

# Evaluating Collaborative Information Seeking Interfaces with a Search-Oriented Inspection Method and Re-framed Information Seeking Theory

Max L. Wilson\*, m.c. schraefel

Intelligence, Agents, and Multimedia Research Group,  
University of Southampton,  
Southampton, SO17 1BJ, UK

## Abstract

Despite the many implicit references to the social contexts of search within Information Seeking and Retrieval research, there has been relatively little work that has specifically investigated the additional requirements for collaborative information seeking interfaces. Here, we re-assess a recent analytical inspection framework, designed for individual information seeking, and then apply it to evaluate a recent collaborative information seeking interface: SearchTogether. The framework was built upon two models of solitary information seeking, and so as part of the re-assessment we first re-frame the models for collaborative contexts. We re-frame a model of search tactics, providing revised definitions that consider known collaborators. We then re-frame a model of user profiles to analyse support for different group dynamics. After presenting an analysis of SearchTogether, we reflect on its accuracy, showing that the framework identified 8 known truths, 8 new insights, and no known-to-be-untrue insights into the design. We conclude that the framework a) *can* still be applied to collaborative information seeking interfaces; b) can successfully produce additional requirements for collaborative information seeking interfaces; and c) can successfully model different dynamics of collaborating searchers.

## Keywords

Collaborative, search, information, seeking, evaluation, inspection, analytical

## 1. Introduction

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 & 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 & Morris, 2008, Morris & Horvitz, 2007a, Morris & Horvitz, 2007b, 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, we must first reconsider how much of the underlying existing and often well-established, solo-focused, IR and IS theory still applies to Collaborative Information Seeking (CIS) contexts.

In this article, we re-assess a recent search-oriented analytical inspection framework<sup>1</sup> (Wilson, 2008, Wilson & schraefel, 2007, Wilson et al., 2009) in the context of collaborative information seeking. The framework was originally designed to evaluate the plethora of solitary information seeking interfaces, but we demonstrate that it can still be used to evaluate

---

\* Corresponding author: relocated to *Future Interaction Technology Lab, Swansea University, Swansea, SA2 8PP. Tel: +44(0)1792 602611. Fax: +44(0)1792 295708. Email: [m.l.wilson@swansea.ac.uk](mailto:m.l.wilson@swansea.ac.uk)*

<sup>1</sup> <http://mspace.fm/sii> - The Search interface inspector.

CIS interfaces. We do this by first re-framing the underlying IS models used within the framework, and then validating the framework's results by correlating them with existing research. The re-assessment is designed to a) show that the framework 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, by performing this re-assessment we have begun, 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.

In the following sections, we first review the history of IS research, especially highlighting where CIS has been addressed, in order to summarise what is already known about collaborative searching and inform our re-assessment of the IS inspection framework. We then present an overview of the original analytical inspection framework, which was developed to assess the extent by which IS interfaces support different search tactics and different user types. We begin our re-assessment by re-framing two established IS models: one of search tactics (Bates, 1979b, Bates, 1979a), and one of searcher profiles (Belkin et al., 1993). We provide extensive detail of this re-framing, so that these sections alone can be used as a reference when applying our modified inspection framework for CIS interfaces. Further, the re-framing can be used to understand how these two models relate to our developing CIS models and assumptions. In Section 4, we then present an evaluation of a recent freely available CIS interface: SearchTogether (Morris & Horvitz, 2007b), using our modified analytical framework. The results of the evaluation are then further validated by correlating its findings with known usability issues identified by the designers of SearchTogether.

## **2. Related Work**

Although forms of implicit collaboration, such as recommender systems (Resnick & Varian, 1997) and even Google's PageRank (Brin & 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.

### ***2.1 Collaboration in IS-focused literature***

Much of the early work into information seeking 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 information seeking was usually performed in conjunction with librarians. Several models, for example, were based on dialogues or conversations between typical searcher and librarian roles: Conversation for Action (Winograd & Flores, 1986), for example, and the Conversational Roles Model (COR) (Sitter & Stein, 1992, Stein & Thiel, 1993). The result of these searcher/librarian models, however, has usually been to design search interfaces to act as the librarian. Belkin and colleagues (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 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 & 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 with 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).

## ***2.2 Defining CIS***

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 take 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 over one machine.

Another initial strand of CIS research has been to better define what counts as collaboration during search. In 2008, 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 (2009, Pickens & 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 & Varian, 1997), and to some extent Google's PageRank (Brin & 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 & Morris, 2008, Morris & Horvitz, 2007b). Concurrency determines whether users are searching synchronously at the same time (Amershi & Morris, 2008), or asynchronously at different times (Morris & Horvitz, 2007a). Finally, searchers can either search together in one environment (Amershi & Morris, 2008, Morris et al., 2006) or in distributed environments (Morris & Horvitz, 2007b).

