

Evaluating Advanced Search Interfaces using Established Information-Seeking Models

Max L. Wilson[†], m.c. schraefel

School of Electronics and Computer Science, University of Southampton, Highfield, Southampton, SO17 1BJ, UK. Tel: +44 (0) 2380 598367. Fax: +44 (0) 23 8059 2865.

Email: mlw05r@ecs.soton.ac.uk, mc@ecs.soton.ac.uk

Ryen W. White

Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA.

Email: ryenw@microsoft.com

When users have poorly defined or complex goals, search interfaces offering only keyword searching facilities provide inadequate support to help them reach their information-seeking objectives. The emergence of interfaces with more advanced capabilities such as faceted browsing and result clustering can go some way toward addressing such problems. The evaluation of these interfaces, however, is challenging since they generally offer diverse and versatile search environments that introduce overwhelming amounts of independent variables to user studies; choosing the interface object as the only independent variable in a study would reveal very little about why one design out-performs another. Nonetheless if we could effectively compare these interfaces we would have a way to determine which was best for a given scenario and begin to learn why. In this article we present a formative inspection framework for the evaluation of advanced search interfaces through the quantification of the strengths and weaknesses of the interfaces in supporting user tactics and varying user conditions. This framework combines established models of users, user needs, and user behaviours to achieve this. The framework is applied to evaluate three search interfaces and demonstrates the potential value of this approach to interactive IR evaluation.

1. Introduction

Keyword search has become the default standard for modern applications of Information Retrieval (IR) such as Web search. While this approach is powerful, it does not support users well when they have poorly defined goals or complex questions, have insufficient pre-search knowledge, or may be using a system with poorly-defined or unpredictable indexing (Belkin, 2000; Salton & Buckley, 1990; Spink *et al.*, 1998; White *et al.*, 2006). To support these situations, which may be up to 83% of all queries (White & Drucker, 2007), richer modes of search, such as faceted browsing (Hearst, 2000) and result clustering (Zamir *et al.*, 1997) are being developed. These modes can capitalise on users' ability to filter and navigate through information using recognition rather than recall. While these more interactive modes of search provide increasingly versatile combinations of functions, the challenge is not to simply

[†] Primary Contact Author

add more features but to combine them to produce synergetic designs (Marchionini, 2006). However, combining multiple features into a single interface creates a fascinating set of challenges in how to compare the support the design offers against other similar interfaces. To evaluate rich support for search, metrics need to consider strategies of information-seeking behaviour so we may understand how well, or not, they are endorsed by a design. The Interactive Track at the Text Retrieval Conference (TREC) (Dumais & Belkin, 2005; Harman, 1997) attempted to provide a methodology for comparing interfaces to IR systems between different experimental sites (Lagergren & Over, 1998). Unfortunately, the experimental methodology did not conform to TREC and Cranfield (Cleverdon *et al.*, 1966) traditions, where only one aspect of a system was varied at any point in time (usually the document ranking algorithm). As a result the track was ultimately retired due to issues with inter-system comparability. However it did make a significant contribution to informing the community's understanding of the search process.

In this article we describe a novel formative application of established models of information-seeking behaviour to inspect and evaluate the support provided for search by three faceted browsers. We use the lessons learned from information-seeking research to estimate, and therefore predict, the support for user needs and the tactics they may employ to meet them. By assessing the ease of using different tactics as a measure of support, the strengths and weaknesses of design can be identified to motivate redesign that can potentially improve the search experience. This approach is used to produce a measure for comparing systems that may be otherwise difficult to compare. It is formative in the sense that it is meant to serve only as part of the overall design process, providing valuable information that can inform design refinements prior to presentation to human participants later in the process.

The focus in IR evaluation has been on designing experiments that are: *insightful*, to assess the attributes, on which they focus, successfully; *affordable*, in respect to the cost of creating and running experiments; *repeatable*, so that others can build on results; and *explainable*, to guide subsequent improvements (Liu & Oard, 2006). Our evaluation framework adheres to these four principles, and through simulating core user interactions, allows for refinements to be made to designs during their formative development. Simulations of user interaction behaviour have been proposed to test the usability of Web sites (Chi *et al.*, 2001), navigation of the World Wide Web (Fu & Pirolli, 2007), and evaluate the effectiveness of implicit relevance feedback algorithms (White *et al.*, 2005) without human participants. The framework proposed in this article is similar in some respects, but is based on established models of information-seeking behaviour that are perhaps more appropriate for IR evaluation, rather than cognitive theory and experimenter intuition.

We first propose the application of a combination of information-seeking models as part of an evaluation framework to estimate the influence of the strengths and weaknesses found in several interface designs in terms of their support for established search tactics. Second, we suggest that these strengths and weaknesses can be attributed to different search conditions to identify support for particular types of user. We begin with an overview of relevant work on modelling information-seeking behaviour. A combination of these models is included in an inspection-based evaluation framework that is then described. The inspection framework is then applied to three example advanced search interfaces to comparatively evaluate the interface support they offer. We conclude with a description of the findings, their implications, and the further work required to transform this research into a robust reusable framework for evaluating rich search environments. Our conclusions also include a discussion of the limitations of our approach and how these limitations can be overcome.

2. Models of Interaction and Information-Seeking Behaviour

The interactive, cognitive and relevance revolutions in Information-Seeking and Retrieval (ISR) literature (Robertson & Hancock-Beaulieu, 1992) depict an interest in the human element in search dating back over four decades. A deeper understanding of how problems

were translated into query statements (Taylor, 1968), the emergence of relevance feedback as a mechanism critically dependent on user interaction (Rocchio, 1971), and an understanding of the role of knowledge deprivation as a catalyst for information seeking (Belkin *et al.*, 1982), provided some of the initial momentum for those movements and underscored the importance of the user in the search process. A number of general information behaviour models have been proposed in the years since these early advances (Dervin & Nilan, 1986; Bates, 1990; Kuhlthau, 1991; Belkin *et al.*, 1995; Marchionini, 1995; Ingwersen, 1996; Saracevic, 1997; Spink & Wilson, 1999). Although many models have been proposed we elected to use two in particular for our evaluation framework that fit together effectively and were manageable in their size and scope. The models not selected at this time are considered for future versions of our framework. Any advances will have to consider the effect on this utility-complexity trade-off before being accepted for inclusion in the framework. We now describe the three most relevant models for our application, one to drive the design and two to be used as part of the final model.

2.1 Stratified Models

Stratified models help determine the range of abstraction levels that each need to be considered, when evaluating the effectiveness of a search system. Saracevic, for example, proposed a model of stratified interaction including hardware, software, user cognition, and situational levels, which suggests that the system and the user have *equal* constraints on the success of a search (Saracevic, 1997). A system involves hardware, processing, and data structures. For a user, their interaction involves cognitive, affective and situational levels, which represent their interpretation, motivation, and their requirements respectively. The system side of this model was later extended by Spink and colleagues (1998), who added a graduated relevance dimension, and Bates (2002), who identified additional levels that interact and affect each other. The key point made by these stratified models is that even if an effective searching algorithm is implemented, or an intuitive user interface is designed, poor indexing or hardware can hinder the entire value of the system; acting as the weakest link in the chain. Similarly, a user's interpretation of results may seriously affect their success in achieving their goals with the system. By exposing the different layers, each can be individually checked for bottlenecks in performance.

2.2 Episodic Models

Belkin and colleagues (1995) produced an episodic model to define and understand the flow in scenarios of human-system interactions that overcomes some of the shortcomings of the stratified approach. The flow definitions are called "scripts" that define the typical steps of interaction between a user and an information system, including multiple exit points for variances such as success or failure. To do this, Belkin and colleagues first highlight four binary dimensions that define 16 unique Information-Seeking Strategies (ISS) (Belkin *et al.*, 1993). They calculated separate scripts for each of these 16 ISS conditions, which allow for switching between them. The ISS dimensions are *Method*, *Goal*, *Mode* and *Resource* and in combination produce sixteen conditions shown in Table 1.

