Designing interfaces to support collaboration in information retrieval

Michael B. Twidale and David M. Nichols
Computing Department, Lancaster University, LA1 4YR UK

http://www.comp.lancs.ac.uk/computing/research/cseg/projects/ariadne/

mbt@comp.lancs.ac.uk

Abstract

Information retrieval systems should acknowledge the existence of collaboration in the search process. Collaboration can help users to be more effective in both learning systems and in using them. We consider how to build systems that more actively support collaboration. We describe a system that embodies just one kind of explicit support; a graphical representation of the search process that can be manipulated and discussed by users. A consideration of this system leads to an analysis of designing systems to support coping behaviour by users; including the need to support both help-giving by people, and recovery from the failure of intelligent agents. We also discuss the idea of interfaces as notations for supporting dialogues between people.

Keywords

Process Visualisation, Information Retrieval, Interface Design, Collaborative Learning

Introduction

> … support for communication and collaboration is as important as support for information-seeking activities and that, indeed, support for the former is needed to support the latter.
>
> (Levy and Marshall, 1995)

New interfaces to information retrieval (IR) systems are increasingly supporting progressive interactive search formulation and refinement. Part of their aim is to make information retrieval easier for the end user without the need for (or with minimal support from) an information intermediary. One approach to supporting these activities has been to design 'intelligent' interfaces that embody some of the knowledge and functions of a human intermediary (e.g. (Belkin and Marchetti, 1990)). An alternative, complementary, approach is to build an interface that actively supports interaction with an intermediary or other user: expanding the 'interactive' nature of the IR system to include other human resources (Hoppe and Zhao, 1994).

This social approach, taking direct advantage of human intermediaries' knowledge, in addition to any sophisticated underlying functionality, such as intelligent agents, is clearly more robust and adaptable than solely constructing an 'intelligent' stand-alone interface. It should be generalisable across databases, reactive to system change and likely to become more common as asynchronous interactions dominate accesses to remote digital libraries. This approach, and systems that take account of it, should fit better with the way that many people work (Grosser, 1991). Information

searching is part of people's larger work activities which generally involve some interaction with colleagues. These interactions can include recommendations of relevant items, the sharing of search tactics, and informal explanations and help about how to use a particular system. Similar activity has been observed in a range of contexts including the use of spreadsheets (Nardi and Miller, 1991). Systems which acknowledge the existence of formal and informal collaborations, and actively support it are likely to prove more useful and usable than current systems which seem to have been designed on the assumption of usage solely by individuals and yet which are often used in collaborative contexts.

The reasons why users are likely to need help are clear: the quantity of information is growing rapidly, the variety of information forms available electronically is also increasing, and there is a rapid rate of change of new information systems, new functionalities, versions and interfaces for existing systems. Consequently even the most enthusiastic user of information searching technology is faced with a continual need to update her skills. Furthermore we can assume that many (perhaps most) users do not find the technology intrinsically interesting but rather a tedious means to an end of obtaining the information that they are interested in. Thus we can expect the problem to persist of users having difficulties with online public access catalogues (OPACs) (Borgman, 1996), despite improvements in interface design. We need to design systems that are usable by Borgman's 'perpetual novices'.

We are proposing explicit support for collaboration as a way of addressing user's difficulties. However, collaboration has its costs; someone has to pay for the interaction with an expert user, and with large numbers of searchers, help from an expert may be hard to get. Thus we must be concerned with supporting cost-effective collaboration, in whatever form it occurs. Possibilities include:

- Helping users to explain quickly what they have already managed to do and what they now need help on.
- Minimising the amount of the expert's time that a help session will take by, for example, giving the expert a better understanding of the context of the query and making the delivery of help more efficient.
- Helping users to minimise their recourse to experts by facilitating the incremental learning of search skills, and by learning from their peers.

In this paper we consider the importance of building systems that more actively support formal and informal collaboration. We describe our first approach to implementing a system that embodies just one kind of explicit support for collaboration; a graphical representation of the search process. Our subsequent analysis of the lessons from developing this system includes issues of viewing interfaces as collaborative notations, and ways of supporting coping behaviour by users.

Collaboration in Information Retrieval

Collaboration already occurs in current physical libraries. However, in addition to the much studied reference interview with a skilled intermediary, substantial informal collaboration occurs. We have observed that many students at Lancaster University Library learn how to use the OPAC,

not from formal courses or documentation, but from their peers, leaning across adjacent terminals or clustering around a single terminal, engaged in synchronous co-located collaborative learning and working (Twidale *et al.*, to appear). It is noteworthy that this collaboration occurs *despite* rather than *because* of the systems – which seem to be designed on the assumption that users work alone. The collaboration is also desirable, leading to effective learning and reducing the load on library staff. We should emphasise that this form of learning still leaves room for improvement. It is very effective in familiarising users with the basics of the system and almost completely eliminating any computer anxiety with respect to the OPAC. However, users still have difficulties with the more advanced search functionalities and techniques, and some students still find other computerised systems (such as bibliographic CD-ROMs) intimidating. Our aim is to investigate ways in which a system that takes account of the fact that it may be used collaboratively, can enhance the effectiveness of such an interaction.

Although collaborative IR will inevitably encompass a wide range of activities we have chosen to concentrate on help-giving and the educational aspects that it entails. We are studying the nature of collaboration as it currently occurs in conventional libraries to give us a starting point for developing more advanced systems for use in digital libraries. Fowell and Levy (1995) discuss issues of remote help-giving, or more generally 'networked learner support'; examining how the working practices of professional librarians will need to adapt to the new environment.

Exploratory Prototyping

Building on information gleaned from the literature, interviews with subject librarians and a small sample of users, we combined some existing computational tools to build an exploratory environment for studying collaborative browsing. The aim was to undertake a rapid series of studies whose results would inform the design of our subsequent system. Consequently it was important to be able to create a number of experimental environments using technology to hand. We were particularly keen at this stage to investigate potential interactions between a relative novice and a more expert user such as a subject librarian. Volunteers were asked to bring along a genuine information need and try to address it in our laboratory setup.