## ***2.3 Designing CIS 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. S<sup>3</sup>, standing for Storable, Shareable Search (Morris & 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 & 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 & 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.

## **2.4 Summary**

The key points that can be drawn from this section are that a) there has been a recent flurry of search systems that directly support CIS, b) the community's 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 we present an evaluation that is designed around IS theory, and show how re-framing the original IS models allows it to accurately inspect CIS interfaces.

## **3. Re-framing the analytical evaluation framework and its underlying theory**

Our recent work (Wilson, 2008, Wilson & schraefel, 2007, Wilson et al., 2009) has been to develop and test an evaluation framework that is designed to systematically inspect prototype interfaces in terms of the tactics they allow users to employ and the types of conditions the searchers may be in. The application of the framework can tell evaluators a) which types of users are well or poorly supported, b) which types of tactics are well or poorly supported, and c) how each element of the user interface is contributing to these types of support. The types of search tactics are discussed in more detail below, but examples include broadening a search, checking what has been done, and weighing up options. User types are broken down by dimensions such as their existing knowledge and confidence in finding an answer.

This inspection approach has been validated in several ways and correlates well with the results of user studies (Wilson, 2008). The framework discovered 10 of the 12 key positive and negative findings (or 83%) discovered by an empirical study of faceted browsers performed by Capra and colleagues (2007). Further, the graphs produced provided new insight into the causes of these key findings, helping to explain the observed behaviours. Although this validation was applied to established systems, our inspection method, like most, can also be applied to early prototypes. The results provide evaluators with ideas about a) which implementations might be better for users, b) which tactics they might want to try and support better, and c) which users they might want to support better. Low-level prototypes can thus be redesigned and re-evaluated before investing in development and user evaluation. Further, the richness of results produced means that it can also be used to learn about the strengths of existing systems, such as Google, so that we can better understand what makes it popular and for which types of users.

To perform such a detailed and systematic inspection of prototype designs, the framework relies on information seeking theory. Specifically, the framework uses one model of tactics by Bates (1979b, 1979a), and one model of users by Belkin and colleagues (1993). One contribution of the framework was to provide a connection between the two models that states which tactics are most useful for each user type.

Below, we assess these two established models from IS theory, 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 article 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 3.1 re-frames Bates' model of search tactics (1979b, 1979a), by providing a description of each of the 32 tactics and how they apply in a collaborative context. In Section 3.2, we re-frame the model of user profiles, provided by Belkin and colleagues (1993), from the perspective of different roles taken within groups of collaborative searchers.

### ***3.1 Re-framing Bates' Model of Information Seeking***

Bates (1979b, 1979a) identified 32 different tactics that people may carry out when searching for information across different technologies. Where these were originally designed to model the tactics of individuals, they have different implications for searchers that are part of a collaborating group or team. We now step through these tactics 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 else's 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 to find alternative methods of going through a structured list. In a collaborative setting, alternative methods might include sharing the load with collaborators.

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. We do not discuss these individually here, but they are each mainly solo decisions. They could still benefit, however, from an awareness of others peoples search terms and phrases. TRACE, for example, is the activity of analysing search results in order to discover new potential terms. This discovery of terms may be useful for the group, and one individual's TRACE discoveries may be very different from another team member's queries. Similarly, the VARY tactic, for trying variations of a term, could be split across a team. Further, two searchers may want to compare the results of trying two separate (RE)ARRANGEMENTS.

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 behaviour. 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.

It is clear from these last three tactics that collaborative search interfaces should support the transfer of developed understanding to other members of a team, as they may significantly alter the direction of the whole group.

Most of the discussion of tactics above could be generalized to the need to either actively share results or passively monitor the progress of others. In the evaluation framework, however, each feature of an interface, such as the keyword search form, the list of results, the communication channels, and so on, are addressed individually in terms of how they support each tactic. This process, therefore, means that the evaluator is encouraged to think about how the keyword search box, for example, can also be used to indicate to the user that a search has been carried out before by another group member. Consequently, it leads to a system where

support of tactics is pervasive to the whole interface rather than having specific functions or features of the design that specifically support individual tactics. This should become clearer as we discuss a specific example in Section 4.