The *Method* dimension defines whether a user is either searching for a particular information object, or scanning a set of information objects. For example, these activities are easily differentiable as finding a specific paper in order to get its reference details, or by searching for any paper that may or may not exist, that can be used to support a point. The *Goal* dimension defines whether a user is learning about something or selecting something. Using the same bibliographic example these can be differentiated as learning who may be the key authors on a topic or selecting a particular paper. The *Mode* dimension defines where a user is recognising and specifying something. One might remember that there was a useful publication at a recent conference in their area of interest, and so is trying to identify it in the proceedings, or may have known the author, title and year and has typed them into a search box on the ACM Portal. The *Resource* dimension defines whether a user is looking for

information items or meta-data about an information item. Usually, within a bibliographic repository users are trying to find specific papers, but it is possible that the user is trying to find significant authors, keywords, or events relating to particular papers.

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 1: Information Seeking Strategies (Belkin *et al.*, 1993).

For example, traditional Web search engines such as Google, Yahoo, and Live Search are best used for ISS15, where the user is searching (*Method*) to select (*Goal*) by specifying (*Mode*) attributes of a specific information object (*Resource*). Consequently traditional Web search engines least support users who are scanning (*Method*) to learn (*Goal*) by recognising (*Method*) some metadata about an information object (*Goal*): this is ISS2. Faceted browsing, described more in Section 3, tries to support users by presenting all the meta-information to the user in advance and letting them choose. Conversely, this best supports ISS2, but may inadequately 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.

This model has been applied to Web searching and navigation studies involving human participants (Belkin *et al.*, 1995; Pharo, 1999), with findings suggesting that the model is insufficiently exhaustive and that there is a potential for interdependency between the method of searching and the mode of retrieval (Pharo, 1999). In an effort to address such concerns Belkin and Cool (2002) extended and expanded upon the four dimensions in the episodic model to incorporate all possible interactions between people and information within five facets. The new model, however, goes into more detail than is easily coverable and reusable within the constraints of this article: more than a hundred unique conditions are produced. We hypothesise that the initial four dimensions are sufficiently expressive to classify seeking behaviours in the framework we present in this article, especially as it represents the core of the larger model, which was later validated by Huvilla and Widen-Wulff (2006).

2.3 Strategic Models

Strategic models define the different strategies and tactics that a user may employ when interacting with information, such as refining a search or returning to the beginning of a search session and beginning anew. Bates proposed a model containing four levels of search strategies: “move”, “tactic”, “stratagem”, and “strategy” (Bates, 1990). 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 endless combinations of moves that can be used to support a tactic, which depends on system implementations. She defines 32 specific information search tactics, as listed in Table 2. “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 what the user is currently working on.

Tactic Category	List of Tactics
Monitoring Tactics	CHECK, WEIGH, PATTERN, CORRECT, RECORD
File Structure Tactics	BIBBLE, SELECT, SURVEY, CUT, STRETCH, SCAFFOLD, CLEAVE
Search Formulation Tactics	SPECIFY, EXHAUST, REDUCE, PARALLEL, PINPOINT, BLOCK
Term Tactics	SUPER, SUB, RELATE, NEIGHBOR, TRACE, VARY, FIX, REARRANGE, CONTRARY, RESPELL, RESPACE
Idea Tactics	RESCUE, BREACH, FOCUS

Table 2: Bates’ 32 Tactics and their definitions. See Bates (1979a) and Bates (1979b) for full definitions of these tactics.

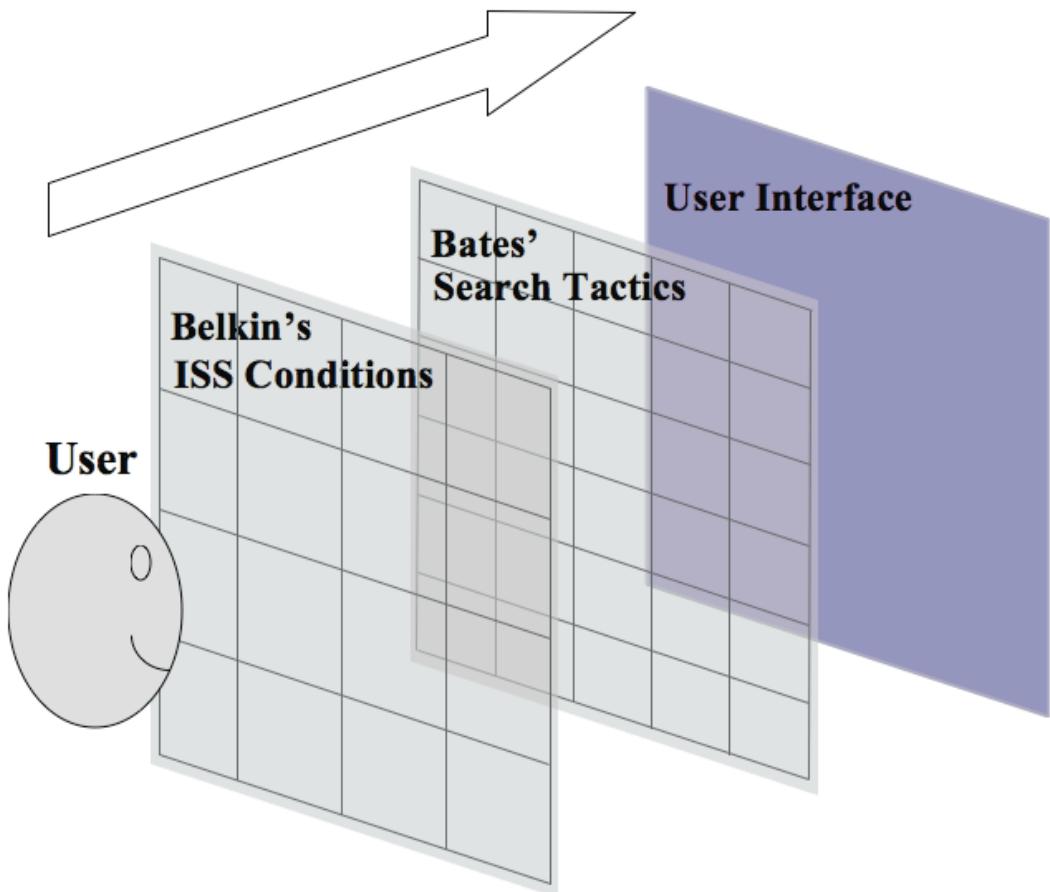


Figure 1: The interaction of the models within our evaluation framework. The parts of each layer act as a viewfinder onto the next layer.

We suggest that strategies and even stratagems may affect the choice of a search interface and the decision will be partly based upon the tactics, such as comparing and collecting, that it supports, as well as the current conditions for the user, such as Belkin's ISSs. Therefore we propose a combination of Bates' lower levels of search activity and Belkin's ISS model to produce a framework for the automated evaluation of advanced search interfaces. This combination is visualised in Figure 1. At any one time, a user is viewing the user interface (UI) from one of 16 ISS conditions, and sees it in terms of the tactics she can employ. The conditions and the tactics act as filters, restricting the space of possible interactions with the interface. 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 a speculative metric between the layers. So each tactic has a total score of how easily they can be applied with each part of a UI. In turn, when a user looks at the potential tactics through one of the 16 ISS conditions, they see how many moves they can make with each tactic. This mapping is described in more detail in Section 4 of the article.

In the next section we describe three faceted browsers that are compared in our study. These browsers were chosen to demonstrate our evaluation method for three reasons: (i) all are faceted browsers, and so provide a good example of how systems designed with similar aims can produce very different designs, (ii) faceted browsers offer a wide range of features to support the search process. As such they are useful for illustrating the potential of our evaluation framework. Comparing systems offering only keyword-search facilities would not provide as many features or such a diverse discussion; especially when the measure is of how broad the interface design is, and (iii) each browser has their own example datasets and by comparing different examples, we can show that the framework is dataset and implementation agnostic.

