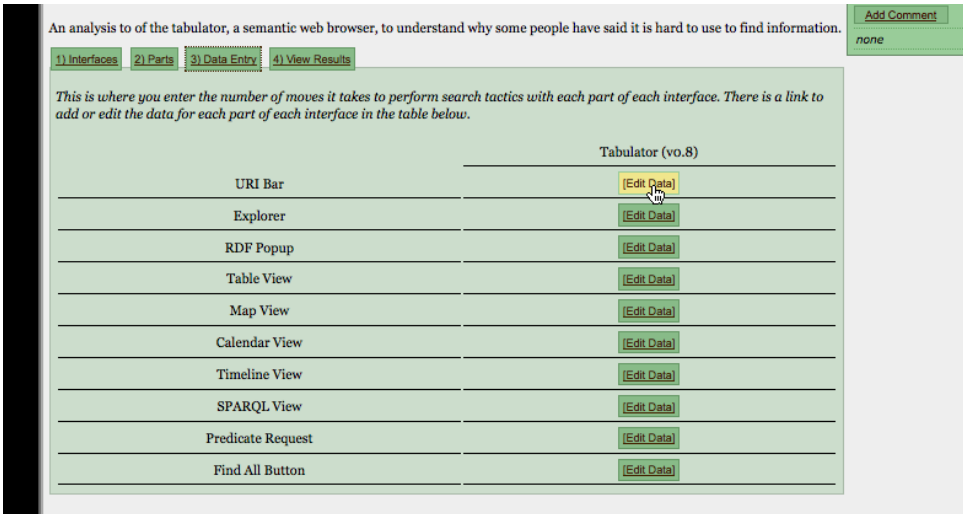


Figure C.7: Data should be entered for every feature contained in each interface. An *Edit Data* button is available for each one point.

Figure C.8: At each data entry point, the evaluator is asked to count how many moves it takes to achieve each tactic with that feature of that interface.

Tactics

Tabulator	(Totals)	CHEC	WEIG	PATTE	CORR	RECO	BIBBI	SELE	SURV	CUT	STRE	SCAF	CLEA	SPEC	EXHA	REDU	PARA	PINP	BLOC	SUPE	SUB	RELA	NEIGI	TRAC	VARY	FIX	REAR
URI Bar	2.8702381		4	6	6	7					8	6		5						4	4	6			6	6	
Explorer	6.99761905	3	4	4	2			3	2		6	7		2	3	3	5	3		3	3	3	4	2		6	7
RDF Popup	1.70833333			4		3	4		4		8	8												4			
Table View	3	3	6	6	6			3			6	6											2	2			
Map View	5.2	5	8	8	8			3		2	8	8			2	2			2		2	2	3	3			
Calendar Viev	4.7	5	8	8	8			3			8	8			2	2			2		2	2	3	3			
Timeline View	4.7	5	8	8	8			3			8	8			2	2			2		2	2	3	3			
SPARQL View	2.71803752	5	7	7	9			9		9	11	9		9	9	9	9	9		9	9	9			9	9	9
Predicate Req	1.75			4			4				2													4			
Find All	0.5					2																					

Figure C.9: The data entered into the Sii website for this Tabulator example, shown together in one table.

SPECIFY with the Explorer: 2 moves. Perhaps the most simple tactic-feature combination is to specify an information need with the explorer. In the simplest scenario, with circumstantial conditions such as appropriate starting point and a single constraint, only two moves are required: choose an attribute to expand (mental), and expand it (physical).

CHECK with the Explorer: 3 moves. The CHECK tactic refers to be the user correlating their current search status against their original goal. The user is able to check that their search is still on a productive path with three moves: review their selections in the explorer (characterized by green highlights or expanded lists), recount their intentions, compare the two. In this case, as a visualization of the users previous selections is produced as they explore, no physical moves are required to carry out this monitoring tactic, only the identification of two comparable items, and their comparison.

CHECK with the Table View: 3 moves. Like with the explorer, the user is able to CHECK their current state against their search goal with the Table view in 3 moves. The attributes selected to produce the result sets are clearly listed as table headers, and so the user can: view the table headers, recount their intention, and compare the two. Again, these are all mental.

CHECK with the Map View: 5 moves. Unlike the Table View and Explorer, a physical action is required before the user can check the direction of their search. As the user can only see the attributes of the result set in the information popup of a selected item on the map, the user is required to: choose an item on the map (mental), select it (physical), view the information popup (mental), recount their intentions (mental), and compare the two (mental).

RECORD with the Find All button: 2 moves. Uniquely in the interface, the Find All feature serves only one tactic: to record ones search for future reference. When the user presses the Find All button, any matching results are stored as a result set that can be used by any of the analysis modules. It does not, itself, display any results, or allow the user to affect their search in anyway. Simply, the moves required are: identify the location of the Find All button (mental) and press it (physical).

C.2 Analysing the Framework's Results

Tab four of the project page, which can be seen in several of the figures above, brings the evaluator to the results page, as shown in Figure C.10.

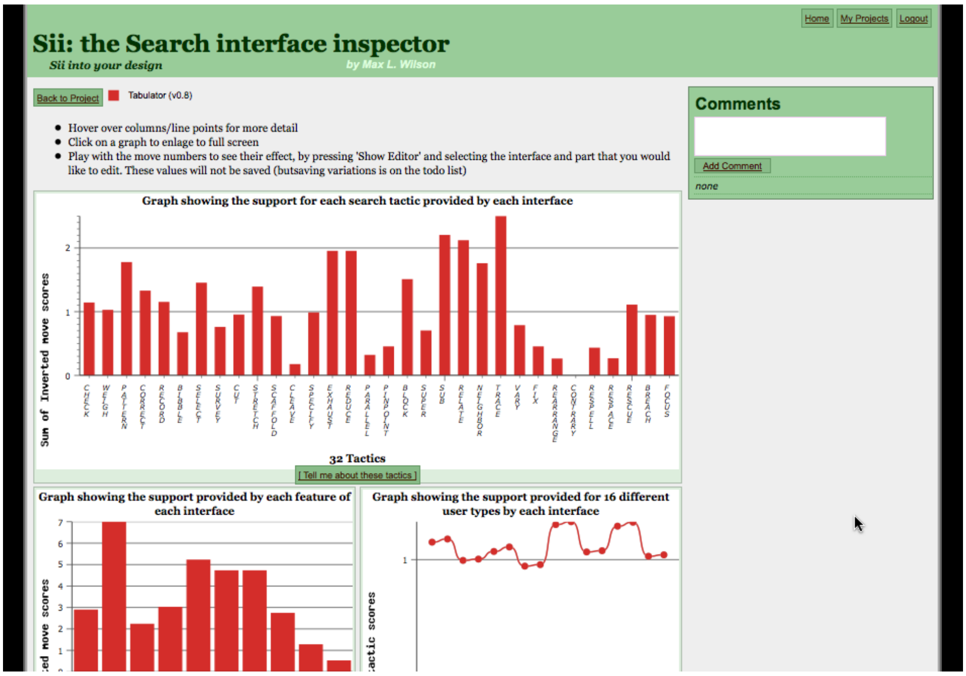


FIGURE C.10: Three graphs are provided to help analyse the interface design. Each can be viewed full screen by clicking on the surface of the chosen graph.

