11

An evaluation scheme for trader user interfaces

Andrew Goodchild*

Department of Computer Science, The University of Queensland, QLD 4072, Australia

This paper provides a set of criteria to evaluate the usability of trader user interfaces with respect to resource discovery. The criteria were identified because trader user interface developers sometimes make unrealistic assumptions about the information seeking behavior of their users, leading to interfaces that are difficult to use for resource discovery. This paper uses some insights from the cognitive aspects of the information science field to justify the usability criteria. The results of this paper are general enough to aid the developer of a user interface to any resource discovery tool.

Key Codes: C.2.4; H.3.3; H.5.2
Keywords: Distributed Systems; Information Search and Retrieval; User Interfaces

1 INTRODUCTION

Resource discovery tools are systems which enable users to search for and employ resources in very large networks [1]. Traders provide an advertising/matchmaking service for objects in a distributed environment [2]. Naturally there is a desire to use the functionality of a trader for resource discovery by attaching a user interface (UI) to it.

Current research in the trader has largely come out of the OSI arena, where the main problems considered have been technical ones. The paradigm behind such research may not be totally suitable for resource discovery. Application of rational methods, as seen in current trader research, leads to systems that suffer from a strong technological bias; and as Ellis [3] likes to describe it: "the user is equated to an peripheral input/output device." It is natural for developers when they attach UIs to traders to want to avoid this problem. However, when developers implement UIs for traders, they make assumptions about the way users seek information. Some of these assumptions maybe unrealistic.

The purpose of this paper is to question the developers assumptions about usability by providing a treatise from the information science literature. This paper does not seek to solve the traders usability problems, nor does it suggest specific technical solutions to existing problems. What this paper does is to introduce an evaluation scheme from the users point of view for trader UIs.

Ideally this usability evaluation scheme should be useful for aiding developers in isolating possible failure points in existing trader UIs and help in identifying new areas for trader UI development. A full set of usability criteria would cover an immensely broad

*email: andrewg@cs.uq.oz.au

K. Raymond et al. (eds.), *Open Distributed Processing*

set of issues, including general user interface design principles [4, pages 65-88]. The focal point of this work is to identify a set of usability criteria derivable from a representative model of the general cognitive behavior of information seekers. In other words, rather than base this work on the human computer interaction (HCI) literature, this work is based upon librarians' observations of how people find information. Information science provides a useful starting point as librarians have been dealing with resource discovery problems for centuries now, and to ignore it would be inviting rediscovery of much existing knowledge.

This paper is related to and supports the quality of service issues raised by Milosevic et al [5]; hence references to it will be made from time to time.

2 THE TRADER UI USABILITY CRITERIA

2.1 Interconnected searches

The typical model, as Bearman [2, page 38] illustrates, for the client (importer) interaction with the trader involves first a service request (import request) followed by a reply from the trader with a match to the most appropriate service offer (import reply). Taken at face value, which I think many trader UI developers are, this model is much like the classical representative model [6] in information retrieval.

The classical model, as illustrated in Figure 1, is concerned with the user's actions within a single episode. The episode starts by the user first identifying an information need, formulating a query (i.e. like an import request) which is matched against the database contents, and finally produces a single output set (i.e. like an import reply).

Unfortunately such a model does not fit well to real-life searches involving people. Salton [7] observed that before people arrive at a final result set, they gradually refine their query. As a result, he enhanced the classical model by introducing interactive feedback, or query reformulation, to improve the output. Salton's model is illustrated in the center of Figure 1.

The first interpretation is also at fault as it presumes that the information need leading to the query is the same, no matter what the user might learn from the documents in the preliminary retrieved set. Hancock [8] disproved this presumption when observing students search in a library; she found evidence that searching is an adaptive process. In real-life searches in manual resources, end users may begin with just one feature of a broader topic, and move though a variety of sources. Each new piece of information they encounter gives them new ideas and directions to follow and, consequently, a new conception of the query. Thus the point of feedback is not only to improve the representation of a static need; but, to provide information that enables a change in the information need itself.

Furthermore, with each new conception of the query, the user may identify useful information and references. In other words, the query is satisfied not by a single retrieved set, but by a series of selection of individual references and bits of information at each stage of the search. A better interpretation of query reformulation is illustrated on the

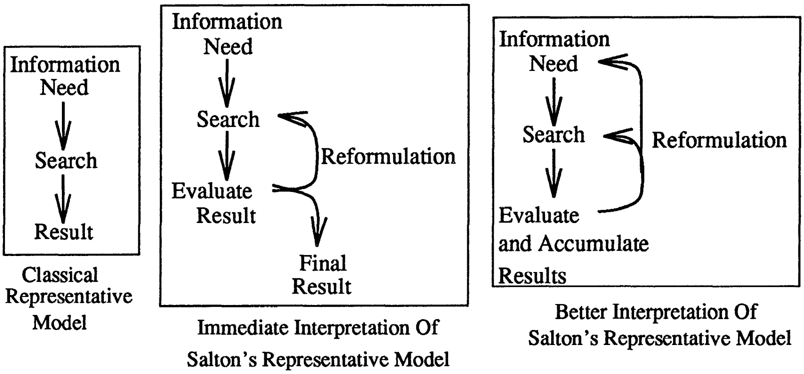

Figure 1: Representative models of information seeking

right hand side of Figure 1. Bates [6] and O'Day and Jefferies [9] also confirmed the same behavior. Hence, the users interaction with the trader is more than Bearman's [2, page 38] model suggests.