3. Faceted Browsers

Faceted browsers are an example class of advanced search interfaces that present different meta-data attributes as a series of orthogonal categories that can be applied in combination to produce multiple filters over the dataset. Typically the different meta-data attributes, known as facets, are created around a series of "Target Objects", which represent the information objects the user might be searching for. For example, in a bibliographic repository, Target Objects would be the instances of literature being held in the repository. Facets in the bibliographic example, therefore, might be author, publication venue, publication date, etc. Users of faceted browsers can then select one item, or sometimes more, from any or all of the facets to express their search needs. Through modelling a domain in a faceted structure, direct manipulation can be used to construct queries (Shneiderman, 1981). Thus, when a user is not clear on appropriate terminology or the meta-data is unpredictable, they do not have to estimate search terms, but can make selections in each facet to apply multiple filters to the dataset.

In the remainder of this section we describe three example faceted browsers developed in academia and motivated by improving user interaction with information¹. Other faceted browsers exist in industry and academia, but are not described here. Endeca² is a commercial faceted browser that is not publicly accessible for research purposes. More recently, \facet has been developed to use faceted browsing for supporting information architecture and evolution

¹ Since performing this research, both mSpace and RB++, described below, have released iterations of their software. The versions described here, and evaluated below, were correct at the time of writing.

² <http://www.endeca.com> - Endeca

(Hildebreand *et al.*, 2006). However, \facet is subtly different to the three interfaces chosen, which are aimed at providing access to information rather than exposing structured data.

3.1 Flamenco

Flamenco³ (Yee *et al.*, 2003), shown in Figure 2, supports both keyword search and faceted browsing, accounting for both those who know their target and those, for example, who have little existing knowledge about the domain. The initial display shows all the possible facets in two columns, with vertical scroll as necessary.

The screenshot shows the Flamenco browser interface for Nobel Prize Winners from 1901 to 2004. At the top, there's a navigation bar with links for Save Search, History and Settings, Return to Search, New Search, and Logout. The main search area has a search bar and a dropdown menu for selecting 'all items' or 'in current results'. Below the search bar, a message says 'These terms define your current search. Click the [x] to remove a term.' followed by the selected filters: 'GENDER: male' and 'YEAR: 1910s > 1917'. To the left, a sidebar lists facets categorized by color: GENDER (all > male), COUNTRY (group results: Denmark 2, United Kingdom 1), AFFILIATION (group results: United Kingdom 1), PRIZE (group results: literature 2, physics 1), and YEAR (all > 1910s > 1917). A 'Recently Viewed Items' section with a 'Go to Item History' link is also present. On the right, the results are displayed as a grid of three portraits with their names and years: Charles Glover Bar... 1877-1944, Henrik Pontoppidan 1857-1943, and Karl Gjellerup 1857-1919.

Figure 2: Flamenco browser showing an example dataset of Nobel Prize Winners

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 of the interface and the results down the right column. The search box remaining at the top left and a breadcrumb trail is located just above the search results, which presents the sequence of selections and queries made by a user. A search query 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 query terms 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 facet, the sub-categories within the facet are shown and a per-facet breadcrumb is shown to depict the history of selections made by the user within the one facet. If there are no sub-items, the facet is effectively minimised (facet representations grow vertically with the number of options within it). If facets are hierarchical, results are automatically clustered into the sub-categories of the latest selection. The user may optionally group the results by any other facet through a single interaction provided by the presence of a new link along side of each facet name. Any potential option for selection is accompanied by numeric volume indicators (NVIs – Wilson & schraefel, (2006)), to estimate the number of Target Objects that can be reached by its selection.

³ <http://flamenco.berkeley.edu/> - Flamenco Home

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

3.2 mSpace

mSpace⁴ (schraefel et al., 2006), shown in Figure 3, also supports both keyword search and faceted selections. Normally, the user is presented with four panels: Facet Browser, Interests, Information, and Preview Cues. These panels remain persistently available throughout the subsequent interactions with mSpace, using a zooming interface technique.

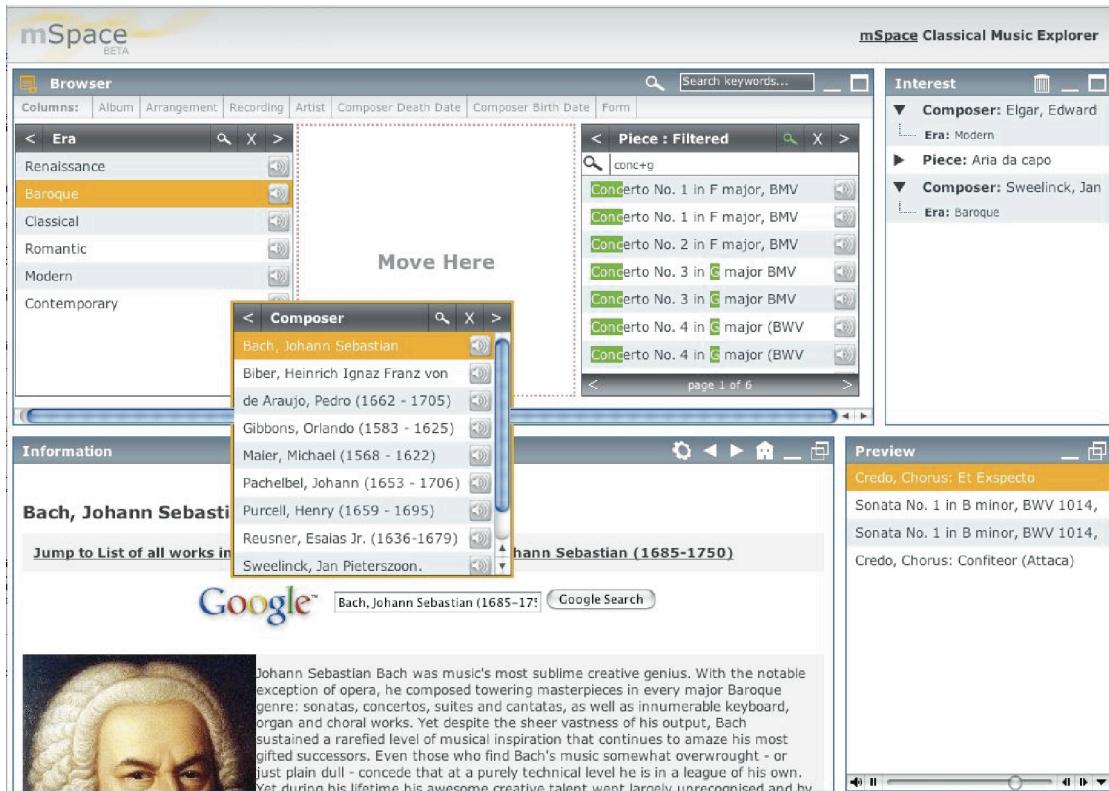


Figure 3: the mSpace browser showing an example dataset of Classical Music.

The facet browser panel holds a series of active facets in columns stood side by side in a horizontal line, and optional facets are listed in the header above. Within a facet, scrolling can be reduced using a per-facet string filter. The number and order of active facets can be changed freely, using “drag and drop”, and horizontal scrolling of the horizontal column-set is provided if necessary. Unlike Flamenco, users may make multiple selections within any facet, and the results combined using logical “OR”. NVIs are also presented for each facet item. While most faceted browsers allow the reordering of facets for aesthetic purposes, the order of the facets in mSpace matters and forms a temporary hierarchy, where filtering, according to a selection in a facet, only occurs in the facets to the right. This left-to-right filtering is also found in the iTunes browser, where a selected Artist filters the albums to the

⁴ <http://mspace.fm/> - mSpace

right, but not the genres to the left. This temporary hierarchy and directional filtering means that the inter-column relationships are exposed, such as all the Artists in a Genre, *and* all the Albums created by one of them. The hierarchy is “temporary” because, unlike in iTunes, mSpace users could re-order the columns to see all of the Genres of a selected Artist, and the Albums from one of them. These inter-column relationships are not conveyed by Flamenco. By reordering the facet-columns in mSpace, the users can construct a path towards Target Objects using meta-data they are comfortable with. Other facets can be simply removed from the active set to avoid confusion. Finally, the ordering of facet-columns also represents an in-place breadcrumb of search decisions.