### 3.2 Re-framing Belkin's model of users

Searchers engaged in collaborative shared tasks often naturally fall into different roles (Golovchinsky et al., 2009, Morris & Teevan, 2008), where the resulting dynamic is influenced by their individual skills. Belkin and colleagues (1993) identified 16 different types of searchers, shown in Table 1, based on unique combinations produced by four binary dimensions: Method, Goal, Mode, and Resource. Below we discuss how different group dynamics and roles map to these user types, but first we describe them in more detail. The Method dimension defines whether a user is scanning for a possible resource that may exist, or searching for a specific resource that they know does exist. The Goal dimension defines whether a user is learning about the domain, or whether they are trying to select a resource for use. The Mode dimension defines whether the user will have to recognise a resource, or if they know how to specify its details. Finally the Resource dimension, defines whether the user is looking for a resource, or metadata about a resource, such as its author. Variations of this original model have been suggested (Huvila & Widén-Wulff, 2006), and more recently Kim (2009) released a revised version for web searching. These are being considered for future versions of the framework, but switching to the latter, for example, would mean that the framework might not accurately evaluate offline search systems. Both revisions are based on the core version shown in Table 1.

**Table 1: 16 unique searcher types, identified by Belkin and colleagues (1993).**

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

In a group setting, a searcher's role may be dependant on their existing knowledge and search experience (Golovchinsky et al., 2009), and here we suggest 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 also discusses some additional CIS roles, however these include more general information behaviour roles, including those not actively involved in CIS activities.

The easiest relationships to define using Belkin's 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, but potentially ISS1, if they consider some example resources as well as metadata. They may also occasionally stretch to ISS3 and ISS4 if 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 Belkin's ISSs for CIS research. This, and the re-framing of Bates' tactics, will be used to discuss the results of the example evaluation below.

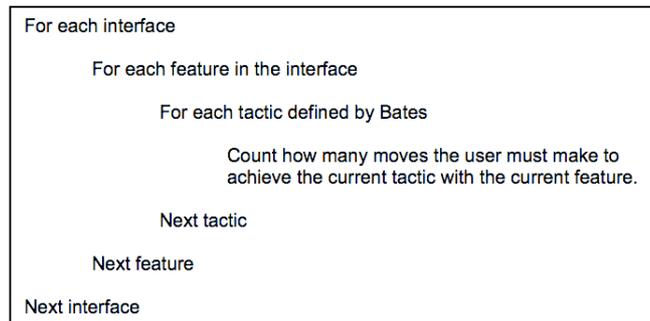
#### **4. Evaluating an example Collaborative Information Seeking interface using the framework and re-framed models**

In this section we evaluate a freely available CIS interface called SearchTogether (Morris & Horvitz, 2007b), described below. SearchTogether makes a good clear example, as it designed for explicit, distributed, collaborative information seeking. Below we first describe the process of using the framework, considering the re-framed models above, and then describe the results found when applying it to SearchTogether.

##### **4.1 Method**

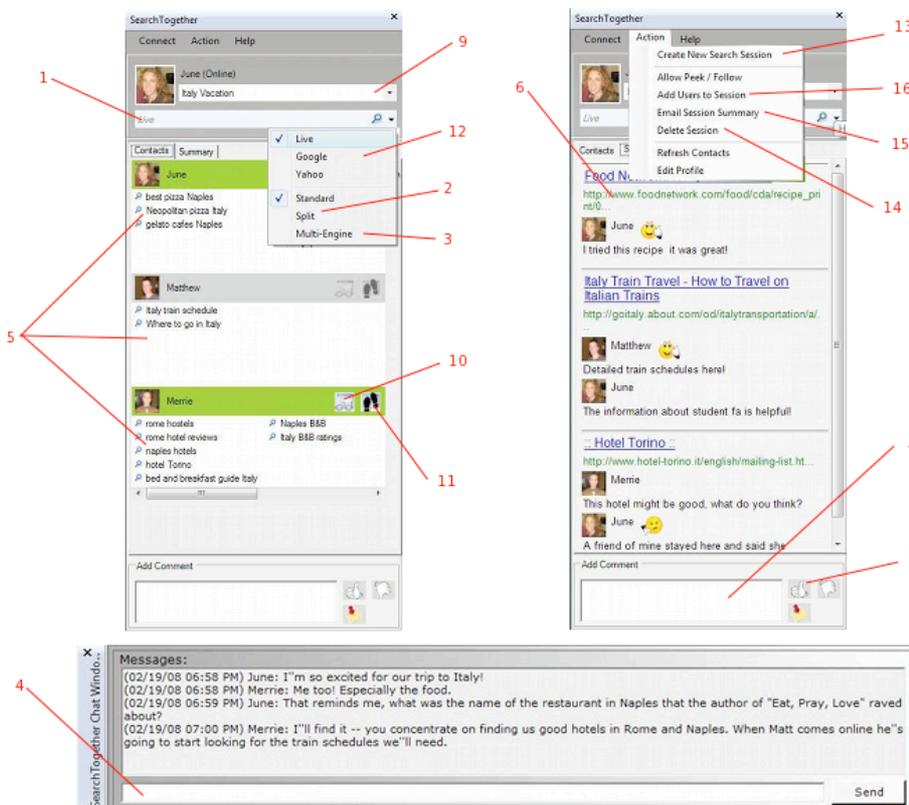
The method of applying the framework, shown in Figure 1, remains unchanged, but instead of using the 32 solitary definitions of Bates' tactics, we use the re-framed definitions provided above. Put simply, the evaluator, or team of evaluators, considers how every aspect of every interface in the evaluation can be used to perform each of the 32 tactics. In this case, there is only one interface being evaluated: SearchTogether, and so consequently, we are more simply considering how every element of the SearchTogether interface (listed in Table 2 and shown in Figure 2) supports each of the 32 tactics. As is clear from Figure 1, the evaluators quantify this support for tactics provided by interface elements, by counting the number of 'Moves'. The term 'Moves' used here, is defined by Bates (1979b, 1979a) as being a set of *mental* and/or *physical* actions. These moves are larger than the keystroke level model (Card et al., 1980), for example, and include things like choosing a keyword term (one mental move), entering the keyword into the search box (one physical move), and pressing the search button (a second physical move). Optional moves, including optionally repeated moves, are not counted. We cannot tell, for example, how many terms a user would enter into a keyword