We undertook a series of preliminary studies of synchronous remote help giving using a combination of rudimentary tools including Unix *talk*. The user would undertake a bibliographic search with a database accessible via Telnet and where necessary ask the expert for help. The interaction was synchronous and remote. These studies were notable for the problems that arose: user and expert would misinterpret what the other had said, they found it extremely difficult to describe what had been done and what should be done next. It became clear that a crucial need was to provide greater context by giving users the ability to share not only the search *product* (the hits obtained) but also the search *process* (how those hits were obtained). This was necessary so that the user could explain to the expert what they had tried so far, for the expert to understand what the user wanted to obtain, what they had tried so far and their opinions about what they had done. If the expert was to do more than undertake a search for the user and give her the result, (i.e. to show how

to do a search so that the user would be able to use the technique herself in the future), it was also necessary to pass process information in the other direction, from expert to user.

We have recently studied interactions between users and staff at a library help desk using ethnographic techniques (Crabtree *et al*, 1997). This study reinforced the importance of context; library staff would almost snatch at any piece of paper that an enquirer was holding (such as a book list, coursework specification or rough search notes) as a way of enriching the context of the articulated problem.

Not surprisingly when confronted by users claiming to be unable to find something, librarians frequently ask what they have tried. Equally unsurprisingly, users find this a difficult question to answer: it is very difficult to remember the search actions when one is focusing on the search goals, and there is a natural human inclination to auto-correct, remembering what one intended to type rather than what one did. Even worse, those most often asking for help are usually the ones who understand the system least and so are most likely to get confused and to lack a specialist vocabulary for describing their actions. By this argument we can see the desirability of recording the search activities so that in response to the question "What did you do?", the user can hand over the record and say: "This".

Thus even in a face-to-face help interaction, it is useful to have an interface that enables participants to share a representation of the user's search, and indeed perhaps of an expert's proposed solution. Where the user and intermediary are separated by distance and also perhaps time, such an interface becomes even more important. The study re-emphasised that information seeking is a multi-stage process (Bates, 1989). It does not consist of just formulating the 'right' query expression and getting back what you want. Often it is only by the forming of a sequence of search expressions and reading the partial results that users are able to decide (or at least articulate) what it is that they actually want. The goals of a search may evolve based on the results of previous search actions. A search activity can consist of the composition of many queries and can involve leaving a terminal, to consult books or people, and may include breaks while other work is done, so that a search may extend over an arbitrarily long period of time, not just the few seconds of composing a query and reading its results. Our systems should acknowledge this and more actively support it. Sadly this knowledge has been available (Bates, 1989), but seems to have been rarely addressed in designing systems and their interfaces. However some systems do provide a few features to support it. For example, BIDS keeps a record of the search queries issued that can be referred to (and replayed) within a session, but not between sessions. Mackinlay *et al.*'s (1995) innovative interface while focusing on the product of searches, does actively support the searching process seen as a larger activity, particularly the technique of citation chaining.

One method to achieve a shared context is to use a video-link between users – typically between end-user and library staff. However, video is not sufficient for many collaborative activities as it fails to make the search a discrete manipulable object (Sugimoto *et al*, 1995). The C-TORI model of (synchronous or near-synchronous) cooperative IR provides facilities for sharing query histories, cooperatively browsing results and cooperative query formulation (Hoppe and

Zhao, 1994). Nevertheless, the costs and inconvenience of synchronising help or collaboration are likely to override the advantages of its immediacy.

In order to support communication between people, the search needs to be preserved and turned into a suitable visualisation. However almost as a side effect, the resulting visualisation may be of use to a sole user. Once a system supports the creation of such an interaction history, it can used in many ways (Hoppe and Zhao, 1994). Single-user uses include: reuse of earlier searches, error recovery, navigation, reminding and user modelling (Lee, 1992). Providing a record of a search activity frees users from having to remember low-level goalstack details and enables them to concentrate on more strategic elements of the search process. As a digital object it can also be communicated to other people - serving as the basis for a much wider range of collaborative activities (Twidale *et al.*, to appear). Such a search object can be stored, highlighted, edited, annotated, replayed and re-executed: typically the kinds of activities that occur in tutorial and help-giving interactions (Lemaire and Moore, 1994). Thus we see the communication of context as a key requirement of a more supportive environment.

Design Criteria

We wished to build a system that supports a number of forms of collaborative activity that help users to achieve their goals in retrieving information. Some collaboration already occurs with existing systems even though deliberate account may not have been taken of this form of systems use.

We focus on supporting help-giving in an academic library, where there are large numbers of students, graduate and undergraduate, who lack important search skills. Users can receive advice from any of professional interm ediaries, experts or peers. We believe however that this work has implications for other contexts such as systems to support collaborative working where a group are cooperating to achieve a common goal and where (perhaps) help-giving plays a lesser role would benefit from similar support.

Collaboration may occur between peers, such as a group of students working together on a group assignment, or between people with different roles, such as a member of library staff and a patron. The nature of the interaction may vary in educational content and form, including peer-learning, tutorial interactions and surreptitious teaching under the guise of help-giving. Although it is desirable to build a system that supports all kinds of interactions, we chose to concentrate on a system that could act as a resource in supporting informal help-giving. The exemplar scenario we had in mind was that of a user attempting to undertake a search, encountering difficulties and then resorting to the expertise of a person at a help desk. For reasons of simplicity, we concentrated on the context of a physical help desk where the participants could talk around a single screen, but intend to use the lessons learned to inform remote help-giving, in addition to the other potential forms of collaboration briefly outlined above.