The second panel, at the top right of the interface, is an interest box that allows users to retain objects, much like a favourites or bookmark list in a Web browser. Uniquely, this saving affordance allows the regeneration of the path taken to find an item, which represents the sequential selections made through any number of facets to find a set of results. The panel on the bottom-left of the interface provides information about the last selection. If the most recent selection is a Target Object⁵, then a user will expect to find the details of the Target Object here: this means that when viewing a specific object, a user has not lost the context of their browsing path. However, in the classical music demo, users can also see information about any facet item, such as composers or eras. This box can also be used to present search results, through either the keyword search at the top-right or as constrained by the selections in the facets.

A preview cue panel is shown on the bottom-right of the interface. This presents a space on the screen for giving example multi-media of Target Objects. One unique attribute of this preview cue panel is that it can be used to display example Target Objects for anything in the facets. In the classical music demo, users can hear examples of classical music from each era, composer, arrangement, etc. This is triggered by ‘play’ icons that appear at the right-hand side of each item in the facet contents.

3.3 RB++

The Relation Browser (Zhang & Marchionini, 2005), named RB++⁶ and shown in Figure 4 currently presents all the facets and their contents persistently: these facets are listed across the top of the UI as columns and grow/shrink to fit on the screen. Users can reorder the facet-columns for aesthetic reasons, using a drop down list that functions as both a mechanism for changing the facet presented in the column below and also, therefore, for displaying label of the chosen facet. Like mSpace, there is no separate breadcrumb visualisation to keep track of the selections made by the user, but the combination of filters is clear as the each facet, and their selections, are persistently displayed, rather than removed for space optimisation, like in Flamenco.

NVIs are presented next to each item in each facet and also represented as an in-place bar graph behind their labels. The population of the bar represents both the number of achievable Target Objects from making that selection and, uniquely, the number of total Target Objects in the dataset. Hovering over items in each facet previews the effects of the selection on each of these NVIs and is made persistent by clicking.

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,

⁵ In mSpace, a column-facet may consist of actual Target Objects, such as a list of musical pieces, so that searchers can use particular Target Objects as filters on the other column-facets.

⁶ <http://idl.ils.unc.edu/rave/> - Interactive Design Laboratory Presents RAVE

the previous selections above are transformed into a Boolean statement, shown at the bottom of the interface, representing the selections, much like a breadcrumb but without temporal order. The facets are also transformed to represent the subset of Target Objects that had been previously achieved through selections in the facets. Thus NVIs represent the number of Target Objects in the new subset. Any subsequent facet selections automatically filter the search results. The Target Object is displayed in a new window once selected in the results.



Figure 4: The RB++ browser showing an example dataset of a movie archive after pressing the ‘Search’ button.

3.4 Browser Datasets

Each of the browsers compared in this study has a different dataset, and have been chosen purposefully to highlight the advantage that the framework described below is dataset agnostic. In the evaluation performed below, the means provided by each interface for searcher to perform different tactics are assessed, and the content of the datasets is not considered a factor in their ability to search and browse. Further, all three browsers have been applied to more than one dataset, including different types of multimedia. As the framework is described in more detail below, it will become clear that the search-based functionality being evaluated is common to all types of data and metadata, and can span across different domains.

The Flamenco browser is exploring Nobel Prize winners, and provides both metadata and multimedia, in the form of pictures, for each person. The mSpace browser above is exploring Classical Music, and has both metadata and multimedia, in the form of pictures and audio, for composers and pieces, respectively. Finally, the RB++ browser above is exploring government statistics and has both metadata and reports, containing various multimedia. Each of these datasets include multiple, orthogonal facets, which could be rendered flatly or hierarchically, and each interface handles hierarchical facets differently. Flamenco allows users to, from top to bottom, make subsequent selections within each layer of a hierarchical facet. mSpace, on the other hand, puts each level of a hierarchical facet into different columns, so that selections can be made in any layer and in any order. RB++ maintains only one key layer per facet.

3.5 Summary of Browsers

It is evident from the description of the interfaces presented in this section so far that they offer both a wide range of different functionality, as shown in Table 3, and very different implementations of overlapping functionality. It is these broad differences between interfaces that make comparisons problematic. The evaluation framework we describe in this article was designed to robustly handle such situations. In the next section we present the framework we use to compare the interfaces.

Feature	Flamenco	mSpace	RB++
Facets	Hierarchies, Vertical	Direction, Horizontal	Horizontal
Selections	Single only	Multiple, by choice	Multiple, by default
Bread Crumb	Separate ordered display	Order, by columns	Unordered columns
Result Viewer	Separate page	In context of facets	Outsourced
Previews	N/A	Multimedia	Number of results
NVI display	Per facet item	N/A	X out of Y
Results Tools	Grouping by facet items	Collection space	Sorting by attributes
Keyword Search	Static	Interactive	N/A

Table 3: Summary of Features in each browser, explained in more detail in Section 5.

4. Evaluation Framework

We developed a framework for the comparative evaluation of multiple advanced search interfaces during the formative stages of the design process. In the design of our framework we employ aspects of two models of interaction behaviour: the ISS conditions from the episodic model (Belkin *et al.*, 1993) and the levels of search strategies (i.e., moves, tactics, stratagems, and strategies) presented by Bates (1990) in her strategic model. In particular, the “moves” of Bates’ model are used to speculatively quantify the “tactics” she later defines. At this stage in the development of the evaluation framework “stratagems” and “strategies” are ignored, as their level of definition requires further work. While Belkin and colleagues (1993) used the ISS conditions to motivate the design of an IR system, we combine the conditions with metrics (the values produced by Bates’ model) to predict the support provided for users by different implementations.

The framework comprises the following four stages:

1. Identify the features, and their interactions, in each interface
2. Count how many moves it takes to employ each tactic with each feature
3. Process the results data
4. Perform visual analyses

These steps cover preparation, measurement, data processing, and analysis. They are each described in more detail below and are followed by an example.

Stage 1: Feature Identification.

First, each individual interface feature and its afforded interactions must be identified for each participating interface. Many of the same features may appear in each design, such as a keyword search function, and so a collated list of all the unique features from all the interfaces must be generated. However, if the comparison is of different prototype designs, rather than of three different systems, then they are likely to have identical sets of features

defined by the same original specification. Regardless, each interface will be evaluated against the *complete* list in the following steps.

As this framework measures the breadth of support provided to users by each interface, completeness is important. It is not possible to objectively identify the full set of features, as it may not be clear, for example, if multiple selection, as provided by mSpace and RB++, should be treated as a separate feature from the ability to make selections in facets. In the example below they are treated separately as not all of the browsers provide multiple-selection of facet-items, and its presence in the two browsers permits different tactics to making a single selection. Consequently, the analysis can be strengthened by the list being constructed separately by multiple human judges working independently, and then compared.

Stage 2: Measuring Support for Tactics.

Following the simple process in Figure 5, each interface feature is addressed one at a time, for each design. For the current feature of the current design, the moves required to employ each tactic are counted. This produces a series of tables, one for each design, where tactics are listed across the top and the interface features down the side. An example table is shown in Table 4. The count of moves is noted in the appropriate cross section between feature and tactic. No support by a feature for a tactic counts as 0. Requiring four moves to use the feature in support of a tactic produces a score of 4.

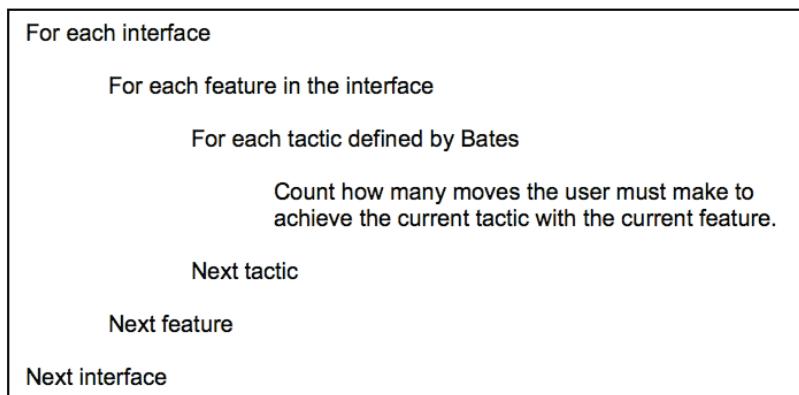


Figure 5: Scoring algorithm used in Step 2.