C.2.1 Graph G1: Interface Parts

Graph G1, shown in Figure C.11, conveys the level of support during search that is being provided by each of the features in the interface, as defined in Step 2. There are several questions to ask of this graph:

1. which ones are tall and why?
2. which ones are short and why?
3. if comparing - which design has the tallest for each and why?
4. if comparing - are there any gaps?

This graph alone both confirms some expectations and reveals some interesting insights. First, it is not surprising, perhaps, that the Explorer provides the broadest amount of support for search, compared to all the other features within the Tabulator. Second, it is probably not surprising that the different visualizations at the bottom of the interface make up the subsequently tall bars within the graph, as these provide the means to analyze the results further.

One perhaps surprising result is that, while the table view may provide the most often used representation for analysis, the map, calendar, and timeline views provide more support for search. This prompts the question, which has probably not been asked as of

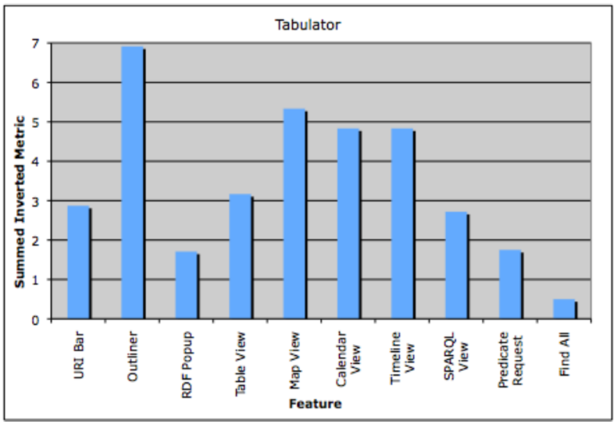


FIGURE C.11: Graph G1 showing the support provided by the features of the Tabulator (Berners-Lee et al., 2006), where taller bars represent stronger support.

yet: what about their design is different to the table view? Consulting the input table (Table C.9) in more detail reveals that compared to the table view, the other views are interactive. With the map, for example, the user is able to zoom in on specific groups of results, thus reducing the number of results found. There is currently no means within the table view to manipulate the results and so the subsequent question is, therefore, how could the table view be altered to permit further investigation.

Another perhaps surprising result is the support for search provided by the URI bar that is persistent at the top of the screen. Investigating the input table, shown in Table C.9, reveals that, as this persistently shows the URI of the last item clicked on, that it can be used for a number of monitoring tactics. As it can also be used as an input to control the main explorer, the URI Bar can also be used for tactics such as expanding, narrowing, and restarting ones search.

Finally, although it appears only to serve as a means to fill the analysis views below, the Find All button, in of itself, supports the tactic of recording ones search. If it merely populated the views, rather than creating query objects that can be compared or combined, then it would support any particular tactic at all.

Although no direct comparison is being performed here, the evaluator may wish to look at other popular search interfaces to see what features they have, and think about how they might add to the tabulator. Many other of the designs evaluated in this thesis, for example, include a keyword search box.

C.2.2 Graph G2: Search Tactics

Graph G2, shown in Figure C.12, conveys the opposing view to Graph G1, by representing the amount of support available for each search tactic. With Graph G2, therefore, we can identify certain tactics where the Tabulator poorly supports users during search. The questions to ask when viewing this graph are:

1. which bars are tall and why?

2. which bars are short or missing?

3. if comparing, which designs do better for each tactic and why?

From first glance, it appears that the Tabulator provides quite a broad range of tactics, but comparison with results from another study (Section 4.1.4, shown in Figure C.13, shows that it actually has fairly low scores across the board. The purpose of Figure C.13 is not to compare the two browsers, as they have different aims, but to demonstrate that equal support for many tactics is different from supporting them all well.

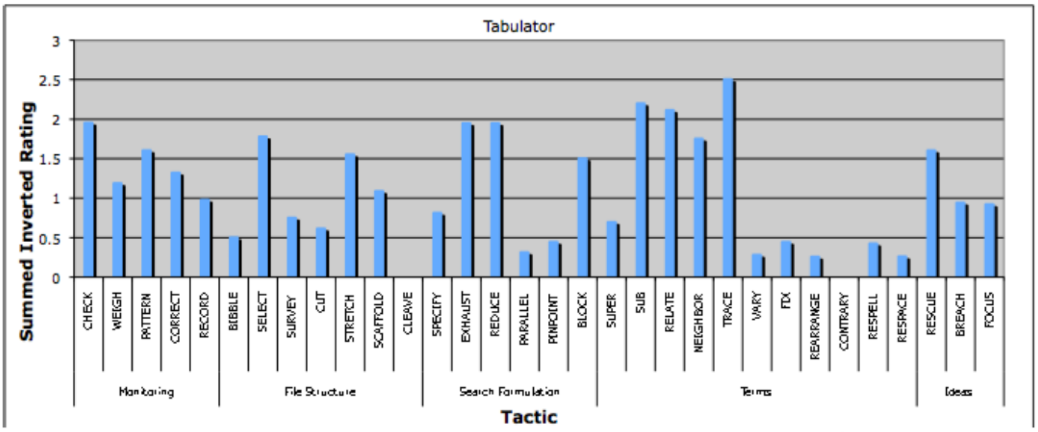


FIGURE C.12: Graph G2 showing the support provided for 32 search *tactics* (Bates, 1979b,a) by the Tabulator (Berners-Lee et al., 2006), where taller bars represent stronger support.

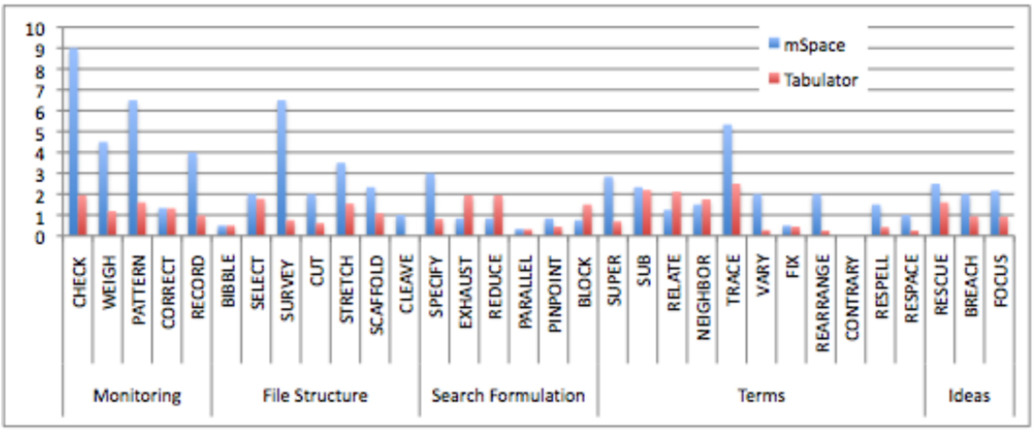


FIGURE C.13: Graph G2 (shown in Figure C.12), combined with the results from a previous analysis of mSpace (shown in Section 4.1.4), where taller bars represent stronger support.