There are two complicating factors in developing a new functionality and resource for collaboration, particularly when designing for non-computer scientists:

- We cannot expect users to be able to precisely specify what they want of a new form of interface that can support (or even effect) new ways of working
- The requirements and use of the interface will co-evolve: people manage to use single user systems collaboratively in ways not envisaged by their developers. This led us to see a need for developing a system to more actively support collaboration. However such a system is itself likely to be used in new ways and to indicate the need for further modifications to more effectively support these ways of working, and so on. Carroll and Rosson (1991) have termed this issue the task-artifact cycle.

The aim was to produce a generic system that could be run with a range of databases, because users may need to consult a number of different databases during their searching and need varying amounts of help. It became clear from our preliminary studies that the terms 'expert' and 'novice' can be misleading: a user may be very proficient with one database but have great difficulties with another, particularly when there is such a variety of interfaces to databases. We chose to concentrate on databases accessible via Telnet with a vt100 terminal as providing a sufficient number while keeping the project tractable.

Finally we must emphasise that a key aim for such a small scale project was not so much to produce a deliverable system, but by developing a prototype to understand better the requirements for the system that should be built.

The Ariadne Interface

We have developed a prototype interface, Ariadne, which attempts to support collaborative activities. Ariadne runs on top of an OPAC or online database, recording the queries and their results and then subsequently producing a visualisation of the search process that can be reflected on, shared and discussed by interested parties. The recording of the information occurs behind the scenes; the user interacts with the database in the normal way. It is only after a search has been completed that the process representation can be viewed. This post hoc visualisation was provided for reasons of ease of implementation, although it does have the advantage that use of the system by novices imposes no cognitive overhead in the form of another, potentially distracting interface to be concerned with in addition to the interface of the database.

Although the system can be used in a variety of ways, it may be useful to understand the following description of use with a particular scenario in mind. Imagine a novice user of a bibliographic database, who chooses to switch on the data recording and then attempts to do some searching. After a few minutes she gets stuck, and takes the recorded search with her to the library help desk. The librarian looks at the search representation and uses it to guide a discussion with the user about what she wants, what she has done and what she might try. The librarian may twist the help desk terminal round to show the user the representation of her search, and point to relevant parts of it during the conversation. They may then switch to running the database at the terminal to continue the search.

<Figure 1 here>

Figure 1 The Ariadne interface

The history of a search attempt has some similarity with the Unix *history* option which lists the previous commands issued. However, in addition to the commands issued to the database, we also need to represent the results received back. An interaction with a database can be regarded as a sequence of input-output pairs. Some interactions will be administrative, such as inputting the letter or number of a menu option and receiving back as a screenful of text representing the next menu in the hierarchy. Other interactions will be search requests, such as typing in one or more keywords and receiving back a screen indicating the results of the search. Still other interactions will be concerned with viewing detailed search results, such as individual records. Ariadne has been developed to work with databases accessible over Telnet. As a result the input requests and resulting output screen are textual. The main challenge is not in obtaining and recording this history of the search process, but how to represent it in a useful and usable manner.

We have chosen a metaphor of playing cards, perhaps laid out for a game of patience. In Ariadne, a search is represented as a sequence of cards, where each contains the input search command (at the top) and the output result screen in thumbnail outline below. The scrollable sequence of cards can be read left to right as a history of the entire search activity. A card can be expanded to a readable size by clicking on them, as shown in figure 1. This reveals it to be part of a search of Lancaster's OPAC. The user has undertaken a title search on the word 'autism' and after scrolling through the list of hits has chosen to look at the details of a particular book. By looking at the unexpanded card we can see that she got to this record by choosing menu option '6' (thus indicating her wish to view details about the 6th item in the list of results). Her next action was to choose menu option 'L' which returns her to the list of returned titles.

<Figure 2 here>

Figure 2 A classic problem: too many hits and then trying to read them all

Additional qualitative information is provided by the vertical position of a card in one of three levels; a high level activity (such as menu selections from the top part of a system's menu hierarchy) is placed on the top level, search queries are placed in the middle level and actions involving the viewing of results on the bottom level. The intention was to provide an impression of the higher level abstractions of information searching. The result is that it is possible to obtain at-a-glance impressions of certain characteristic patterns of behaviour such as the common novice 'error' of composing a search that yields hundreds of hits and then proceeding to read through them all. (NB: Whether this is in fact an error is dependant upon circumstances: there are occasions when this is a sensible way of learning the language of a new subject domain.) Figure 2 illustrates a case where, using the BIDS bibliographic database, a user composed a search that yielded 4822 hits and then chose the display options and proceeded to try and read through them. The placing of the cards on the three levels is based on a small set of rules that have to be hand-coded for each database which Ariadne supports. In the future work section we consider ways of tackling this limitation.

As might be expected, a search session can yield a substantial trace and so in addition to the scrolling, facilities are provided to elide sub-sequences of cards, replacing them by a single group

card. A sequence of cards that have been elid ed, or folded together are indicated by a FOLD card (which is also in a different colour). Folds can be nested, in the manner of a file directory structure. All cards (including fold cards) can be annotated, eith er by the originator of the search or by others commenting upon it. The presence of an annotation is revealed, along with an indication of its size by marking the bottom section of a card. Selecti ng a card puts the annotation in the bottom left subwindow. Figures 1 and 2 contain examples of folds and annotations.

We wish to emphasise the simplicity of the Ariadne approach. It is merely an interface acting as a filter which provides a way of visualising activity in order to support dialogue about that activity. Clearly more sophisticated features could be adde d, but for a prototype we wanted to investigate the idea of process representation in a simple form.

Iterative Development and Studies of use

Ariadne has been used in a sequence of inform al, formative evaluations during its development. A key issue of concern was whether novice us ers would find this quite new interface and visualisation so confusing that it could not support the giving of help about the database they were using. As in the prelim inary studies, volunteer s undertook authentic activities, bringing along a search task that they had to undertake anywa y. These were generally of a background research nature such as finding information for an essay, dissertation or literature review, or to get a sense of the literature on a f ield to help inf orm the creation or refinement of an essay topic. So f ar over 50 volunteers have used the system.