Interface Feature	Tot. by Feature	CHECK	WEIGH	PATTERN	RECORD	...
Keyword Search	4		1			...
Filtering	0					...
Grouping	9	1	1	1		...
...
Tot. by Tactic		3	5	2	0	...

Table 4: Partial table of results, showing the first 6 tactics of 32 and three example interface features.

As with some existing models, such as the GOMS model (Card *et al.*, 1980), optimal user interaction is measured and so repeat and optional moves are ignored. For example, selecting multiple items involves choosing and selecting 2+ items, selecting a third or fourth item is considered as repeating some of the moves required when selecting 2+ items. Optional moves include scrolling: a desired item may be the first or last item and the optimal situation is that it is one of the items is visible without scrolling.

Like with the identification of features, multiple judges can be used to strengthen the analysis, especially when the evaluators are inexperienced with applying the framework. Like other techniques that can involve multiple judges, such as ethnographic observation or inductive coding of qualitative data, inter-judge reliability can be measured using techniques such as Cohen or Fleiss' Kappa.

Stage 3: Data Processing.

With the process above, a poorly designed interaction produces a high score and better designs reduce towards a score of one. It is not possible to perform every tactic with every feature of a user interface, and such cases, under the steps described above, produce a score of 0. Consequently, before metrics can be summarised, the numbers, except for zeros, must all be inverted. When inverted, better designs receive a higher score, with a single move achieving a score of 1. Poorer designs approach a score of 0, where no support is provided at all. These inverted metrics can then be summed by feature and by tactic. This calculates the support provided by a feature for all tactics and the support provided for a tactic across all features, respectively. Summary scores for each feature can be seen towards the left of Table 4. Summary scores for each tactic are produced at the bottom of each column in the table.

Stage 4: Visual Analysis

Once the results have been processed in Stage 3, the results are ready for analysis. To support this analysis, 3 graphs can be produced to represent the results visually. For example, a graph can be produced including the summed values for each feature in each design. Strong features will produce tall bars, and a comparison of user effort within and between systems can indicate a strong feature design.

Another graph can be produced including the summed values for each tactic in each design. Again, tall bars indicate strong support for a tactic. This comparison may identify tactics that may require improved support through redesign. Finally, each tactic supports particular parts of Belkin's four dimensions of user information-seeking conditions. For example, CHECK, a tactic for users checking their decisions so far, supports users who are trying to learn as their *Goal*. Table 5 shows the 16 unique conditions produced by the four dimensions and the tactics that support them. The mapping shown in Table 4 was produced by careful analysis and discussion of the terms in the two models. Part of our future work will be to validate these decisions. For each of the sixteen conditions, the sum of the total support values can then be calculated. The calculated value for each condition can be graphed to show the difference in support for different user conditions. As results would be biased towards ISS conditions that have more assigned tactics, clearer results are produced by normalising the results by the number of tactics assigned to each condition.

To provide context we now present an example. A defining feature in faceted browsers is the ability to make a selection within a facet. The identification of this feature represents Step 1 in the process. Step 2 involves counting the number of moves it takes to complete the entire process of making a selection. The series of interactions required to do this may involve multiple tactics. For example, the first tactic is to CHECK the decisions made. If the interface responds to a selection by moving it to a list elsewhere in the page, the user has the added move of locating the "selected items list" on the page, before finding and reading their selection. However, this added move (scoring 3) takes more effort than if the selection remained in place (scoring 2). To handle this appropriately in the framework we invert their scores to give 1/3 and 1/2 respectively. The within-facet selection feature produces scores for

other tactics used and other features will also produce additional scores for the CHECK tactic. In the section to follow it is the sum of all tactics supported by a single feature and the sum of all features support for a single tactic that are visualised graphically.

ISS	Tactics that support this condition
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 5: The Tactics (from Table 2) assigned to each ISS condition identified in Table 1.

5. Trial Results

The above procedure was follow and applied to the three faceted browsers discussed Section 3. As we recommend in Section 4, the features of each interface were identified and

compared by two evaluators. While some decisions are clear, such as the difference between using the facets and performing a keyword search, a good example of a less clear, but important, distinction is between making and changing a selection in the facets. While these could have been combined when analysing mSpace and RB++ alone, the two are made distinct by the interactions required to change a selection in Flamenco. As such, evaluators may find it useful to think of the full set of actions a user can make, rather than identifying different sections of the layout. This may be especially clear, as the main action in the evaluation process is to count the number of “moves” required to carry out each tactic by making that action. The results of performing the evaluation on the three browsers are described below.

In the remainder of this section we describe some of the more pertinent results discovered when applying the framework to the three faceted browsers above. The chosen results are strong examples of what can be learned from its application.

5.1 Support for Tactics

Users need to adopt a variety of techniques to search effectively in the complex environments for which advanced search interfaces are suited. An essential aspect of interface design for information systems lies in how well it facilitates information seeking within its target domain. To assess this we measure the support offered by each of the faceted browsers for each of Bates’ tactics. Figure 6 shows the support provided by each interface for each of the 32 known tactics. A number of observations can be drawn from Figure 6. First, each interface has a higher peak for SURVEY. This is an expected peak when evaluating faceted interfaces because the user is presented with optional selections at each stage. This peak would not be so visible in keyword only interfaces.

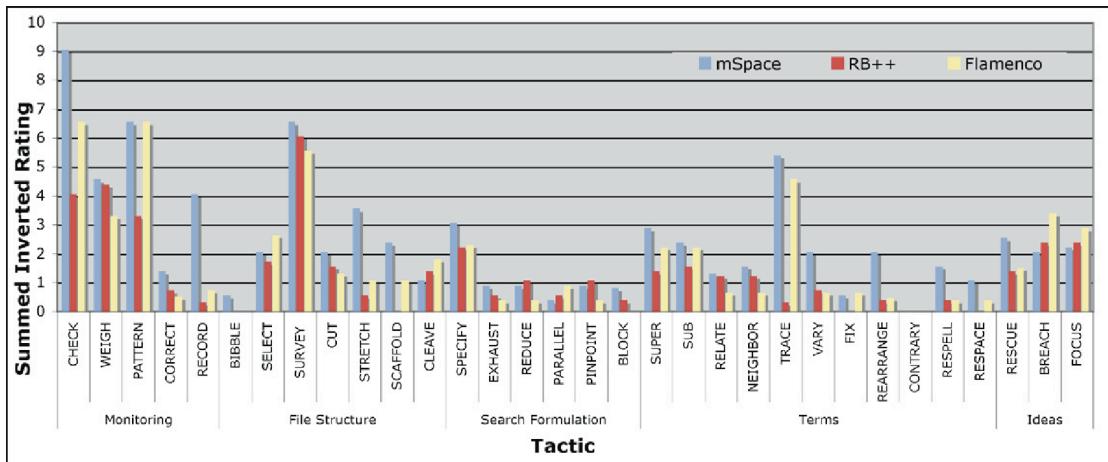


Figure 6: Graph showing the support provided by the three test interfaces for each tactic (in the groups defined by Bates), where taller bars indicate stronger support.

The first tactic, CHECK, has different levels of support in all three interfaces: this tactic is to see what actions have been made in order 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 one’s 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 for Flamenco in Figure 6, over RB++.

The large differences in the score assigned to the support for the RECORD tactic suggest 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 value from any facet. Although any state reached in Flamenco and mSpace can be saved using the parent application⁷, and pages displaying Target Objects in all three interfaces can be saved in this way, a double-click move can store facet items in the Interest panel of the mSpace browser at any point in time. mSpace users may also drag and drop facet-values into the collection space.

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 brings up not only information about its Target Object, but can also be used to see its position in the facets. Users may recover a path used to find items in the Interest panel by dragging it onto the facet browser panel or double-clicking the item, displaying a quick jump to a previous path.

It may be noted that, 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 might be to find an antonym of a selection (rather than everything but a selection). According to the support that was found, mSpace is specifically higher over all of the Term Tactics (SUPER to RESPACE). These higher ratings, however, are provided simply by the increased combination of features, rather than by the better design of any particular feature. 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 facet-columns so that a selection is placed higher up the temporary hierarchy. 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. In more easily supporting SUPER, however, the RELATE and NEIGHBOR tactics are poorly supported in Flamenco because of the aforementioned four step process to change a selection. REARRANGE is well supported by mSpace due to the ease in reordering facets. Finally, tactics like RESPELL are also 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 poorly 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.