From Figure C.12, we can see that there are two tactics that are entirely unsupported; although results from other analyses show that these are often the hardest to support. CONTRARY, for example, is to find the opposite of something, which is inherently different from showing everything but something (BLOCK). While TRACE, consulting results to find new search constraints, is often well supported, the tabulator supports this better than actually defining or altering ones search constraints. Consulting the input table reveals that this is due to the many ways of visualizing results, but that the only way to specify ones searches is through the single explorer interface.

One key tactic is to SPECIFY ones constraints, and we can see that it has much more support, compared to some other tactics relating to refining search constraints. This supports the opinion held by many that the Tabulator can be hard for a user to specify what they would like to find with the Tabulator interface.

It is also clear in the graph, that the first half of the term tactics receive much more support than those in the latter half. This is shows that it is easier to expand and narrow upon ones search than it is to specify variations within them. That is, a user is restricted to either specifying a specific value of a particular attribute, or that they would like any value of a particular attribute. It is difficult using the specify-then-analyze model of the Tabulator to explore variations in either phase, as the results of a users actions are so distantly removed from the actions themselves.

C.2.3 Graph G3: User Types

Graph G3 is designed to convey how different types of users are supported. The 16 user types are made up of four dimensions of two options, as displayed in Table 2. Like the pattern created by the pairs of options in the table, Graph G3, shown in Figure C.14 also has patterns. Further descriptions of these dimensions can be found in previous work (Chapter 2) and from the originally published model (Belkin et al., 1993). These four dimensions lead to four interrogation angles, discussed in turn.

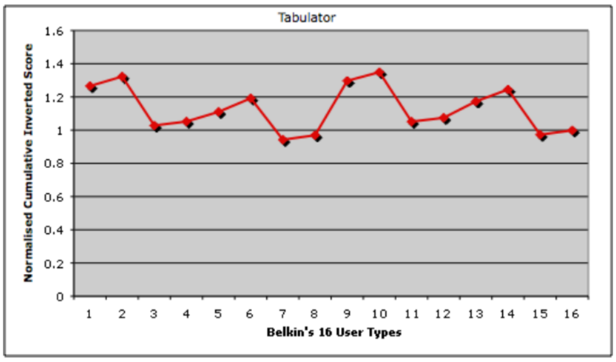


FIGURE C.14: Graph G3 showing the support provided for 16 searcher profiles (Belkin et al., 1993), read in conjunction with Table 2.1, where peaks represent stronger support.

Method of Search. The first and the second half of the graph, for example, are almost identical, indicating that the Tabulator is just as supportive for people who are scanning or searching, where the latter is characterized by searching for a known item. The second half of the graph is slightly higher, representing slightly better support for those who are searching.

Goal of Search. There is also a clear pattern across the different quarters of the graph, where the odd quarters are noticeably higher than the even quarters. Unlike many browsers, this means that users who are intending to learn more generally about a topic are better supported than those who are specifically aiming to retrieve a certain piece of information.

Mode of Search. The most prominent difference seen is between the odd and even eighths of the graph. This drop indicates that it is significantly harder to use for people who can specify exactly what they need, than it is for people who are likely to recognize the information they need when they see it. This emphasizes one of the results shown in Graph G2 and matches the opinion held by many that it is actually hard to use the Tabulator to find specific information, and that users are almost entirely dependant on what is presented to them as they explore. Ultimately, the user is required to begin at varying starting points, and to seek the information they can only navigate through links and associations. Most existing web browsers provide keyword search paradigms to search for and jump directly to the information they need, and allow navigation from there.

Resource Being Sought. The final pattern seen is between the odd and even sixteenths of the graph, which are slightly higher for the latter part of each pair. This indicates that it is slightly easier to find metadata than it is to find particular information objects. This is perhaps not surprising for a browser of the data-web, which promotes exploration of inter-object associations.

C.3 Producing Design Recommendations

There are three ways, one for each graph, to consider design changes. Each of these methods will be further enabled by the additional functionality planned in the Developing research agenda described in Chapter 6.

Improving weak features.
First, the designer can look at the short bars, or even gaps, in Graph G1, and decide whether these features could be redesigned to a) require fewer steps, or b) to support a larger number of tactics. Both of these options should increase the support provided by a feature. In this example, one of the key areas of focus should be in improving the users ability to Specify what they are looking for. Further, the designer may want to consider how to extend the design with a keyword search feature. Finally, one of the clearest design recommendations produced by the comparison of interface features, is that the Table view could be easily extended to be interactive, perhaps allowing users to filter, reorder, or remove columns.

Increasing support for under-supported tactics.
Participants in the pilot studies above regularly focused on under supported tactics, to consider which, if any, features could be altered in order to better support the feature.

Graph G2 shows some individual tactics that require better support. Primarily, these poorly supported tactics include: SPECIFY, FIX, REARRANGE and PARRALLEL. Each of these equate to the users ability to determine and vary their requirements easily. Currently, the primary method of expressing search constraints with the tabulator is through navigating within the explorer, which provides very little means for the user to quickly vary their constraints, as they must re-navigate to do so. Consequently, the tabulator should explore alternatives to navigation, as opposed to alternative navigation methods.

Increasing support for under-supported users.
Graph G3 is created from Graph G2, using a mapping that approximates the needs of each type of user to the tactics they may primarily need to apply. Consequently, evaluators might want to choose particular user profiles in G3, and prioritise tactics in G2. Notably, from the analysis in this appendix, the Tabulator may want to try and support the ability to find previously known results (Select in the Mode dimension).

C.3.1 Analysing Design Changes

While reviewing the findings can provide both clear design recommendations and general areas for improvement, one key challenge for designers is in knowing whether any changes will solve them. Another advantage of this framework, however, is that because paper prototypes can be just as easily evaluated as implemented systems, speculative redesigns can included within the already performed evaluation.

There are two methods for evaluating design changes. First, an editor, shown in Figure C.15, is made available under the graphs. Evaluators can make changes for each feature of each interface with this editor and see the changes reflected in the graphs. The evaluator may think of ways to reduce the number of steps involved in using a feature and, by changing the values in the editor, see the changes to the graphs. Should the evaluator wish to directly compare these changes with the original, or see the effect of adding an entirely new feature, they can add a new interface into the same comparison project. The evaluator could add the user interface to the interface list in tab 1 (Figure C.3), with a new version name. Following this, the evaluator can add the new feature to the feature list, enter the necessary data, and then see both designs appear in the all three graphs.