The users varied from teams of four Ps ychology undergraduates undertaking a group report-writing assignment to individual students of W omen's Studies researching for a dissertation for a Master's degree. W e also studied relatively e xpert information searchers from the Com puting Department and the Library. Som e of the volunteers chose to return and continue using the system for their task for up to five sessions. Sessions were recorded on either audio or video tape. W e were not aiming to undertake a controlled experim ent but rather an opportunistic investigation of the nature of using, learning and teaching inform ation searching techniques and the evolving requirements of a system to facilitate this.

Given this approach, it is not appropriate for the experim enter to be passive observers. In order to maximise the degree of interaction (and also in order to offer som e recompense to the volunteers for giving up their tim e), at the end of a session th e experimenter occasionally acted as an expert, performing an educating / f acilitating role. The aim here was as usual to explore the requirem ents, functionality and usability of Ariadne, but this time from the perspective of a help-giver. Consequently it is inappropriate to discuss the de gree to which subjects were successful in their searches. Rather we are m ostly concerned with the problems that arise and how Ariadne can help, or potentially help, in resolving them.

The functionality of the system has been continually evolving and the interface has been refined during the course of the studies. Coupled with our attempt to get volunteers with as wide a range of backgrounds and abilities, this means that the findings must be tentative:

- The recording of the interaction allows us as researchers to obtain a growing and detailed set of data about the behaviour of users when browsing databases.
- All users, even complete novices in the use of bibliographic databases could understand the concept of the process representation. They were able to use the scroll bars to review what they or others have done and to open up a card to read its details.
- The use of folding and annotation was more problematic. The early versions were difficult for people to understand how to use. One reason for this might be utility: the studies were of synchronous use of the system whereas folding and annotation become much more useful in asynchronous working. Thus in addition to improving their interface, we need to study their use in an asynchronous context.
- When searching, subjects (individuals and groups) forgot what they had done a few minutes previously and what they said they were going to do next. Not surprisingly they also found it difficult to remember what they had done in a previous search interaction (which may have been a week before). Note that when searching, users are using the conventional vt100 interface to the database. Ariadne is recording their session but they do not have access to the history visualisation until after a search has been completed
- When the Ariadne visualisation of an earlier search was made available, both individuals and groups could use the it to review their actions and comment on what they had done and what they were intending to do next.
- Typing errors can scale up into strategic errors. The user may try a sensible strategy, make a typing error which causes the query to yield no hits, or inappropriate ones and then abandon the strategy because they are unaware of their error. The record is useful, particularly, as in the case of a very experienced librarian we observed, when the user is quite convinced that they did not make a typing error.

We observed many of the classic errors that have been noted and analysed in the literature (Tenopir, 1984). One particularly notable activity was users' lack of systematicity: they would often combine or generalise searches (a powerful browsing strategy) but fail to consider applying the technique to all the keywords they had been trying, often because they got side-tracked by a particularly interesting finding. A related problem occurred when moving up their goalstack. Often they had a complex strategy that was composed of a number of queries. One of the query steps would lead to an interesting result. They would side-track to investigate this result, but on returning to the main strategy, fail to resume at the correct point, missing out some issues that they had intended to investigate. The record is particularly useful in such circumstances for pointing out that they have not yet exhausted all the possibilities.

Novice users may have only a very superficial understanding of what a database is and consequently how it can and should be searched. Their mental model seems to equate the query language and search engine with an intelligent librarian. Consequently they seem to assume that the system will take account of syntactic and semantic variation. An example is a student who did not realise the searching on 'women' would fail to catch entries that contained the word 'woman'. When this was pointed out to her she was surprised that the database was that "stupid".

The record can be used as an effective teaching tool. It provides a medium for discussing the concepts and skills of information browsing. People can point to a sequence of actions and discuss what was done, whether it was effective or not, how it might have been done differently, and how the same technique could be done in another context, or even in another database. For an expert, the record is a very efficient means of determining the level of understanding of the user. Certain patterns of behaviour become dramatically evident, such as the 'get back hundreds of hits and try to read them all' error illustrated in Figure 2.

The record can be used to summarise a sequence of actions. So far this has only been observed in the case of the expert giving the novice(s) an overview of what they had done, allowing a degree of abstraction and permitting a dialogue on how to generalise the strategies that had been displayed. The scrolling and folding options are particularly useful in allowing this kind of abstraction, and were comprehensible to subjects. However, it took several iterations to achieve reasonable usability so we do not have data of non-experts using them effectively in an educational context. We intend to provide mechanisms to enable users to also be able to undertake this kind of reflective activity for themselves.

We found that on occasions when the experimenter was acting as a supportive expert (such as a subject librarian), it is all to easy to make mistakes about what the novices have done. In such circumstances, the record can also support an expert's faulty memory of a novice's actions. Even an expert sitting alongside a novice can forget or misinterpret what has been done. A subsequent attempt at tutoring will naturally be very unproductive in such circumstances

A main advantage of such an interface is that discussions (both remote and co-located) can then be undertaken between participants about sophisticated searching techniques without the need to learn the *specialised vocabulary* of the information profession. By contrast, instead of using the abstract conceptual terminology, the participants can point to, or highlight, the particular instances that were of significance in the search under discussion.

Evaluation issues and future work

Clearly we need to undertake further studies and evaluations of the effectiveness of the Ariadne system. These should include both their effect in synchronous and asynchronous interactions. The studies outlined above were intended to inform the continual development of the system, although they do also provide insights into the general needs of interfaces to support collaboration.

Our concern throughout the project has been to develop a system that can fit into the actual practices of information systems users. Consequently we have been concerned that the studies of use be as authentic as possible. This was not the case for the location: we were constrained to undertake our studies in the laboratory rather than in the ideal of the library itself. However we did use genuine users' search needs rather than inventing problems that (perhaps unconsciously) show the power of the system in the most favourable light (Twidale *et al.*, 1994).