5.2 Support for Features

Figure 7 shows the significant contribution of different interface features. Certain elements of the previous discussion can be seen here clearly. Flamenco's four steps to change a selection are reflected in the slight drop of their bar. It may also be noted that Flamenco has no preview cue, and thus the bar is absent from the graph. The ease of multiple selections in RB++ is also

⁷ Usually a Web Browser such as Firefox or IE

clearly shown. One feature to compare is 'View Item'. RB++ has a significant drop in support here, as the implementation has a significant separation between Target Objects and Browser. Target Object Pages may be simply launched from the browser, but there are no ways in which the user can interact with the browser when viewing them. The only option is to return to the browser. In Flamenco and mSpace, users can make further selections from the Target Object page that force automatic interactions with the facets: this is most direct in mSpace where the facets are always present.

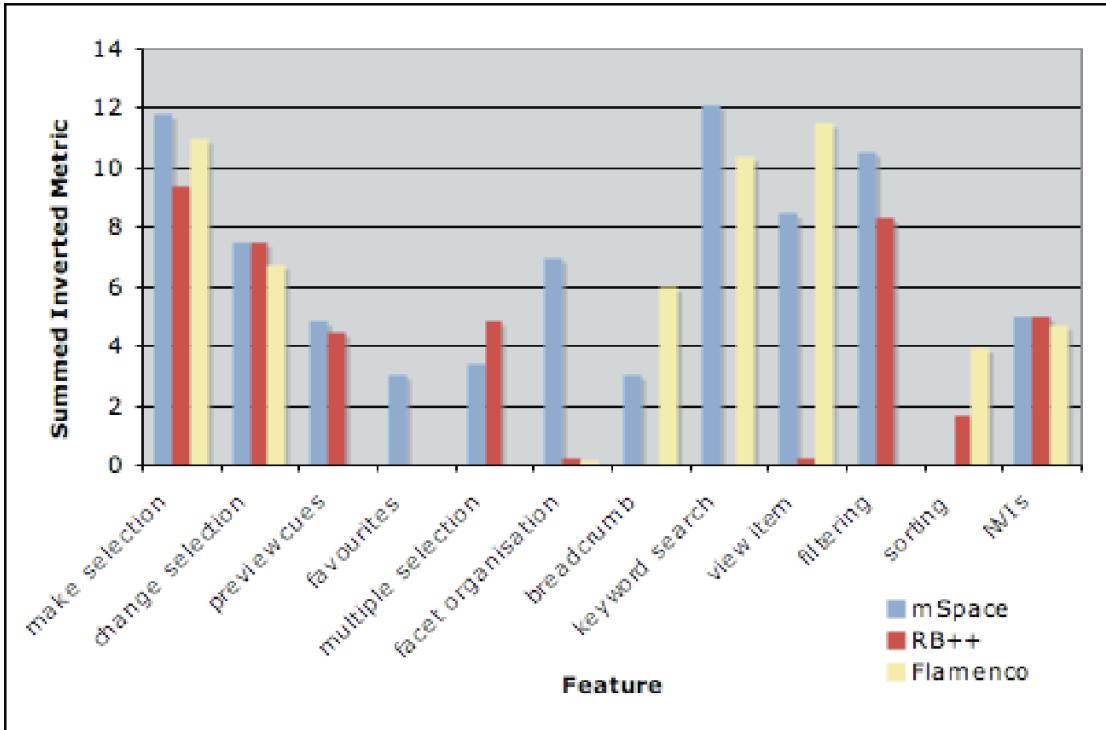


Figure 7: Graph showing the support provided by the three test interfaces by each of their design features, where taller bars indicate stronger support by the feature.

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

5.3 Supporting User Conditions

One question that was motivated the work described in this article is: under what user conditions does a browser provide good support. By attributing each “tactic” to support one of Belkin’s four dimensions of search, such as supporting learn (*Goal*) or meta-information (*Resource*), the support for each ISS condition can be quantified using the Summed Inverted Metrics as before: the assignment of tactics to ISS conditions is shown in Table 5.

The pattern that is seen almost identically for each interface in Figure 8 is indicative of the mapping between Bates’ tactics and the pattern of ISS conditions defined by Belkin and colleagues. Predictably, as was shown in Figure 6, there are three distinct lines, showing that mSpace provides the widest support for search. This height difference, which represents the

average support of each tactic attributed to an ISS condition, does not show us new information. Instead what should be drawn from the graph is hidden within this pattern and shown in the differences in peaks and troughs for each interface condition. Quite clearly the graphs rise and fall in alternating pairs. This represents the alternation between recognise and specify (*Mode*) and is perhaps a predictable outcome for faceted browsers. By including more lessons learnt from the information seeking work on keyword search, such as relevance feedback, we might see a balance between these two conditions. Within each of these alternating pairs, the mSpace bars marginally increases where the others fall. This indicates an increased support for meta-information (*Resource*).

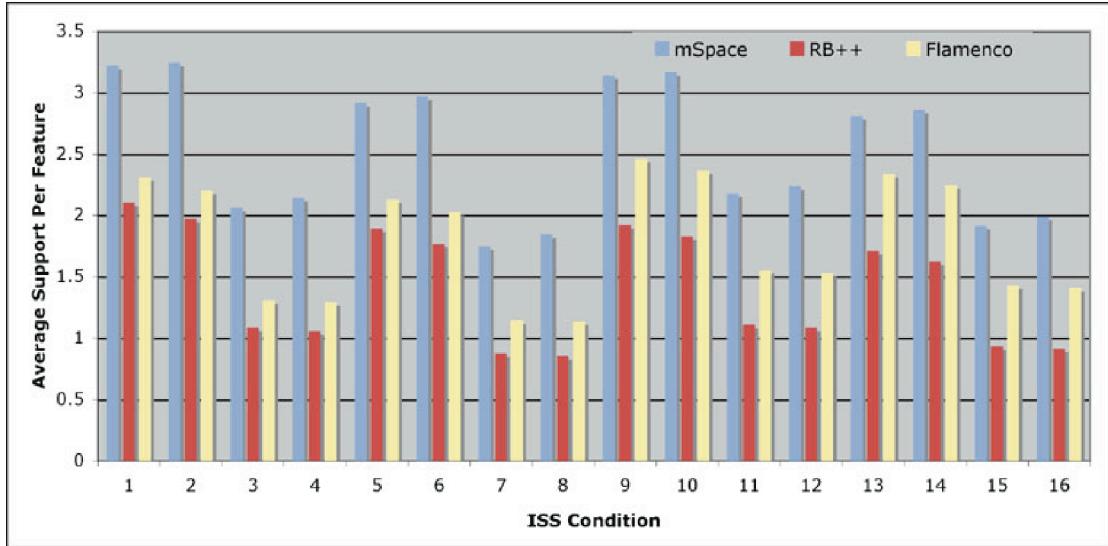


Figure 8: Graph showing the support provided by the three test interfaces for each type of user condition, peaks show greater support for that type of user.

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

The final pattern we draw from Figure 8 is shown every fourth condition 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.

6. Discussion

We begin our discussion by summarising the strengths and weaknesses of each browser, according to the analyses presented above.

Flamenco's strengths lie notably in its goal oriented design. The focus on the display and sorting of results, along with the prioritisation unused facets means that it supports best the types of users that are primarily ready to specify their needs when searching for a particular

item (Figure 8). This is strengthened further by the tendency for Flamenco to have higher bars in the tactics that help narrow search results in Figure 6. Flamenco, however, has notable absences in Figure 7, showing that its support could be greatly improved by providing functionality such as multiple selection and preview cues.

mSpace provides the broadest range of functionality of the interfaces, and this is reflected simply in the often higher bars shown in all three visual analyses. Its strengths, however, lie in supporting users who are learning, rather than seeking specific information. This support for learning is further shown by the high bars for features such as organising facet, filtering, and easily changing selections (Figure 7), and the uniquely strong support for some tactics, such as RECORD and BIBBLE (Figure 6). The browser notably provides no means of sorting or manipulating results, which provides support for multiple tactics in the other two browsers.