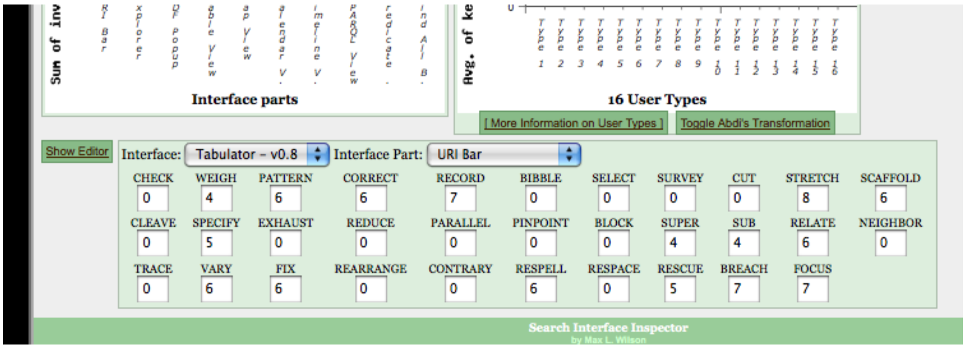


FIGURE C.15: Evaluators can make temporary changes to the data entered in the three steps above using the editor under the graphs. These changes are reflected in the graphs, but not made permanent, so that the original analysis remains in tact.

C.3.2 Further Advice

As with all inspection-based evaluation techniques, the decisions, and consequently the data entered, are subject to some interpretation, especially when estimating mental user actions. This subjectivity is inherent in inspection methods, however, as the evaluators are estimating user behavior, rather than to recording it. As with other methods, like the Cognitive Walkthroughs, judgments could be made and discussed by multiple evaluators. This is especially important when evaluators are less familiar with an interface, like, for example, when the interface is new and has been designed by a different party.

While striving to record accurate sets of moves, internal consistency is also important, as different interfaces, or even different interface features, become less comparable if evaluator decisions vary over time. Practice, and pilot applications can help establish internal consistency, and statistical methods such as Cohens Kappa (Cohen et al., 1960), or Fleiss Kappa (Fleiss et al., 1971) can be used to assess the inter-rater reliability of multiple judges if splitting up analyses, as opposed to jointly discussing them necessary. In the simplest instance, re-checking a small number of early data entries, after completing an analysis, can help improve the reliability of that particular evaluation.

Bibliography

Allen, T. J. (1977). *Managing the Flow of Technology.* MIT Press, Cambridge, MA.

Amershi, S. and Morris, M. R. (2008). Cosearch: a system for co-located collaborative web search. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1647–1656. ACM Press.

Back, J. and Oppenheim, C. (2001). A model of cognitive load for IR: implications for user relevance feedback interaction. *Information Research*, 6(2):Workshop Paper 2.

Baddeley, A. (2000). The episodic buffer: a new component of working memory? *Trends in cognitive sciences*, 4(11):417–423.

Baddeley, A. D. and Hitch, G. (1974). *Working Memory*, pages 47–89. Academic Press.

Bartneck, C. and Hu, J. (2009). Scientometric analysis of the chi proceedings. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 699–708, New York, NY, USA. ACM.

Bastien, J. and Scapin, D. (1993). Ergonomic criteria for the evaluation of human-computer interfaces. Technical Report 156, Institut National de Recherce en Informatique et en Automatique (INRIA).

Bates, M. (1989). The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–424.

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

Bates, M. J. (1979b). Information search tactics. *Journal of the American Society for Information Science*, 30(4):205–214.

Bates, M. J. (1990). Where should the person stop and the information search interface start? *Information Processing and Management*, 26(5):575–591.

Bates, M. J. (2002). The cascade of interactions in the digital library interface. *Information Processing and Management*, 38(3):381–400.

Beitzel, S. M., Jensen, E. C., Chowdhury, A., Grossman, D., and Frieder, O. (2004). 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. ACM Press.

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

Belkin, N. J., Marchetti, P. G., and Cool, C. (1993). Braque: design of an interface to support user interaction in information retrieval. *Information Processing and Management*, 29(3):325–344.

Belkin, N. J., Oddy, R. N., and Brooks, H. M. (1997). *Ask for information retrieval: part I.: background and theory*, pages 299–304. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Berners-Lee, T., Chen, Y., Chilton, L., Connolly, D., Dhanaraj, R., Hollenbach, J., Lerer, A., and Sheets, D. (2006). Tabulator: Exploring and Analyzing linked data on the Semantic Web. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop*.

Bertini, E., Catarci, T., Dix, A., Gabrielli, S., Kimani, S., and Santucci, G. (2009). Appropriating heuristic evaluation for mobile computing. *International Journal of Mobile Human Computer Interaction*, 1:20–41.

Blandford, A., Hyde, J., Green, T., and Connell, I. (2008). Scoping analytical usability evaluation methods: a case study. *Human-Computer Interaction*, 23(3):278–327.

Boehm, B. (1986). A spiral model of software development and enhancement. *SIGSOFT Software Engineering Notes*, 11(4):14–24.

Borlund, P. (2003). The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3):Paper 152.

Bossert, J. L. (1991). *Quality function deployment: a practitioner's approach*. CRC Press.

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

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

Buckland, M. K., Norgard, B. A., and Plaunt, C. (1992). Design for an adaptive library catalog. In *Networks, Telecommunications, and the Networked Information Revolution: Proceedings of the ASIS Mid-Year Meeting*, pages 165–171. American Society for Information Science.

BWB General Directive 250 (1992). Software Development Standard for the German Federal Armed Forces, V-Model, Software Lifecycle Process Model. Technical report, Bundesamt fur Wehrtechnik und Beschaffung.

Byström, K. and Hansen, P. (2002). *Work tasks as unit for analysis in information seeking and retrieval studies*, pages 239–251. Greenwood Village, CO, Libraries Unlimited.

Capra, R. and Marchionini, G. (2008). The relation browser tool for faceted exploratory search. In *JCDL'08: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 420–420. ACM New York, NY, USA.

Capra, R., Marchionini, G., Oh, J., Stutzman, F., and Zhang, Y. (2007). Effects of structure and interaction style on distinct search tasks. In *JCDL'07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 442–451. ACM Press New York, NY, USA.

Card, S. K., Moran, T. P., and Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7):396–410.

Card, S. K., Newell, A., and Moran, T. P. (1983). *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA.

Carroll, J. M. (1995). *Scenario-based design: envisioning work and technology in system development*. John Wiley & Sons, Inc. New York, NY, USA.

Chandler, P. and Sweller, J. Cognitive load while learning to use a computer program. *Applied cognitive psychology*, 10(2):151–170.

Chandler, P. and Sweller, J. (1991). Cognitive load theory and the format of instruction. *Cognition and instruction*, 8(4):293–332.

Chin, Jr., G., Kuchar, O. A., and Wolf, K. E. (2009). Exploring the analytical processes of intelligence analysts. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 11–20, New York, NY, USA. ACM.