Ariadne was originally developed in C++. We have recently ported it to Java and a demonstration version (users can review existing searches but can't create new ones) is available at:

http://www.comp.lancs.ac.uk/computing/research/cseg/projects/ariadne/demo.html

We hope that it will make our ideas clearer than from just seeing two screendumps of what is crucially a dynamic system.

We noted in the introduction that one of the causes of a need for help-giving was the rapid rate of change of systems and their interfaces. This implies a need for 'future proofing' of help support systems like Ariadne. The simplicity of Ariadne's approach means that it can be applied to any text-based interaction with a database such as can be achieved via a vt100 Telnet connection. However for each new database it will be necessary to develop a new set of rules for determining the vertical positioning of the cards in the interface. This is needed because we do not have information of the underlying semantics of the search actions - these and their interpretation are left to the human users of the system. We are currently investigating the possibilities of developing a version of Ariadne for Z39.50 compliant databases. With a Z39.50 connection, we do know the semantics of each request and so can build an interface that should be able to be used with any compliant database.

At the early stages of this project we made claims for the generality of the current text-based Ariadne which now have a rather hollow ring. We developed versions for Lancaster's OPAC and for BIDS as a proof of concept, claiming this showed it could be applied to any text based database. However both those systems now also offer a web-based interface. These are currently alternatives to the main text based interfaces, but we can expect increasing numbers of users to prefer the newer interface. Clearly we need to address how Ariadne can be used with a web-based interface to an information system employing forms, rather than sending simple text commands. We are currently investigating this, concentrating on the development of a version of Ariadne that can sit alongside a web-based search engine such as Alta Vista. There is no essential difference between supporting a web-based search engine and a web interface to a bibliographic database. Thus we hope to recover our claims for generality in the next version.

Discussion

As this is an interdisciplinary journal, we feel it is important to clarify our approach, which is in the engineering style of 'building in order to understand'. We undertook some preliminary studies of activities in physical libraries in order to better to understand these activities and the implications they have for potential interactions in digital libraries, where those interactions may be either remote or co-located. We identified an important attribute of a more supportive system, namely a visualisation of the search process. We then developed a system that implemented a version of that attribute. The development was based on iterative development and testing of the evolving prototype with a range of users all bringing authentic search problems. The development-test-analysis cycle enabled us not only to improve the design of the system in question, but also to use the evolving tool as a basis for more detailed study of user's understanding of the search process and of functionality that could help them. We note here some of the broader issues that have emerged from these informal qualitative studies.

In part, the design of the interface was inspired by work on interfaces for tutoring systems drawing on the Artificial Intelligence and Education paradigm, particularly regarding the explicit representation of processes and abstract concepts (Collins and Brown, 1988). The visualisation of the search process provided by Ariadne is a means for searchers to reflect on their activities. Reflection in information searching, as in other domains, is facilitated by making explicit "the student's problem-solving processes - their thrashings, false starts and restarts, and partial successes" (Collins and Brown, 1988). Access to a representation of the intermediate states of the search process, such as in Ariadne, is crucial for effective post-problem reflection (Foss, 1987; Katz and Lesgold, 1994). Similarly, a computational representation can make visible aspects that are internalised to an expert but are difficult for a novice to acquire precisely because of their conventionally invisible nature (Hollan, Hutchins and Weitzman, 1984). The reification process offers great advantages in facilitating both learning about the concept and cooperative working using the concept.

Increasingly users are encouraged to undertake searches for themselves rather than directly employ an intermediary. We can gain a better understanding of this interaction as it currently operates, and how it might operate, by studying it both as it occurs in conventional physical libraries, and by comparing it with other similar interactions. These include computer support telephone lines, where users of systems receive synchronous remote help (Pentland, 1992). Note that one of the significant differences from a reference interview and the work of information intermediaries is that this kind of interaction is not strictly intermediation: the user has tried to do the search herself and now has hit a problem and needs help to recover. The help giver may need to perform a number of tasks: to help the user to find the information, to help the user learn a new technique using the system to find this information and to improve her future effectiveness, to introduce the user to a new generic information searching concept and to help the user recover from an anomalous state as a result of their own explorations and attempted repairs. This anomalous state may be a system state (as when a software user calls a help desk because their computer is behaving oddly, perhaps as a result of their attempts to fix a relatively minor problem), or a cognitive state (as when a user has made inferences from her actions and the system's responses that although understandable, are inadvisable, leading to a substantial misconception about a searching technique). This form of working has implications for the design of a system that can best support it.

Designing for coping and for failure

Our observations confirm those of others; that many users have great difficulties with databases. They make many errors and perform far less optimally than an expert would. Nevertheless, many users do manage to cope, being able to retrieve some information, even if it is not all that is relevant, and even though the methods that they use might horrify an information professional. Many users have coping strategies, and indeed quite modest goals for information retrieval. They may not want a definitive result, or they may not wish to invest the resources of learning how to undertake an expert search, or to employ the services of an expert intermediary. It

is clear that one reason for using coping strategies is that users are overwhelmed with the number of activities that they have to do, of which acquiring information search skills is just one. Furthermore, the growth in range and number of data sources and the great variation in interfaces even within a single medium such as CD-ROMs, add to the complexity of the task. The fact that interfaces often change between versions again does not provide much of an incentive for investing time and effort in learning search skills.

One technological approach to tackling this problem is to try to develop agents and intelligent interfaces which will enable users to specify their information needs in their own language (or a close approximation to it) and to receive useful results in a comprehensible form. However, we believe that the results of work in this area can be better exploited if greater effort is made in supporting coping and failure. That is, we should design systems while explicitly acknowledging that they may fail to achieve their users' goals on their own. This means diverting resources from solely attempting to improve the success rate and instead allocating significant resources to coping with failure. By 'failure' we mean not only cases where the system is unable to satisfy the user's requests, but also where the user has difficulty in using or learning to use the system. Here the problem may be in articulating the request in a way that the system understands or in interpreting and working with the results obtained to refine the search. It is such 'failures' that can precipitate a visit to a help desk, or the demand for more accessible forms of help-giving.