search box, nor whether the user will have to scroll to find their answer. These two examples are not counted, but the entering of a search term would be, as would the scanning of the search results.



**Figure 1: The process of applying the analytical inspection framework. The main counting step is carried out with the revised definition of Bates' tactics.**

The process of applying the framework is similar to that of a Cognitive Walkthrough, in that an expert, or for more rigor, a group of experts, consider each element of the search interface in turn, and walkthrough the 'Moves' required to use it to perform each of the 32 tactics. In Cognitive Walkthrough evaluations, groups of experts step through a scenario of use, discussing how clear it will be for the user to use it. There is nothing to stop evaluators from asking the Cognitive Walkthrough questions at the same time as using the framework described here, as each tactic represents a potential scenario of use.



**Figure 2: Design of the CIS interface Search Together, where callout identifiers are described in Table 2.**

**Table 2: The 16 interface elements identified in the evaluation of SearchTogether, where the callout identifiers match those included in Figure 2.**

<b>Fig. 2 Callout</b>	<b>Interface Element</b>	<b>Description</b>
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.	Chatting 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.

## 4.2 Results

The method described above was applied, using the CIS-oriented descriptions of Bates' 32 tactics, by the developers of the framework. The full detail of the results, including the number of moves counted for applying each tactic with each interface element, is available online<sup>2</sup>. Here we present the graphs, which are interactive online, and present the findings conveyed by these graphs.

The first graph, shown in Figure 3, 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 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 4).

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 tactic

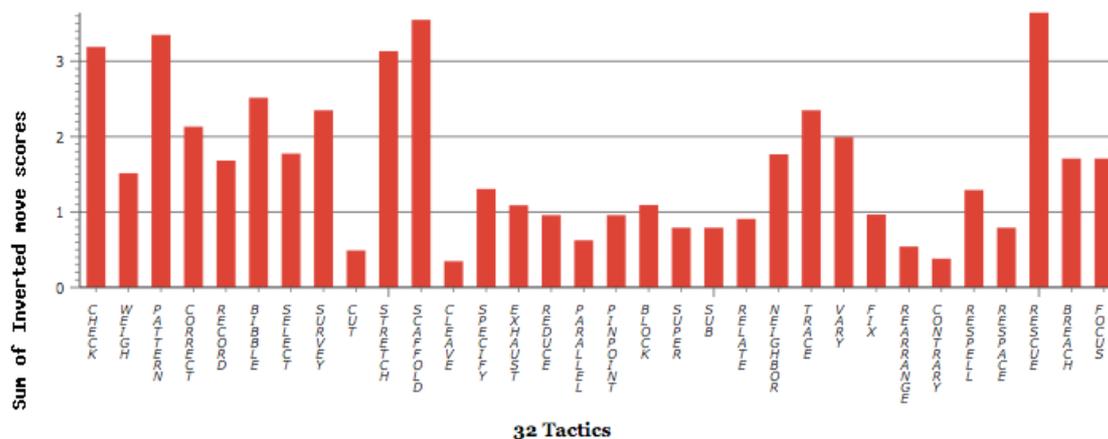
<sup>2</sup> <http://mspace.fm/sii/results.php?pid=000010>

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<sup>3</sup>, 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 4, 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.

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 Figure 4). 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 Figure 3, where the multi-engine search and split search should make it much easier for users to EXHAUST the potential results.