Clarkson, E. C., Navathe, S. B., and Foley, J. D. (2009). Generalized formal models for faceted user interfaces. In *JCDL '09: Proceedings of the 2009 joint international conference on Digital libraries*, pages 125–134, New York, NY, USA. ACM.

Cockton, G. and Woolrych, A. (2001). Understanding inspection methods: Lessons from an assessment of heuristic evaluation. In *People and Computers XV: Interactions Without Frontiers: Joint Proceedings of HCI 2001 and IHM 2001*, page 171. Springer.

Cohen, J. et al. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Cook, T. and Campbell, D. (1979). *Quasi-experimentation: Design and analysis for field settings*. Rand McNally Chicago.

Cool, C. and Belkin, N. J. (2002). 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.

Cooper, A. (1999). *The inmates are running the asylum. SAMS*. Macmillan Computer Publishing.

Costabile, M. F., Garzotto, F., Matera, M., and Paolini, P. (1997). SUE: A Systematic Usability Evaluation Methodology.

Csikszentmihalyi, M. (1990). *Flow: The Psychology of Optimal Experience*. Harper & Row.

De Angeli, A., Matera, M., Costabile, M., Garzotto, F., and Paolini, P. (2000). Validating the SUE inspection technique. In *AVI'00: Proceedings of the working conference on Advanced Visual Interfaces*, pages 143–150. ACM New York, NY, USA.

De Angeli, A., Matera, M., Costabile, M., Garzotto, F., and Paolini, P. (2003). On the advantages of a systematic inspection for evaluating hypermedia usability. *International Journal of Human-Computer Interaction*, 15(3):315–335.

Dervin, B. (1992). *From the mind's eye of the user: The sense-making qualitative-quantitative methodology*, pages 61–84. Libraries Unlimited.

Diaper, D. and Stanton, N. (2003). *The handbook of task analysis for human-computer interaction*. CRC.

Diriye, A. (2008). Internal report: Search task support. Technical report, University College London Interaction Centre.

Diriye, A., Blandford, A., and Tombros, A. (2009). A polyrepresentational approach to interactive query expansion. In *JCDL '09: Proceedings of the 2009 joint international conference on Digital libraries*, pages 217–220, New York, NY, USA. ACM.

Dumais, S., Cutrell, E., and Chen, H. (2001). Optimizing search by showing results in context. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 277–284, New York, NY, USA. ACM Press.

Dumais, S. T. and Belkin, N. J. (2005). *The TREC interactive tracks: Putting the user into search*, pages 123–153. MIT Press.

Egghe, L. (2006). Theory and practise of the g-index. *Scientometrics*, 69(1):131–152.

Ellis, D. (1989). A behavioural model for information retrieval system design. *Journal of Information Science*, 15:237–245.

Fidel, R. (1985). Moves in online searching. *On-line review*, 9(1):61–74.

Fleiss, J. et al. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Følstad, A. (2007). Work-Domain Experts as Evaluators: Usability Inspection of Domain-Specific Work-Support Systems. *International Journal of Human-Computer Interaction*, 22(3):217–245.

Foster, A. (2004). A nonlinear model of information-seeking behavior. *Journal of the American Society for Information Science and Technology*, 55(3):228–237.

Fuhr, N., Govert, N., Kazai, G., and Lalmas, M. (2002). INEX: INitiative for the Evaluation of XML retrieval. In *Proceedings of the SIGIR 2002 Workshop on XML and Information Retrieval.*

Furniss, D., Blandford, A., and Curzon, P. (2008). *Towards Maturing Usability Practice in Website Design: Grounding how practitioners work to inform research requirements*, chapter 7, pages 144–167. Human-Computer Interaction Series. Springer-Verlag.

Gaffney, G. (2000). What is Card Sorting? *Information & Design.*

Gaver, B., Dunne, T., and Pacenti, E. (1999). Cultural probes. *Interactions*, 6(1):21–29.

Ghani, J. and Deshpande, S. (1994). Task characteristics and the experience of optimal flow in human-computer interaction. *The Journal of Psychology*, 128(4):381–391.

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

Golovchinsky, G., Adcock, J., Pickens, J., Qvarfordt, P., and Back, M. (2008). Cerchiamo: a collaborative exploratory search tool. In *CSCW'08: Proceedings of the ACM 2008 conference on Computer supported cooperative work (Demo Session).*

Golovchinsky, G., Qvarfordt, P., and Pickens, J. (2009). Collaborative information seeking. *Computer*, 42(3):47–51.

Gong, R. and Kieras, D. (1994). 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. ACM Press.

Gray, W. and Salzman, M. (1998). Damaged merchandise? A review of experiments that compare usability evaluation methods. *Human-Computer Interaction*, 13(3):203–261.

Gray, W. D., John, B. E., and Atwood, M. E. (1992). 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. ACM Press.

Greenfield, A. (2006). *Everyware: The dawning age of ubiquitous computing.* New Riders.

Grudin, J. (1989). The case against user interface consistency. *Communications of the ACM*, 32(10):1164–1173.

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

Hansen, P. and Järvelin, K. (2005). Collaborative Information Retrieval in an information-intensive domain. *Information Processing and Management*, 41(5):1101–1119.

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

Hart, S. and Staveland, L. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Human mental workload*, 1:139–183.

Hearst, M. A. (2000). Next generation web search: Setting our sites. *IEEE Data Engineering Bulletin: Special Issue on Next Generation Web Search*, 23(3):38–48.

Hearst, M. A. (2006). Clustering versus faceted categories for information exploration. *Communications of the ACM*, 49(4):59–61.

Hearst, M. A. (2009). *Search User Interfaces.* Cambridge University Press.

Hertzum, M. and Jacobsen, N. (2003). The evaluator effect: A chilling fact about usability evaluation methods. *International Journal of Human-Computer Interaction*, 15(1):183–204.

Hirsch, J. (2005). An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences*, 102(46):16569–16572.

Hirshfield, L. M., Solovey, E. T., Girouard, A., Kebinger, J., Jacob, R. J., Sassaroli, A., and Fantini, S. (2009). Brain measurement for usability testing and adaptive interfaces: an example of uncovering syntactic workload with functional near infrared spectroscopy. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 2185–2194, New York, NY, USA. ACM.

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

Hsieh-Yee, I. (1993). 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.

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

Hu, P. J. H., Ma, P. C., and Chau, P. Y. K. (1999). Evaluation of user interface designs for information retrieval systems: a computer-based experiment. *Decision Support Systems*, 27(1-2):125–143.

Huvila, I. and Widen-Wulff, G. (2006). 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.

Huynh, D. F., Miller, R., and Karger, D. (2006). Enabling web browsers to augment web sites' filtering and sorting functionalities. In *ACM Symposium on User Interface Software and Technology*.

Ingwersen, P. (1992). *Information retrieval interaction*. Taylor Graham Publishing, London, UK, UK.