One way of viewing Ariadne is as an example of an interface for supporting coping and failure. In the case of Ariadne, the process visualisation supports failure by making it easier to ask a person for help in cases where the user recognises that she has got stuck. We see this approach as complementary to the design of intelligent interfaces and use of agents. As Nardi and O'Day, (1996) note, it is helpful to distinguish those activities currently done by experts that are tractable for agents to undertake, and those which are not. Even if an agent were to fail in its independent search, if it provided a comprehensible visualisation of its attempts, this could be user by the searcher (or by an expert) as the basis for a more intelligent human-originated but agent-inspired search. Although ultimately complementary in operation, the *development* of failure-supporting features will be in competition with agent development for project resources.

Clearly designing with failure in mind can feel rather peculiar, particularly as it requires the allocation of resources away from designing the underlying functionality which is intended to improve effectiveness. So it can become self-fulfilling if it means that insufficient resources are available to enable the agents to be perfected. Nevertheless we believe that such an allocation of resources is worthwhile and it is worth researching into how to develop systems and interfaces that acknowledge and support failure recovery by people.

We see the investigation and development of interfaces for coping as supporting a culture of incremental, opportunistic learning and help-giving. The information systems that are being developed and the contexts of their use are too complex and changing too rapidly for us to assume that the majority of users are either experts, or indeed novices who want to become experts as soon

as possible. We should support Borgman's (1996) 'perpetual novices' (who may nevertheless be experts in other domains) and their inevitable coping strategies, failures and need for help.

Interfaces as notations

Interfaces such as a process representation also can be analysed in terms of notations for mutual understanding:

- The process representation can be considered as acting like a debugging visualisation (for example, Ungar *et al.* (1997)) to help a user understand what the system was trying to do, and how that contrasted with what the user wanted the system to do.
- The process representation can be considered as a notation for supporting understanding between two people. In the same way that engineers may explain their ideas in a meeting by drawing box and stick diagrams on a whiteboard, a process representation may help a user to explain to another person what they had done and serve as a basis of explaining what they wanted to do next.

A recent ethnographic study of activity at a library help desk (Crabtree *et al*, 1997) revealed in more detail a number of issues that confirm the importance of context. One example was the frequent practice of librarians turning their OPAC screen round so that the user could see it as well. This was not solely to show the results of a successful query but also at earlier stages of the help-giving interaction where the main focus of activity is for user and librarian to mutually clarify the underlying information need of the user. In order to do this, the librarian may undertake searches or choose menu options on the OPAC, and the conversation incorporates what can be seen on the screen, and includes participants pointing at the screen. Note that this is a kind of interaction using a computer interface that is valuable but different and supplemental to the 'conventional' understanding of the purpose of an interface.

Conventionally, an interface is viewed as a medium for communication between a user and the underlying complex functionality. In the case of an OPAC that means a way of composing queries and receiving the results, both in a form easy to learn and use by the envisaged users. In the scenario above, in addition to that conventional use, the interface is also being used as a medium for supporting discussion between two people, where the results on the screen help them establish the kind of items that are wanted as well as (later in a help interaction) hopefully yielding a result. The interface is serving as a notation that can be used as a basis for clarifying dialogues which can refer to it verbally or by physically pointing to it. Unlike notations on paper (such as schematic diagrams) that similarly can be used to supplement a dialogue, the interface-as-notation can be dynamic, with the process of interactions being the focus of the dialogue as well as a single screen.

So an interface that was designed for use by a single user is also being used collaboratively. Furthermore, this collaborative interaction is different to the conventional view of interfaces to support collaboration. The interface is not one through which the users collaborate, but acts more like a tool to enhance the existing collaboration. In this scenario, user and librarian are next to each other and so do not need computer mediated support to enable collaboration. However it is easy to

envisage remote (a)synchronous help-giving contexts where the interface to the database serves this dual purpose.

This observation has similarities with the work of Greatbatch *et al.* (1993). They noted the problems that arose in doctor-patient interactions when the doctor was using a computer system to create a prescription as opposed to pen and paper. Here the interface to the prescription software had been designed as a single user system and was indeed being used solely by the doctor, but had a negative impact on the doctor-patient conversation that occurred while the prescription software was being used. Problems arose because "it appears that the design is centred around the individual user" when in fact the system has a social context of use.

Conclusion

We have proposed the development of interfaces for information retrieval systems that are not merely oriented to the delivery of the results of a search, but attempt to support collaboration with other users. These others may be expert intermediaries, co-workers, peer learners, etc. Such systems should help users to cope with continual and rapid change in data size, data types, systems, functionality and interfaces. One way of supporting collaboration is to provide a visualisation of the search process that can be manipulated and discussed by the users. We believe that support for collaborative use of an IR system can improve the learning and understanding both of that system and of generic search skills. In addition, by acknowledging the importance of other people in the search process, it can lead to a more robust search process, more able to cope with, and indeed exploit, the failures of any underlying intelligent search mechanisms.

Acknowledgements

Work on the Ariadne project has been funded by the JISC New Technologies Initiative of the HEFCs and by the British Library Research and Innovation Centre. We wish to acknowledge the contributions of Gareth Smith, Jonathan Trevor and Damon Chaplin to the development of various versions of Ariadne. We would like to thank Debra Cash and Kate Erlich for their contribution to our understanding of the changing roles of intermediaries.