**Figure 3: Graph showing the support provided for each tactic by the SearchTogether interface, where taller bars represent better support.**

Looking further at Figure 4, we see that the summary page provides the third highest amount of support across the SearchTogether interface. This has been discussed to some extent already, as it allows users to spark new ideas for search by monitoring what has

<sup>3</sup> See other examples online <http://mspace.fm/sii>.

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.

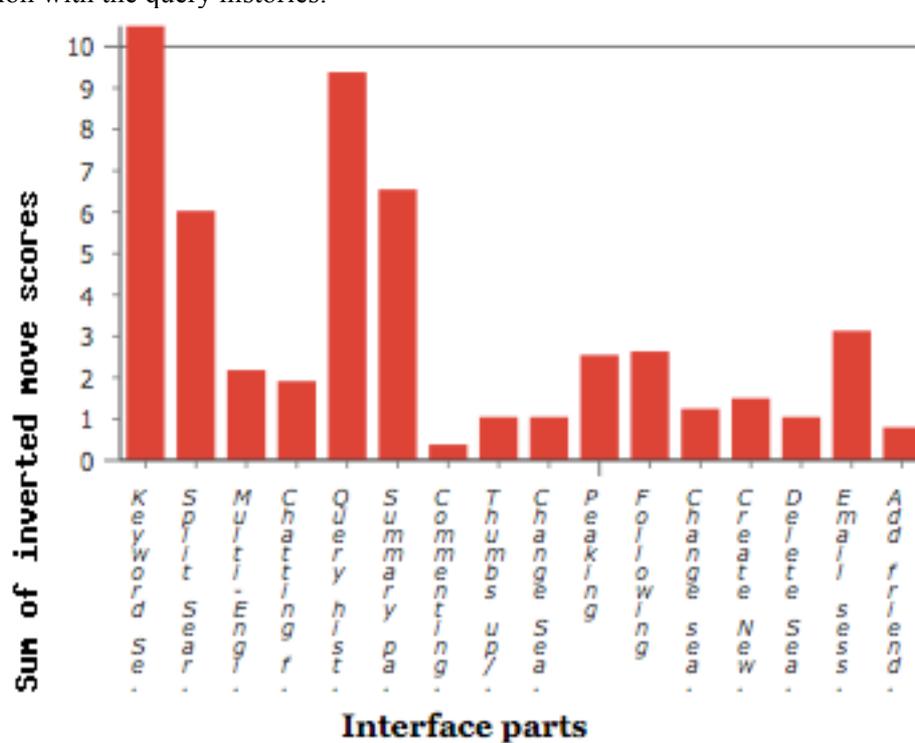
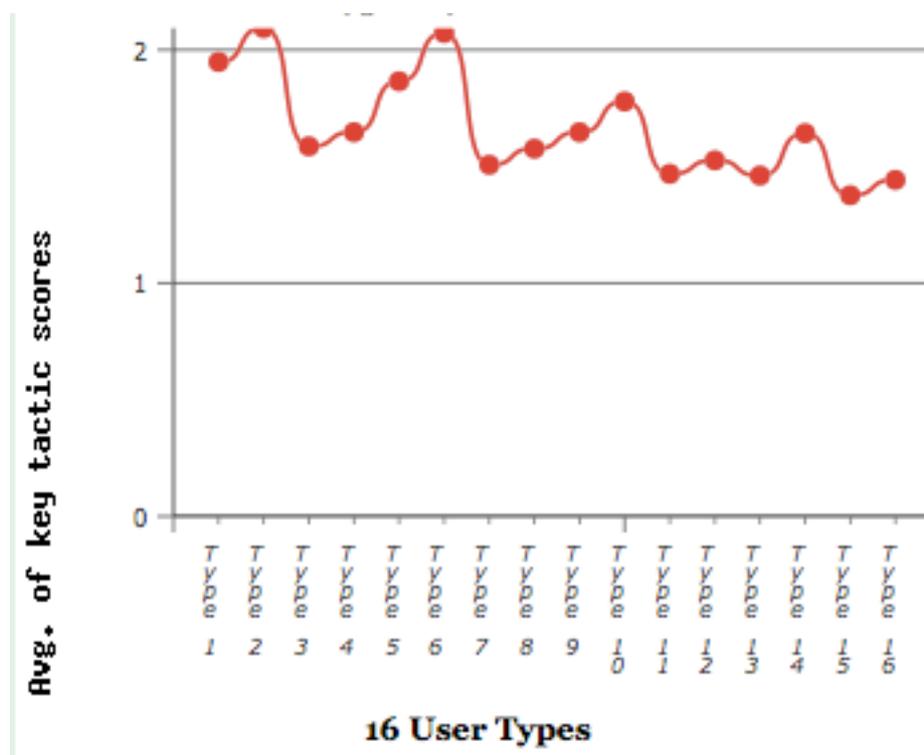


Figure 4: Graph showing the support provided by each element of the SearchTogether, across all tactics, where taller bars represent better support.