RB++ shows particular support for users who are going to recognise relevant information when they see it (Figure 8). That is, RB++ can be used to quickly apply multiple constraints, but is less goal-focused, and supports users in accessing the breadth of relevant information. This is conveyed by the almost even balance that the interface provides for tactics that expand, narrow, translate, and re-focus the results (Figure 6). The browser does suffer in all three graphs, given the absence of the familiar and powerful keyword search function, and also the ability to view Target Objects in the browser itself (Figure 7).

6.1 Implications for Design

There is a difference between unintentional and intentional design, where unintentional refers to designs that have not yet received study, or had available development resources, so that the full extent of its functional capabilities have been realised. For example, keyword search is clearly stronger in mSpace than Flamenco, but both could be easily improved according to existing research. Instead the development of both browsers, and indeed RB++, has focused on alternatives to keyword search, rather than competing over its best implementation. Consequently, we simply acknowledge here that there is value in adding known enhancements to features such as keyword search, as is demonstrated by the difference in support for users identified, by the framework, between the mSpace and Flamenco browsers. One output of our framework is that features are systematically critiqued to expose how the functionality of features may be improved. Other features may be intrinsic to a design: Facet Organisation in mSpace is one of these. Facet organisation has little use, but is possible, in both RB++ and Flamenco and is mainly used to bring popular facets to the forefront. The ability to order facet-columns in mSpace, however, supports a number of different Tactics. Multiple selections are also supported by both mSpace and RB++, but not supported in the Flamenco design. Instead, Flamenco has purposely supported faster selections towards a Target Object and adding multiple selection would slow this down. In order to provide this feature, making a normal selection would require more “moves”.

The framework’s analyses help provide a number of insights into the amount of support for different information seeking tactics provided by different types of browsers. The detail of the graphs produced also allows evaluators to begin to understand which parts of the interface have provided support for the different tactics. To go beyond simply knowing if a tactic is supported or not, the measure used provides an estimate as to how well each tactic is supported by each part of the interfaces. Taken from these analyses, the findings of our study emphasize the following key points that should be considered when designing a faceted-browsing system:

- **Maintain keyword search:** It is clear from Figure 7 that keyword search should still be integrated into faceted browsers to support users in both methods. Including the enhancements proposed by information-seeking and retrieval research, such as relevance feedback, can optimise these. This is backed up by recent work that shows both keyword search and faceted meta-data are used almost evenly when co-presented in a single interface (Wilson & schraefel, 2008a).

- Think about different users: While supporting a user in selection making is important for users who are confident of their target (focused search), optimising the ability to change selections and make multiple selections is important for users who are searching for a potentially relevant but unknown object (exploratory search). Figure 8 shows just how different the support provided by a single interface can be for different types of users.
- Consider meta-data relationships: We suggest that representing the temporary hierarchies, created by making selections in facets, to the user is important for keeping track of user actions and understanding the effects of facets on each other. Although facets can be used in any order, the spatial ordering of them in mSpace has been shown to support a number of search tactics by exposing the relationships between different meta-data, as well as between the meta-data and the user (Figure 6). This consistency of display is discussed further by Wilson and schraefel (2008b).
- Show information in context: The interoperability of viewing information pages with the browsing of facets is a key element in maintaining the search context. mSpace is a good example of this, supporting a number of different tactics with its layout.
- Include sorting and filtering: Notably mSpace does not support any grouping, filtering, or sorting of results. Figure 6 shows, however, that the ability for the user to arrange data and results so that they can effectively find the information they want supports a number of key tactics in the other two browsers.
- Facilitate information gathering: The collection of information during search is important (the RECORD tactic), especially when users are trying to locate relevant information rather than specific information (schraefel *et al.*, 2002).
- Offer previews: Previewing the effects of actions is important for making decisions in search (such as the WEIGH tactic) and should be shown as soon as possible (schraefel *et al.*, 2003; Zhang & Marchionini, 2005). Figure 6 clearly shows that the preview feature of certain interfaces provides a significant amount of support for various tactics.
- Support logical opposites: One challenge is to support the CONTRARY tactic, which is rarely supported in browsers (unsupported according to Figure 6) and involves looking for the logical opposite of a term; this is different to selecting the inverse set of values in a facet, as some terms may be synonymous or connected to each other.

Ultimately, it is important to consider the varying intentions and needs of different users. Designers should try to ensure that a broad range of tactics are supported, in case they are required by various user types, and that the support is easily accessible. Through the adoption of these design recommendations, systems can take a step towards this goal and improve the search experience for all users.

7. Future Work

7.1 Framework Validation

Through our evaluation framework we create a performance benchmark that is essential to drive advancements in search system development. The formative approach above uses established models from the IR community. While this evaluation framework has shown to produce some useful and provocative insights for improving design, in order for this measure to be accepted in the design of information systems, the results must be corroborated against existing measures. Consequently, we intend to further validate this combined application of models with the results of our own, and other researchers', user studies. This task is not trivial. Typical user studies of information systems involve creating goal- based scenarios and measuring results such as task completion time, result accuracy and user preference (Borlund, 2003). Yet this measure is designed to evaluate both focused and exploratory search environments that support a wider sense of human information behaviour, such as when a

user finds information that turns out to be irrelevant, partially relevant, or even contrary to achieving their goal (Godbold, 2006). Such broad information behaviour means that typical measures such as time performance or result accuracy may not be valid, as they assume that the user has a goal and knows an accurate result. Consequently, typical pre- and post-evaluation validation methods (Gong & Kieras, 1994) also cannot be applied, as our evaluation framework does not claim to improve aspects such as time performance.

7.2 Framework Extension

There are many interesting challenges that extend from our research and we look forward to investigating them in collaboration with the research community. Our framework naturally pushes interface designers to develop prototypes that provide as many features for use with as few moves as possible. However, this can lead to very busy interfaces. Consequently we intend to measure the effect of busy interfaces on users through work such as Cognitive Load Theory (Chandler & Sweller, 1996; Paas *et al.*, 2003). Prior work in this area has shown that cognitive load can have a significant influence on information seeking behaviour, in particular query reformulation (Beaulieu, 1997; Bruza *et al.*, 2000). Another approach may be to more formally model flow and change within an interface (Yuan & Belkin, 2007), so that appropriate support is provided as the user progresses with their search.

Formalising Bates’ “stratagems” may be an important goal for the future, and may help us define a mapping between the two models used in the framework. One approach may be to define stratagem level activities such as comparison, aggregation, and synthesis (Marchionini, 2006), which may more formally tie together the link between Bates’ “tactics” and Belkin’s ISS conditions.

Another interesting question that emerged from the analysis, is whether all of Bates’ tactics are equally important to a given user. For example, Flamenco prioritises the making of new decisions, as it removes facets that have been used and foregrounds unused facets. Alternatively, mSpace and RB++ provide consistent layouts of facets, leaving the users’ previous choices, and the items un-chosen alternatives, in full view so that they can be checked at any time. If a knowledgeable user makes new decisions more often than they check their old decisions then an interface that prioritises new decision-making perhaps scores higher. Realistically, the importance or relevance of tactics may also vary depending on each of the ISS conditions in the episodic model. For example, users who know what they are looking for value making new decisions more than checking their old decisions, but users who do not know what they are looking for may often look to change their old decisions in order to achieve their goal. Instead of attributing tactics to different dimensions in the episodic model, different weightings could be applied to each tactic depending on the sixteen ISS conditions. Further work is needed to investigate any static or varying importance associated with each tactic. Combining any varying importance with a measure of cognitive overload would provide a very strong metric for agreeing on the best trade-off of features in an advanced search system.

8. Conclusions

This article makes three contributions. First, we have designed an evaluation framework that combines existing research models in a way that can estimate the user-focussed strengths and weaknesses of IR browsers. Second, we have presented the application of this framework to evaluate three faceted browsers. Although these models have been designed to encompass elements of user search, applied in combination it is conceivable that they can be used to identify the strengths and weakness of an advanced search interface such as a faceted browser. Third, by applying this evaluation to three interfaces we have then been able to estimate the size of these strengths and weaknesses over: the support for tactics, the support provided by interface features, and the support for sixteen unique user conditions.