Ingwersen, P. and Järvelin, K. (2005). *The Turn: Integration of Information Seeking and Retrieval in Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Järvelin, K. and Ingwersen, P. (2004). Information seeking research needs extension towards tasks and technology. *Information Research*, 10(1):Paper 212.

Jeffries, R., Miller, J., Wharton, C., and Uyeda, K. (1991). User interface evaluation in the real world: a comparison of four techniques. In *CHI'91 Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, pages 119–124. ACM New York, NY, USA.

John, B., Blackmon, M. H., Polson, P. G., and Teo, L. (2009). Rapid theory prototyping: An example of an aviation task. In *Proceedings of the Human Factors and Ergonomics Society 53rd Annual Meeting*.

John, B., Vera, A., Matessa, M., Freed, M., and Remington, R. (2002). Automating CPM-GOMS. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 147–154, New York, NY, USA. ACM.

John, B. E. (1990). Extensions of GOMS analyses to expert performance requiring perception of dynamic visual and auditory information. *CHI'90: Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, pages 107–116.

John, B. E. and Kieras, D. E. (1996). Using GOMS for user interface design and evaluation: which technique? *ACM Transactions on Computer-Human Interaction*, 3(4):287–319.

John, B. E. and Newell, A. (1987). Predicting the time to recall computer command abbreviations. In *CHI '87: Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*, pages 33–40, New York, NY, USA. ACM.

John, B. E., Rosenbloom, P. S., and Newell, A. (1985). A theory of stimulus-response compatibility applied to human-computer interaction. In *CHI '85: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 213–219, New York, NY, USA. ACM.

John, R. I. and Mooney, G. J. (2001). Fuzzy user modelling for information retrieval on the world wide web. *Knowledge and Information Systems*, 3:81–95.

Jones, M., Buchanan, G., Harper, R., and Xech, P.-L. (2007). Questions not answers: a novel mobile search technique. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 155–158, New York, NY, USA. ACM.

Kammerer, Y., Nairn, R., Pirolli, P., and Chi, E. (2009). Signpost from the masses: learning effects in an exploratory social tag search browser. In *CHI'09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 625–634. ACM New York, NY, USA.

Karlson, A. K., Robertson, G. G., Robbins, D. C., Czerwinski, M. P., and Smith, G. R. (2006). Fathumb: a facet-based interface for mobile search. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 711–720, New York, NY, USA. ACM.

Karvonen, K. (2000). The beauty of simplicity. In *CUU '00: Proceedings on the 2000 conference on Universal Usability*, pages 85–90, New York, NY, USA. ACM.

Kelly, D., Dumais, S., and Pedersen, J. O. (2009). Evaluation challenges and directions for information-seeking support systems. *Computer*, 42(3):60–66.

Kieras, D. (2001). Using the Keystroke-Level Model to Estimate Execution Times.

Kim, J. (2009). Describing and predicting information-seeking behavior on the web. *Journal of the American Society for Information Science and Technology*, 60(4):679–693.

Kirsh, D. (2009). How interaction improves sense making. In *CHI 2009 Sensemaking Workshop.*

Koenemann, J. and Belkin, N. J. (1996). 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. ACM Press.

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

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

Kules, B., Kustanowitz, J., and Shneiderman, B. (2006). Categorizing web search results into meaningful and stable categories using fast-feature techniques. In *JCDL'06: Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pages 210–219. ACM New York, NY, USA.

Law, E. L.-C., Scapin, D., Cockton, G., Springett, M., Stary, C., and Winckler, M., editors (2009). *COST294-MAUSE Closing Conference Proceedings : Maturation of Usability Evaluation Methods: Retrospect and Prospect.* IRIT Press, Toulouse, France.

Lederer, A. L. and Prasad, J. (1991). The validation of a political model of information systems development cost estimating. In *SIGCPR '91: Proceedings of the 1991 conference on Computer Personnel Research*, pages 164–173, New York, NY, USA. ACM.

Lewis, C., Polson, P. G., Wharton, C., and Rieman, J. (1990). Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 235–242, New York, NY, USA. ACM.

Li, J., Huffman, S., and Tokuda, A. (2009). Good abandonment in mobile and pc internet search. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, New York, NY, USA. ACM.

Lida, B., Hull, S., and Pilcher, K. (2003). Breadcrumb navigation: An exploratory study of usage. *Usability News*, 5(1).

Lovins, J. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11:22–31.

Maeda, J. (2006). *The Laws of Simplicity (Simplicity: Design, Technology, Business, Life).* The MIT Press.

Mankoff, J., Dey, A. K., Hsieh, G., Kientz, J., Lederer, S., and Ames, M. (2003). Heuristic evaluation of ambient displays. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 169–176, New York, NY, USA. ACM.

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

Marchionini, G. (2006). Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46.

Marchionini, G. (2009). Executive summary. In *NSF Workshop on Information Seeking Support Systems*, page 6.

Marshall, C. C. and Shipman III, F. M. (1997). 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. ACM Press.

Mayer, R. (2001). *Multimedia learning*. Cambridge University Press.

Mayer, R. E. and Moreno, R. (2003). Nine ways to reduce cognitive load in multimedia learning. *Educational psychologist*, 38(1):43–52.

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

McGuffin, M. J. and schraefel, m. c. (2004). A comparison of hyperstructures: zzstructures, mspaces, and polyarchies. In *HYPERTEXT '04: Proceedings of the fifteenth ACM conference on Hypertext and hypermedia*, pages 153–162, New York, NY, USA. ACM Press.

Medlock, M., Wixon, D., Terrano, M., Romero, R., and Fulton, B. (2002). Using the RITE method to improve products: A definition and a case study. *Usability Professionals Association, Orlando FL July*.

Mizzaro, S., Nazzi, E., and Vassena, L. (2008). Retrieval of context-aware applications on mobile devices: How to evaluate? In *IIiX'08: Proceedings of the second conference on Information Interaction in Context*. ACM.

Moody, M. K. (1991). Documents search strategies and general reference search strategies: an analysis and comparison. *The Reference librarian*, 14(32):57–69.

Moore, G. A. (1991). *Crossing the chasm*. Harper Collins Publishers, New York.

Morris, M. R. (2008). A survey of collaborative web search practices. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1657–1660, New York, NY, USA. ACM.

Morris, M. R. (2009). RE: IP&M submission: comments on SearchTogether Analysis. Personal Communication.

Morris, M. R. and Horvitz, E. (2007a). S3: Storable, shareable search. In *Interact'07: 11th IFIP TC13 Conference in Human-Computer Interaction*.

Morris, M. R. and Horvitz, E. (2007b). SearchTogether: an interface for collaborative web search. In *UIST'07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 3–12. ACM Press New York, NY, USA.

Morris, M. R. and Teevan, J. (2008). Understanding groups properties as a means of improving collaborative search systems. In *1st International Workshop on Collaborative Information Retrieval*.