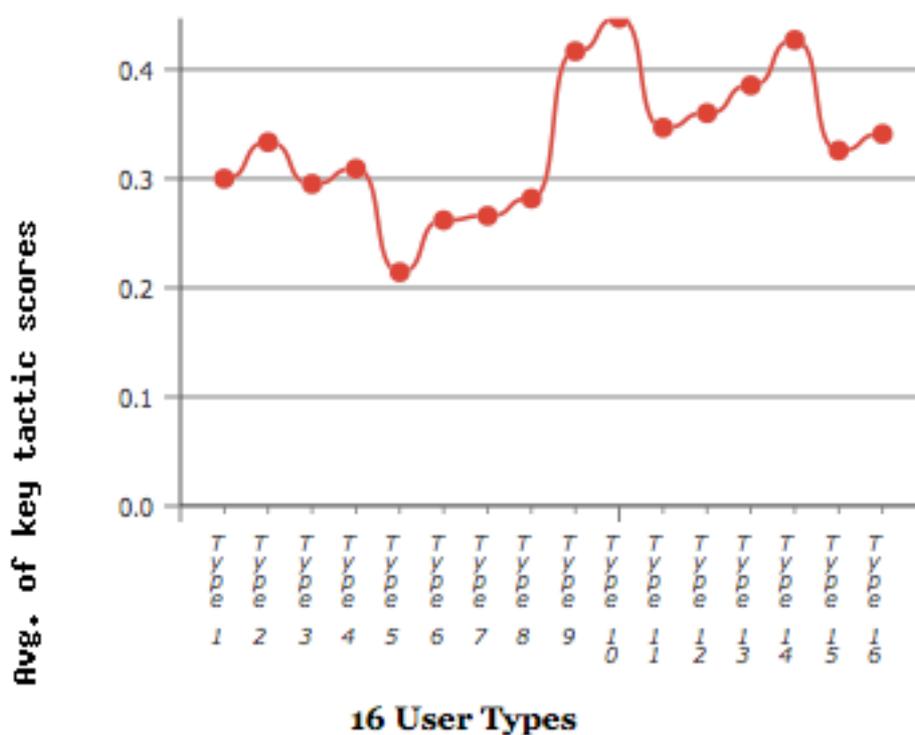
Using Table 1 and Figure 5 together, we can see that because the left hand side is higher, on average, than the right hand side, the design of SearchTogether provides a larger amount of support for users who are *scanning*, than for users who are able to *specify* what they need. This is supported mainly 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 searches 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 6), 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 and Figure 6), 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.

In terms of the CIS roles discussed in Section 3.2, 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 6), 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.



**Figure 5: Graph showing the support provided for each of Belkin's 16 user types by the SearchTogether interface, where higher points represent better support. Read in conjunction with Table 1, which describes the user types (where each type relates to an equivalent ISS number).**



**Figure 6: Graph showing the support provided for each of Belkin's 16 user types by SearchTogether's Basic Keyword Search function alone. Read in conjunction with Table 1, which describes the user types (where each type relates to an equivalent ISS number).**

## 5. Re-assessing the evaluation framework

In Section 4, we presented an analysis of SearchTogether, using a modified version of an existing analytical inspection framework and the re-framed underlying models presented in Section 3. We now re-assess the framework's applicability to CIS in light of results produced.

The aim of the analytical 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 presented in Section 4, 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 variation of the graph shown in Figure 4.

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. Again, design changes can be tested to see how example group dynamics are affected in the types of searchers shown in Figure 5. One suggested view 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 & Morris, 2008), described briefly in Section 2.3. 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.

The final concern in this re-assessment is that the framework maintains the accuracy demonstrated in previous validations with solitary search interfaces (Wilson, 2008). In line

with these previous validations, we sought to correlate the results above with the findings of user studies. Where in previous validations, however, we directly compared results with the findings described in publications, here we chose to communicate 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 & 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, we aimed to know: 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.

To understand our first aim, we provided SearchTogether's designers with a table of search-oriented usability statements, both positive and negative, produced by our analysis. The designers were asked to then classify these results, as shown in Table 3, 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 3, the two latter untrue classifications, with either empirical or qualitative evidence, were not used by the designers of SearchTogether. Consequently, we can first conclude that none of our results were known to be false. One statement (12), 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 keep watch for any future supporting or contradicting evidence. Similarly, one statement (1) was not known either way. The analytical framework 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 & Morris, 2009). CoSense uses data from SearchTogether to provide an interface for making sense of an overall CIS session.