Both moves and well-defined tactics, from the model of strategic search interaction defined by Bates (1990), have been used to estimate the support for each tactic provided by the features of three faceted browsers: mSpace, Flamenco and RB++. These metrics have first been summed by tactic to show which of Bates' tactics are particularly supported by a browser. Second, by summing the metrics by feature, we can show the support provided by its implementation. Identifying weak or even missing features can promote changes and advances in implementation to support more tactics or reduce the moves required to achieve each tactic. Finally, by summarising and normalising these metrics into an episodic model of Information-Seeking Strategies (Belkin *et al.*, 1995), we have identified particular strengths and weaknesses of the three faceted browsers for users in different search conditions. It is clear that our framework has the potential to be a useful design tool in the development of advanced search interfaces. This work also opens some interesting research questions, which we look forward to investigating in the future.

9. References

- 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.
- Beaulieu, M. (1997). Experiments on interfaces to support query expansion. *Journal of Documentation*, 53(1), 8-19.
- Belkin, N. J. (2000). Helping people find what they don't know. *Communications of the ACM*, 43(8), 58-61.
- Belkin, N. J., Cool, C., Stein, A., & 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., & 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., & Brooks, H. M. (1982). Ask for information retrieval: parts I and II. *Journal of Documentation*, 38(2), 3.
- Borlund, P. (2003). The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3), Paper 152.
- Bruza, P., McArthur, R., & Dennis, S. (2000). Interactive Internet search: keyword, directory and query reformulation mechanisms compared. In *Proceedings of the 23rd Annual International Conference on Research and Development in Information Retrieval (ACM SIGIR'00)*, 280-2871.
- Card, S. K., Moran, T. P., & Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM*, 23(7), 396-410.
- Chandler, P., & Sweller, J. (1996). Cognitive load while learning to use a computer program. *Applied cognitive psychology*, 10(2), 151-170.

- Chi, E. H., Pirolli, P., Chen, K., & Pitkow, J. (2001). Using information scent to model user information needs and actions and the web. In Proceedings of the ACM SIGCHI Conference on Human factors in Computing Systems (CHI'01), 490-497l.
- Cleverdon, C. W., Mills, J., & Keen, M. (1966). Factors determining the performance of indexing systems: Aslib Cranfield Research Project, Cranfield, England.
- Cool, C., & Belkin, N. J. (2002). A classification of interactions with information. In Proceedings of the 4th International Conference on Conceptions of Library and Information Science (CoLIS4), 1-15l.
- Dumais, S. T., & Belkin, N. J. (2005). The TREC interactive tracks: Putting the user into search. In E. Voorhees & D. Harman (Eds.), *TREC: Experiment and Evaluation in Information Retrieval* (pp. 123-153): MIT Press.
- Fu, W., & Pirolli, P. (2007). SNIF-ACT: a cognitive model of user navigation on the World Wide Web. *Human-Computer Interaction*, 22(4), 355-412.
- Godbolt, N. (2006). Beyond information seeking: towards a general model of information behaviour. *Information Research*, 11(4), Paper 269.
- Gong, R., & Kieras, D. (1994). A validation of the GOMS model methodology in the development of a specialized, commercial software application. In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'94), 351-357l.
- Harman, D. K. (1997). The TREC conferences. Morgan Kaufmann Multimedia Information And Systems Series, 247-256.
- 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.
- Hildebrand, M., Ossenbruggen, J. v., & Hardman, L. (2006). /facet: a browser for heterogeneous semantic web repositories. In Proceedings of the 5th International Conference on the Semantic Web (ISWC'06), 272-285l.
- Huvila, I., & Widén-Wulff, G. (2006). Perspectives to the classification of information interactions: the Cool and Belkin faceted classification scheme under scrutiny. In Proceedings of the 1st international conference on Information interaction in context (IIIX'06), 144-152l.
- Lagergren, E., & Over, P. (1998). Comparing interactive information retrieval systems across sites: The TREC-6 interactive track matrix experiment. In Proceedings of the 21st Annual International Conference on Research and Development in Information Retrieval (ACM SIGIR'98), 164-172l.
- Liu, B., & Oard, D. W. (2006). One-sided measures for evaluating ranked retrieval effectiveness with spontaneous conversational speech. In Proceedings of the 29th Annual International Conference on Research and Development in Information Retrieval (ACM SIGIR'06), 673-674l.
- Marchionini, G. (2006). Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4), 41-46.
- Paas, F., Renkl, A., & Sweller, J. (2003). Cognitive Load Theory and Instructional Design: Recent Developments. *Educational Psychologist*, 38(1), 1-4.
- Pharo, N. (1999). Web information search strategies: A model for classifying web interaction. In Proceedings of the 3rd International Conference on the Conceptions of the Library and Information Science (CoLIS3), 207–218l.
- Robertson, S. E., & Hancock-Beaulieu, M. M. (1992). On the evaluation of IR systems. *Information Processing and Management*, 28(4), 457-466.

- Rocchio, J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART Retrieval System: Experiments in Automatic Document Processing* (pp. 313-323): Prentice Hall.
- Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4), 288-297.
- Saracevic, T. (1997). The stratified model of information retrieval interaction: extension and applications. In Proceedings of the Annual Meeting of the American Society for Information Science (ASIS'97), 313-327l.
- schraefel, m. c., Karam, M., & Zhao, S. (2003). Listen to the Music: Audio Preview Cues for the Exploration of Online Music. In Interact, 192--199l.
- schraefel, m. c., Wilson, M. L., Russell, A., & 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., & Zhao, S. (2002). Hunter gatherer: interaction support for the creation and management of within-web-page collections. In Proceedings of the 11th International Conference on World Wide Web (WWW'02), 172-181l.
- Shneiderman, B. (1981). Direct manipulation: A step beyond programming languages.
- Spink, A., Greisdorf, H., & Bateman, J. (1998). From highly relevant to not relevant: examining different regions of relevance. *Information Processing and Management*, 34(5), 599-621.
- Taylor, R. S. (1968). Question-negotiation and information-seeking in libraries (Vol. 29): College and Research Libraries.
- White, R. W., & Drucker, S. M. (2007). Investigating behavioral variability in web search. In Proceedings of the 16th international conference on World Wide Web (WWW'07), 21-30l.
- White, R. W., Kules, B., Drucker, S. M., & schraefel, m. c. (2006). Introduction. *Communications of the ACM*, 49(4), 36-39.
- White, R. W., Ruthven, I., Jose, J. M., & Rijsbergen, C. J. V. (2005). Evaluating implicit feedback models using searcher simulations. *ACM Transactions on Information Systems*, 23(3), 325-361.
- Wilson, M. L., & schraefel, m. c. (2006). mSpace: what do numbers and totals mean in a flexible semantic browser. In Proceedings of the 3rd International Semantic Web User Interaction Workshop (SWUI'06),
- Wilson, M. L., & schraefel, m. c. (2008a). A longitudinal study of exploratory and keyword search. In ACM/IEEE-CS Joint Conference on Digital Libaries (JCDL'08), 52-56l.
- Wilson, M. L., & schraefel, m. c. (2008b). Improving Exploratory Search Interfaces: Adding Value or Information Overload? In Second International Workshop on Human-Computer Interaction and Information Retrieval (HCIR'08), 81-84l.
- Yee, K.-P., Swearingen, K., Li, K., & Hearst, M. (2003). Faceted metadata for image search and browsing. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'03), 401-408l.
- Yuan, X., & Belkin, N. J. (2007). Supporting multiple information-seeking strategies in a single system framework. In Proceedings of the 30th annual international conference on Research and development in information retrieval (ACM SIGIR'07), 247-254l.

Zamir, O., Etzioni, O., Madani, O., & Karp, R. M. (1997). Fast and intuitive clustering of web documents. In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD'97), 287–290.

Zhang, J., & Marchionini, G. (2005). Evaluation and evolution of a browse and search interface: relation browser. In Proceedings of the 2005 National Conference on Digital Government Research (DG.O'05), 179-188.