Morris, M. R., Teevan, J., and Bush, S. (2008). Enhancing collaborative web search with personalization: groupization, smart splitting, and group hit-highlighting. In *CSCW '08: Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 481–484, New York, NY, USA. ACM.

Mu, X. (2004). *Smartlinks in a video-based collaborative distance learning system: a cognitive model and evaluation study*. PhD thesis, University of North Carolina at Chapel Hill Chapel Hill, NC, USA.

Mumford, E. and Henshall, D. (1978). *Participative Approach to Computer Systems Design: A Case Study of the Introduction of a New Computer System*. Halsted Press, New York, NY, USA.

Nielsen, J. (1994). Usability inspection methods. In *CHI '94: Conference companion on Human factors in computing systems*, pages 413–414, New York, NY, USA. ACM.

Nielsen, J. (1999). *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, Thousand Oaks, CA, USA.

Nielsen, J. and Molich, R. (1990). 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. ACM.

Nielsen, J. and Phillips, V. (1993). Estimating the relative usability of two interfaces: Heuristic, formal, and empirical methods compared. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 214–221. ACM New York, NY, USA.

Norman, D. (2002). *The design of everyday things*. Basic Books.

Norman, D. and Draper, S. (1986). *User centered system design*. Lawrence Erlbaum Associates Hillsdale, NJ.

O'Brien, H. and Toms, E. (2005). Engagement as Process in Human-Computer Interactions. In *Proceedings of the American Society for Information Science and Technology Poster Session*.

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

O'Brien, H. L. and Toms, E. G. (2008). 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.

O'Brien, H. L., Toms, E. G., and Kalloway, E. K. (2008). Developing and evaluating a reliable measure of user engagement. In *Proceedings of the American Society for Information Science and Technology*, pages 1–10.

O'Day, V. L. and Jeffries, R. (1993). Information artisans: patterns of result sharing by information searchers. In *COCS '93: Proceedings of the conference on Organizational computing systems*, pages 98–107, New York, NY, USA. ACM.

Olson, G. and Moran, T. (1998). Commentary on "Damaged merchandise?". *Human-Computer Interaction*, 13(3):263–323.

Olson, J. R. and Olson, G. M. (1990). The growth of cognitive modeling in human-computer interaction since goms. *Human-Computer Interaction*, 5(2):221–265.

Paas, F., Renkl, A., and Sweller, J. (2003a). Cognitive Load Theory and Instructional Design: Recent Developments. *Educational Psychologist*, 38(1):1–4.

Paas, F., Tuovinen, J. E., Tabbers, H., and Van Gerven, P. W. M. (2003b). Cognitive Load Measurement as a Means to Advance Cognitive Load Theory. *Educational Psychologist*, 38(1):63–71.

Paek, T., Dumais, S., and Logan, R. (2004). WaveLens: A new view onto internet search results. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 727–734. ACM New York, NY, USA.

Paivio, A. (1969). Mental imagery in associative learning and memory. *Psychological Review*, 76(3):241–263.

Paul, S. A. and Morris, M. R. (2009). CoSense: enhancing sensemaking for collaborative web search. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1771–1780, New York, NY, USA. ACM.

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

Pharo, N. (1999). Web information search strategies: A model for classifying web interaction. In *CoLIS3: Proceedings of the 3rd International Conference on the Conceptions of the Library and Information Science*, pages 207–218.

Pharo, N. (2004). A new model of information behaviour based on the search situation transition schema. *Information Research*, 10:Paper 203.

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

Pickens, J. and Golovchinsky, G. (2007). Collaborative exploratory search. In *HCIR'07: Proceedings of the First Human Computer Interaction and Information Retrieval Workshop*, pages 21–22.

Pickens, J., Golovchinsky, G., Shah, C., Qvarfordt, P., and Back, M. (2008). Algorithmic mediation for collaborative exploratory search. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322, New York, NY, USA. ACM.

Pirolli, P. and Card, S. (1995). 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. ACM Press/Addison-Wesley Publishing Co.

Preece, J., Rogers, Y., and Sharp, H. (2001). *Beyond Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, Inc. New York, NY, USA.

Prekop, P. (2002). A qualitative study of collaborative information seeking. *Journal of Documentation*, 58(5):533–547.

Pressman, R. (1992). *Software engineering: A practitioner's approach*. McGraw-Hill, NY.

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

Ranganathan, S. (1960). *Colon classification: basic classification*. Asia Publication House.

Reid, G., Eggemeier, F., and Nygren, T. (1982). An individual differences approach to SWAT scale development(Subjective Workload Assessment Technique). In *26th Annual Meeting of the Human Factors Society*, pages 639–642.

Resnick, P. and Varian, H. R. (1997). Recommender systems. *Communications of the ACM*, 40(3):56–58.

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

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

Robertson, S. E. and Hancock-Beaulieu, M. M. (1992). On the evaluation of IR systems. *Information Processing and Management*, 28(4):457–466.

Robertson, S. E. and Sparck Jones, K. (1976). Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, 27(3):129–46.

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

Rowley, D. E. and Rhoades, D. G. (1992). The cognitive jogthrough: a fast-paced user interface evaluation procedure. In *CHI'92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 389–395. ACM New York, NY, USA.

Royce, W. W. (1970). Managing the development of large software systems: concepts and techniques. In *IEEE Wescon*, pages 1–9. IEEE.

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

Rubio, S., Diaz, E., Martin, J., and Puente, J. (2004). Evaluation of Subjective Mental Workload: A Comparison of SWAT, NASA-TLX, and Workload Profile Methods. *Applied Psychology*, 53(1):61.

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

Saracevic, T. (1995). Evaluation of evaluation in information retrieval. In *SIGIR '95: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 138–146, New York, NY, USA. ACM Press.

Saracevic, T. (1997). The stratified model of information retrieval interaction: Extension and applications. In *Proceedings of the ASIS Annual Meeting*, pages 313–327. American Society for Information Science.

Scapin, D. and Law, E. (2007). Review, Report and Refine Usability Evaluation Methods (R3 UEMs). In *COST294-MAUSE 3rd International Workshop*.

schraefel, m. c. (2009). Building knowledge: What's beyond keyword search? *Computer*, 42(3):52–59.

schraefel, m. c., Karam, M., and Zhao, S. (2003). Listen to the music: Audio preview cues for the exploration of online music. In *Interact'03: Proceedings of the Ninth IFIP TC13 International Conference on Human-Computer Interaction*. IOS Press.

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

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

schraefel, m. c., Zhu, Y., Modjeska, D., Wigdor, D., and Zhao, S. (2002). 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. ACM Press.

Schuler, D. and Namioka, A., editors (1993). *Participatory Design: Principles and Practices*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.

Shah, C. (2008). Toward Collaborative Information Seeking (CIS). In *Proceedings of the 1st International Workshop Collaborative Information Retrieval*.

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

Shneiderman, B. and Plaisant, C. (2006). 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. ACM.

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

Sitter, S. and Stein, A. (1992). Modeling the illocutionary aspects of information-seeking dialogues. *Information Processing and Management*, 28(2):165–180.