We can conclude from these results that the designers of SearchTogether have evidence from user studies that support a half of the statements produced by our analysis (8/16). The associated conclusion, however, is that the remaining half of the statements represent novel insights, for which there is no evidence against. 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.

We believe the combination of evidentially supported statements and still unproven insights provide strong support of the depth of analysis provided by our analytical framework. The results are further strengthened by the amount of findings that can be identified compared to a series of user studies.

**Table 3: 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.**

<b>Key findings about SearchTogether produced by the analysis from our modified framework</b>	<b>Fact of the Design</b>	<b>Empirically proven</b>	<b>Qualitatively reported</b>	<b>Not known, believed true</b>	<b>Not known</b>	<b>Not known, believed untrue</b>
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 'todo' 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 greater expression and prioritization of the results in the summary.			X			
<b>Totals</b>	<b>1</b>	<b>1</b>	<b>6</b>	<b>6</b>	<b>1</b>	<b>1</b>

In addition to reflecting on the results of our analysis, we also wanted to know if there were any additional known search-oriented usability issues that we did not discover. Ignoring implementation bugs, which are not the focus of our inspection method, and sensemaking-based design considerations that led to a separate tool (Paul & 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. Consequently, however, 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 are already well supported, and so our 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, our analysis did not highlight a need to further improve it. Although further improvements to strong areas of a design can be easily modelled within our 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) has, we suspect, been suggested in light of similar findings to our results that led to usability statement 14 (in Table 3). 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. We look forward to seeing if this design idea becomes more concrete, so that its contribution can be analysed with our framework.

We can conclude from analysing these additional ‘undiscovered’ usability concerns 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, we do not suspect that users of our framework will typically identify these types of issues while inspecting their interfaces for weaknesses. Consequently, we do not consider that our framework has ‘missed’ these additional issues.

## 6. Conclusions

In this article, we have re-assessed an existing analytical inspection framework for information seeking interfaces in terms of its applicability to collaborative information seeking software. First, in Section 3, we re-framed the analytical inspection framework, and the underlying models used to ground the method, from the perspective of groups of searchers, explicitly collaborating on a shared task. In Section 4, we then describe how this modified framework is applied, using the re-framed models, to an example collaborative information seeking interface: SearchTogether. Through this example, as discussed further in Section 5, we 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.

We have made three main contributions. First, we have initiated a review of how established information seeking models still apply to collaborative information seeking, by re-framing two key models from the perspective of groups of searchers working on a shared task. Re-framing these example models highlights the way that we should reconsider the applicability of existing IS models and assumptions to collaborative settings. Second, we have described an analytical inspection framework, available online, that can be used to evaluate the designs of both new and existing collaborative search interfaces. The framework provides insight into how collaborative search interfaces support different types of search and different individual roles when searching as a group. Third, we have provided an evaluation, and potential design changes, of a freely available collaborative information seeking interface:

SearchTogether. Together these contributions support us in the on-going design and evaluation of interface designs that enable groups of searchers to work together on shared information seeking tasks.

## References

- Allen, T. (1977) *Managing the flow of technology: Technology transfer and the dissemination of technological information within the R&D organization*, MIT press Cambridge, MA.
- Amershi, S. & Morris, M. R. (2008) CoSearch: a system for co-located collaborative web search. In: *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, (pp. ACM, Florence, Italy).
- 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.
- 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.
- Brin, S. & Page, L. (1998) The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1-7), 107-117.
- Capra, R., Marchionini, G., Oh, J. S., Stutzman, F. & Zhang, Y. (2007) Effects of structure and interaction style on distinct search tasks. *JCDL07*, 442-451.
- 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.
- Golovchinsky, G., Adcock, J., Pickens, J., Pernilla, Q. & Back, M. (2008) Cerchiamo: a collaborative exploratory search tool. In: *Proceedings of the ACM 2008 conference on Computer supported cooperative work (Demo Session)*, (pp. ACM Press, San Diego, CA, USA).
- Golovchinsky, G., Qvarfordt, P. & Pickens, J. (2009) Collaborative information seeking. *IEEE Computer*, 42(3).
- Hansen, P. & Järvelin, K. (2005) Collaborative Information Retrieval in an information-intensive domain. *Information Processing and Management*, 41(5), 1101-1119.
- 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 (IiX'06)*, (pp. 144-152). ACM Press, Copenhagen, Denmark.
- Järvelin, K. & Ingwersen, P. (2004) Information seeking research needs extension towards tasks and technology. *Information Research*, 10(1), 10-1.
- 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.
- 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.
- Marchionini, G. (1995) *Information Seeking in Electronic Environments*, Cambridge University Press.
- Morris, M., Paepcke, A. & Winograd, T. (2006) Teamsearch: Comparing techniques for co-present collaborative search of digital media. In: *Proceedings of the First IEEE*