References

Bates, M.J. (1989) 'The design of browsing and berrypicking techniques for the online search interface' *Online Review* **13**, 407-24

Belkin, N.J. and Marchetti, P.G. (1990) 'Determining the functionality and features of a intelligent interface to an information retrieval system' *Proceedings of the 13th International Conference on Research and Development in Information Retrieval (SIGIR'90)*

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

Carroll, J.M. and Rosson, M.B. (1991) 'The task-artifact cycle', in **Carroll, J.M.** (ed), *Designing Interaction: Psychology at the human computer interface* Cambridge University Press

Collins, A. and Brown, J.S. (1988) 'The computer as a tool for learning through reflection' in **Mandl, H. and Lesgold, A. (ed)** *Learning Issues for Intelligent Tutoring Systems*, Springer-Verlag

Crabtree, A., Twidale, M.B., O'Brien, J. and Nichols, D.M. (1997) 'Talking in the Library: Implications for the Design of Digital Libraries' *Technical Report CSEG/1/97* Computing Department, Lancaster University

Foss, C.L. (1987) 'Learning from errors in Algebraland' *Technical Report IRL-87-0003* Institute for Research on Learning, Palo Alto, California

Fowell, S. and Levy, P. (1995) 'Developing a new professional practice: a model for networked learner support in higher education' *Journal of Documentation* **51**, 271-80

Greatbatch, D., Luff, P., Heath, C. and Campion, P. (1993), 'Interpersonal communication and human-computer interaction: an examination of the use of computers in medical consultations' *Interacting with Computers* **5**, 193-216

Grosser, K. (1991) 'Human networks in organizational information processing' *Annual Review of Information Science and Technology* **26**, 349-402

Grudin, J. (1994) 'Groupware and social dynamics: eight challenges for developers' *Communications of the ACM* **37**, 92-105

Hollan, J.D., Hutchins, E.L. and Weitzman, L. (1984) 'STEAMER: an inspectable simulation-based training system' *AI Magazine* **5**, 15-27

Hoppe, H.U. and Zhao, J. (1994) 'C-TORI: an interface for cooperative database retrieval' in **Karagiannis, D. (ed)** *Database and Expert Systems Applications*, Springer-Verlag

Katz, S. and Lesgold, A. (1994) 'Implementing post-problem reflection within coached practice environments' *Proceedings of the East-West International Conference on Computer Technologies in Education (EW-ED'94)* Vol. 1 125-30

Lee, A. (1992) *Investigations into History Tools for User Support* PhD Thesis, Department of Computer Science, University of Toronto

Lemaire, B. and Moore, J. (1994) 'An improved interface for tutorial dialogues: browsing a visual dialogue history' *Proceedings of the Conference on Human Factors in Computing Systems (CHI'94)* 16-22

Levy, D.M. and Marshall, C.C. (1995) 'Going digital: a look at assumptions underlying digital libraries' *Communications of the ACM* **38**, 77-84

Mackinlay, J.D., Rao, R. and Card, S.K. (1995) 'An organic user interface for searching citation links' *Proceedings of the Conference on Human Factors in Computing Systems (CHI'95)* 67-73

Nardi, B.A. and Miller, J.R. (1991) 'Twinkling lights and nested loops: distributed problem solving and spreadsheet development' *International Journal of Man Machine Studies* **34**, 161-84

Nardi, B.A. and O'Day, V. (1996) 'Intelligent agents: what we learned at the library' *Libri* **46**, 59-88

Pentland, B.T. (1992) 'Organizing moves in software support hot lines' *Administrative Science Quarterly* **37**, 527-48

Sugimoto, S., Gotou, S., Zhao, Y., Sakaguchi, T. and Tabata, K. (1995) 'Enhancing usability of network-based library information system – experimental studies of a user interface for OPAC and of a collaboration tool for library services' *Proceedings of Digital Libraries '95*

Tenopir, C. (1984) 'To err is human: seven common searching mistakes' *Library Journal* **109**, 635-36

Twidale, M.B., Nichols, D.M. and Paice, C.D. (to appear) 'Browsing is a collaborative process' *Information Processing & Management*

Twidale, M.B., Randall, D. and Bentley, R. (1994) 'Situated evaluation for cooperative systems'*Proceedings of Conference on Computer Supported Cooperative Work (CSCW'94)*, Chapel Hill, NC, ACM Press, 441-52

Ungar, D., Lieberman, H. and Fry, C. (1997), 'Debugging and the experience of immediacy' *Communications of the ACM* **40**, 38-43