Smeaton, A. F., Foley, C., Gurrin, C., Lee, H., and McGivney, S. (2006). Collaborative searching for video using the físchlár system and a diamondtouch table. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 151–159, Washington, DC, USA. IEEE Computer Society.

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

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

Spencer, R. (2000). The streamlined cognitive walkthrough method, working around social constraints encountered in a software development company. In *CHI'00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 353–359. ACM New York, NY, USA.

Spink, A. (1997). Study of interactive feedback during mediated information retrieval. *Journal of the American Society for Information Science*, 48(5):382–394.

Stein, A. and Thiel, U. (1993). A conversational model of multimodal interaction in information systems. In *AAAI '93: Proceedings of 11th National Conference on Artificial Intelligence*, pages 283–288. The MIT Press.

Strauss, A. and Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage Publications.

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

Talja, S. (2002). Information sharing in academic communities: types and levels of collaboration in information seeking and use. *New Review of Information Behavior Research*, 3:143–159.

Tarrant, D., Carr, L., and Payne, T. (2008). Releasing the power of digital metadata: examining large networks of co-related publications. In *JCDL '08 Poster: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 471–471, New York, NY, USA. ACM.

Tedeschi, B. (1999). Good web site design can lead to healthy sales. Online Newspaper Article.

Teevan, J., Alvarado, C., Ackerman, M. S., and Karger, D. R. (2004). The perfect search engine is not enough: a study of orienteering behavior in directed search. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 415–422, New York, NY, USA. ACM Press.

Thimbleby, H. (2007). *Press on: principles of interaction programming*. The MIT Press.

Tischler, L. (2005). The beauty of simplicity. *Fast Company*, 100:52–59.

Tsang, P. and Velazquez, V. (1996). Diagnosticity and multidimensional subjective workload ratings. *Ergonomics*, 39(3):358–381.

Twidale, M. B., Nichols, D. M., and Paice, C. D. (1997). Browsing is a collaborative process. *Information Processing and Management*, 33(6):761–783.

Vredenburg, K., Mao, J., Smith, P., and Carey, T. (2002). A survey of user-centered design practice. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, pages 471–478. ACM New York, NY, USA.

Wharton, C., Rieman, J., Lewis, C., and Polson, P. (1994). The cognitive walkthrough method: a practitioner's guide. pages 105–140.

White, H. D. and McCain, K. W. (1998). 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.

White, R. and Roth, R. A. (2009). *Exploratory Search: Beyond the Query-Response Paradigm*. Morgan & Claypool.

White, R. W. and Drucker, S. M. (2007). Investigating behavioral variability in web search. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 21–30, New York, NY, USA. ACM.

White, R. W., Kules, B., Drucker, S. M., and schraefel, m. c. (2006). Introduction. *Communications of the ACM*, 49(4):36–39.

Wildemuth, B. M. (2004). The effects of domain knowledge on search tactic formulation. *Journal of the American Society for Information Science and Technology*, 55(3):246–258.

Wildemuth, B. M., de Bliek, R., He, S., and Friedman, C. P. (1992). Search moves made by novice end users (1). In *ASIS '92: Proceedings of the 55th annual meeting on Celebrating change : information management on the move*, pages 154–161, Silver Springs, MD, USA. American Society for Information Science.

Wildemuth, B. M., Jacob, Fullington, de Bliek, R., and Friedman, C. P. (1991). A detailed analysis of end-user search behaviors. *ASIS'91: Proceedings of the 54th ASIS Annual Meeting*, pages 302–12.

Wilkinson, R. (1994). Effective retrieval of structured documents. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 311–317, New York, NY, USA. Springer-Verlag New York, Inc.

Wilson, M., Russell, A., schraefel, m. c., and Smith, D. A. (2006a). mspace mobile: a ui gestalt to support on-the-go info-interaction. In *CHI '06: CHI '06 extended abstracts on Human factors in computing systems*, pages 247–250, New York, NY, USA. ACM.

Wilson, M. L. (2007). 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.

Wilson, M. L. (2009). Keyword search: Quite exploratory actually. In *Proceedings of the 3rd International Workshop on Human-Computer Interaction and Information Retrieval (HCIR'09)*, pages 106–108.

Wilson, M. L., André, P., and schraefel, m. c. (2008). Backward highlighting: enhancing faceted search. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 235–238, New York, NY, USA. ACM.

Wilson, M. L., Kules, B., schraefel, m. c., and Shneiderman, B. (2009a). Designing Future Search Interfaces for the Web: Leveraging Interactive Result Visualization to Aid Exploration and Discovery. *Foundations and Trends in Web Science*, 1(4):1–100.

Wilson, M. L., Russell, A., Smith, D., and schraefel, m. c. (2006b). mspace mobile: Exploring support for mobile tasks. In *Proceedings of the British Conference on Human-Computer Interaction*.

Wilson, M. L. and schraefel, m. c. (2006). mspace: What do numbers and totals mean in a flexible semantic browser. In *SWUI'06: The 3rd International Semantic Web User Interaction Workshop*.

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

Wilson, M. L. and schraefel, m. c. (2008a). Evaluating collaborative search interfaces with information seeking theory. In *1st International Collaborative Search Workshop*.

Wilson, M. L. and schraefel, m. c. (2008b). Improving exploratory search interfaces: Adding value or information overload? In *Proceedings of the Second Human Computer Interaction and Information Retrieval Workshop (HCIR08)*.

Wilson, M. L. and schraefel, m. c. (2008c). A longitudinal study of exploratory and keyword search. In *JCDL'08: ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 52–56. IEEE Computer Society.

Wilson, M. L. and schraefel, m. c. (2009a). Evaluating collaborative information seeking interfaces with a search-oriented inspection method and re-framed information seeking theory. *Information Processing & Management*, (In Press).

Wilson, M. L. and schraefel, m. c. (2009b). Reading between the lines: identifying user behaviour between logged interactions. In *SIGIR09 Workshop Position Paper: Understanding the User - Logging and interpreting user interactions in information search and retrieval*. Position Paper.

Wilson, M. L. and schraefel, m. c. (2009c). Sii: the lightweight analytical search interface inspector. In *JCDL'09 Workshop on Lightweight User-Friendly Evaluation Methods for Digital Librarians*. DLib Magazine.

Wilson, M. L., schraefel, m. c., and White, R. W. (2009b). Evaluating advanced search interfaces using established information-seeking models. *Journal of the American Society for Information Science and Technology*, 60(7):1407–1422.

Wilson, T. (1981). On User Studies and Information Needs. *Journal of Documentation*, 37(1):3–15.

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

Winograd, T. and Flores, F. (1986). *Understanding computers and cognition.* Ablex Publishing Corp. Norwood, NJ, USA.

Yee, K.-P., Swearingen, K., Li, K., and Hearst, M. (2003). 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. ACM Press.

Zhang, J. and Marchionini, G. (2005). 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.