- International Workshop on Horizontal Interactive Human-Computer Systems, (pp. 97-104). IEEE Computer Society, Adelaide, Australia.
- Morris, M., Teevan, J. & Bush, S. (2008) Enhancing collaborative Web search with personalization: Groupization, smart splitting, and group hit-highlighting. In: Proceedings of the ACM 2008 conference on Computer supported cooperative work, (pp. 481-484). ACM Press, San Diego, CA, USA.
- Morris, M. R. (2008) A survey of collaborative web search practices. In: Proceedings of the SIGCHI conference on Human factors in computing systems, (pp. ACM Press, Florence, Italy.
- Morris, M. R. (2009) RE: IP&M submission: comments on SearchTogether Analysis. (Ed, Wilson, M. L.), (pp.
- Morris, M. R. & Horvitz, E. (2007a) S3: Storable, Shareable Search. In: Interact, (pp. ACM press.
- Morris, M. R. & Horvitz, E. (2007b) SearchTogether: an interface for collaborative web search. Proceedings of the 20th annual ACM symposium on User interface software and technology, 3-12.
- Morris, M. R. & Teevan, J. (2008) Understanding Groups' Properties as a Means of Improving Collaborative Search Systems. In: 1st International Collaborative Search Workshop, (pp., Pittsburgh, PA, USA.
- O'Day, V. & Jeffries, R. (1993) Information artisans: patterns of result sharing by information searchers. In: Proceedings of the ACM Conference on Organizational Computing Systems, (pp. 98-107). ACM Press, Milpitas, CA, USA.
- Paul, S. & Morris, M. (2009) CoSense: enhancing sensemaking for collaborative web search. In: Proceedings of the 27th international conference on Human factors in computing systems, (pp. 1771-1780). ACM Press, Boston, MA, USA.
- Pickens, J. & Golovchinsky, G. (2007) Collaborative Exploratory Search. In: Human Computer Information Retrieval, (pp. 21--22). Boston, MA, USA.
- Pickens, J., Golovchinsky, G., Shah, C., Qvarfordt, P. & Back, M. (2008) Algorithmic mediation for collaborative exploratory search. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, (pp. 315-322). ACM Press, Singapore.
- Prekop, P. (2002) A qualitative study of collaborative information seeking. *Journal of Documentation*, 58(5), 533-547.
- Resnick, P. & Varian, H. (1997) Recommender systems. *Communications of the ACM*, 40(3), 57.
- schraefel, m. c. (2009) Building Knowledge: What's beyond Keyword Search? *Computer*, 42(3), 52-59.
- Shah, C. (2008) Toward Collaborative Information Seeking (CIS). In: 1st International Collaborative Search Workshop, (pp., Pittsburgh, PA, USA.
- Sitter, S. & Stein, A. (1992) Modeling the Illocutionary Aspects of Information-Seeking Dialogues. *Information Processing and Management*, 28(2), 165-80.
- Smeaton, A., Foley, C., Gurrin, C., Lee, H. & McGivney, S. (2006) Collaborative searching for video using the fischlar system and a diamondtouch table. (pp. 8).
- Stein, A. & Thiel, U. (1993) A Conversational Model of Multimodal Interaction in Information Systems. In: Proceedings of the 11 National Conference on Artificial Intelligence (AAAI'93), (pp. 283-288). AAAI Press/MIT Press, Washington DC.
- Talja, S. (2002) Information sharing in academic communities: types and levels of collaboration in information seeking and use. *New Review of Information Behavior Research*, 3143-159.
- Twidale, M., Nichols, D. & Paice, C. (1997) Browsing is a collaborative process. *Information Processing and Management*, 33(6), 761-783.
- Wilson, M. L. (2008) A Transfer Report on the Development of a Framework to Evaluate Search Interfaces for their Support of Different User Types and Search Tactics. (pp.
- Wilson, M. L. & schraefel, m. c. (2007) A Combined Model for Evaluating Search Interfaces. In: Submitted to ACM Conference on Information Retrieval 2007, (pp.

- Wilson, M. L., schraefel, m. c. & White, R. W. (2009) Evaluating Advanced Search Interfaces using Established Information-Seeking Models. *Journal of the American Society for Information Science and Technology*.
- Wilson, T. (1981) On user studies and information needs. *Journal of Documentation*, 37(1), 3-15.
- Winograd, T. & Flores, F. (1986) *Understanding computers and cognition*, Ablex Publishing Corp. Norwood, NJ, USA.