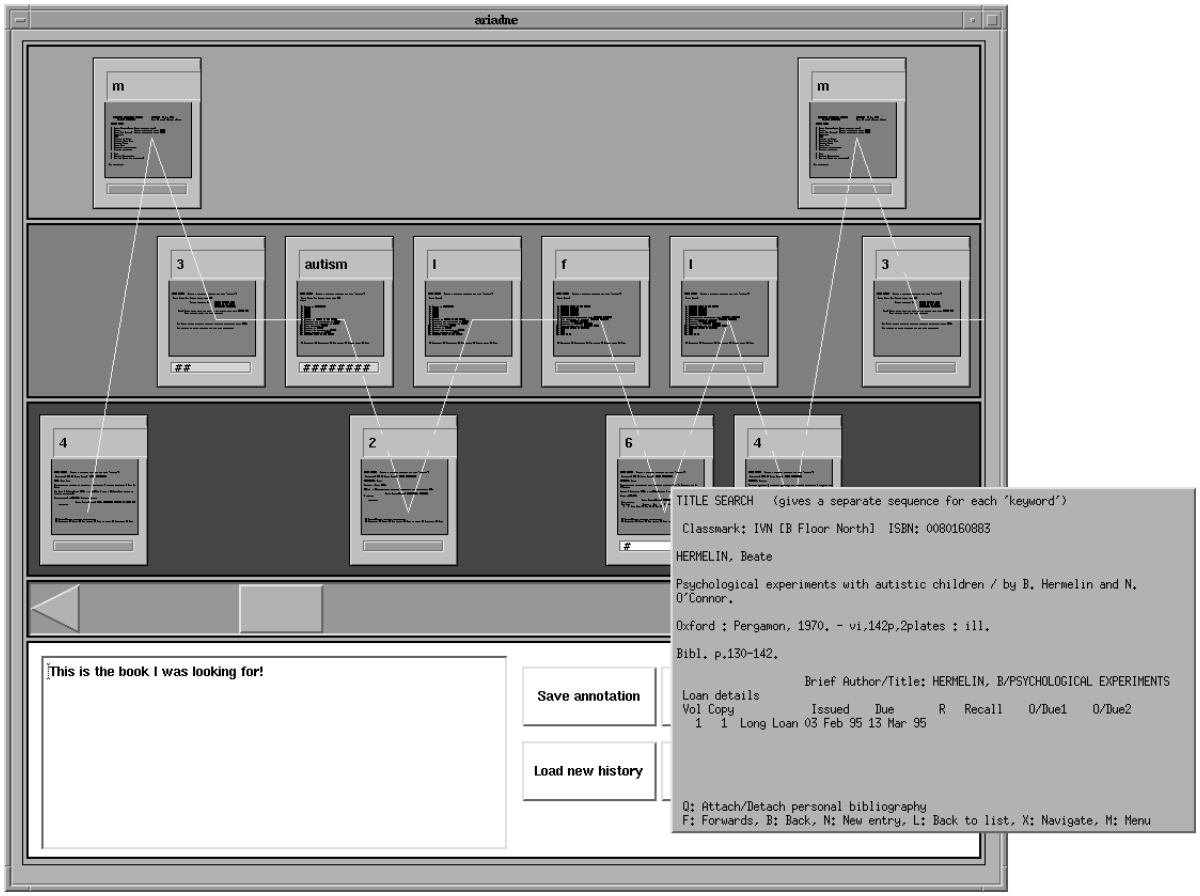

Figure 1 The Ariadne interface

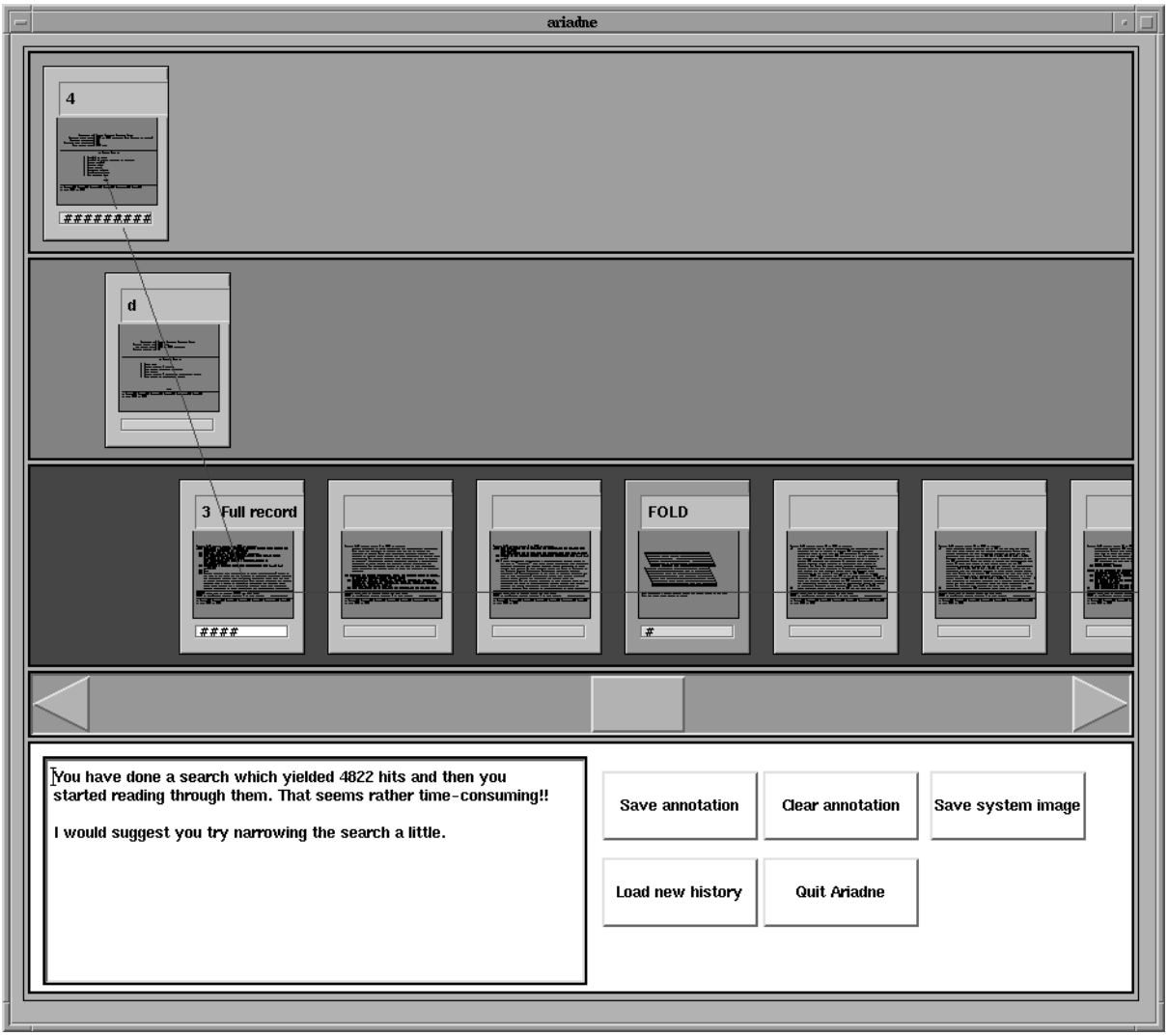

Figure 2 A classic problem: too many hits and then trying to read them all

Twidale, M. and Nichols, D. (1998) Designing interfaces to support collaboration in information retrieval, *Interacting with Computers*, 10(2) 177-193.

http://dx.doi.org/10.1016/S0953-5438(97)00022-2