When the trader UI has poor provision for interconnected searches, users are often left dissatisfied with the tool. Immediately, it can be seen that responsiveness and reliability are an important criteria. For example, when online catalogs were batch oriented frequent users of such catalogs were often frustrated by the one-shot query orientation of the system and preferred continuous feedback [10]. In general, as Rushinek [11] observed, a lack of responsiveness usually leads to user dissatisfaction. This suggests:

Usability Criteria 1 *the trader UI must be reliable with quick response times.*

Milosevic et al [5] are indeed right when they suggest that characteristics of quality services are responsiveness and reliability. Users would find a trader UI with long delays for import requests and replies or traders which are down quite often frustrating to use simply because they cannot "interact" with the trader to perform interconnected searches.

Another case of poor provision for interconnected searches is not acknowledging that its the accumulation of results that matters and not the final retrieved set. For example the Z39.50 standard [12] until recently didn't make this acknowledgment. Z39.50 version 1 was stateless, meaning that the server could discard the results as soon it had sent them to the client. Version 2 supports state, allowing the user to perform queries upon queries in an interconnected fashion. This leads to:

Usability Criteria 2 *the trader UI must support the accumulation of search results.*

The result of this criterion is that it maybe inappropriate to assume that the binding between the UI and the trader is stateless. Users need to be able to refine their searches by performing additional searches on their existing search. In addition, a positive step would be to provide the ability to store search results for later use or for combining with several other searches. Management and refinement of search results is a feature of a trader UI that is underdeveloped and can be improved.

2.1.1 Changing information needs

Supporting changing information needs in a trader UI is a little more difficult. O'Day and Jefferies [9] made a key observation in trying to understand changing information needs. Their observation was that there was some decision that led to the next step in the interconnected search: a "trigger". These triggering actions where the result of encountering something interesting, or explaining a change, or finding missing pieces or pursuing a plan. The motivation for triggers was identified by Kuhlthau [13]. Kuhlthau described information seeking as a process which moves the user from a state of uncertainty to understanding. She asserted that it was the anxiety associated with uncertainty that motivated users to search for information.

O'Day and Jefferies [9] observed that interconnected searches stopped when there were no more compelling triggers for further searching or in a few cases, there were specific inhibiting factors like lack of time, money, or expertise in a field to allow effective searching. O'Day and Jefferies labeled this behavior as encountering "stop conditions". Again, Kuhlthau provides some insight into the motivational aspects. She asserted that a sense of confidence is associated with the understanding that search results bring. If users start experiencing redundancy in their search results, their confidence grows until they are satisfied with their search. If users experience many different unique search results their confidence shrinks until they are disappointed with their search.

Hence stop conditions are strongly related to anxiety in the searcher, and as a result inducing artificial ones will prematurely terminate searches. For example Bysouth [14, page 60] noted that complicated connect time pricing schemes for online databases induce artificial stop conditions, inhibiting interaction. Thus we must take great care in designing tools and ensure that:

Usability Criteria 3 *the trader UI must not induce artificial stop conditions.*

All features of a trader UI that users are likely to encounter must be readily understandable; otherwise, the anxiety of use is likely to cause the user to stop using the trader prematurely. This is in alignment with Milosevic et al [5] concept of "simplicity" as one of the factors determining the quality of service. An example of artificial stop conditions would be complicated charging schemes which result in the users perceiving a high cost, thus deterring them from extended searches in attempting to locate their resource.

Lack of sufficient triggering information can also lead to users prematurely stopping their search, in many cases falsely satisfied, even though there is much more relevant material to be found. For example Bates [15] observed this behavior in subject catalog users. This problem is directly linked to the fact that the library she studied could not afford the resources to put "see also" entries in their subject card catalog, thus inhibiting the production of triggers. If an initial search is unsuccessful, then the users biased feeling that no relevant material exists is confirmed, artificially inducing a stop condition.

This problem is related to Bates' PRINCIPLE OF VARIETY [16]: "The variety of query formulation must be as great as the variety of document (resource) descriptions for search success."

Authors and indexers produce a great variety of terms in their indexing, so to cope successfully searchers must also produce an equal variety in formulating searches on any given topic. For example some interdisciplinary topics like artificial intelligence can be classified into many different areas in the library. As a result you may find books in areas where you do not normally expect them; so the searcher is now responsible for generating many ways to effectively search. The variety that searchers can generate in their queries will inevitably be much less than the variety of resource descriptions, so to help the users cope the trader UI developer could:

1. *Reduce the variety of resource descriptions.* Traditionally this has been achieved by vocabulary control, i.e. reduce the variation by eliminating variety in words by morphological analysis to reduce different word forms to common "stems", for example: removing plurals, verb conjugations, synonyms, etc. More sophisticated techniques based upon natural language understanding have yet to be shown to be cost effective [17]. The simpler techniques have had considerable success in improving search performance – but it has its limits. We can decrease the variety in language used to describe information, but we cannot reduce the variety of information itself without defeating the purpose of the information retrieval system.

2. *Increase the variety of users queries.* The principle mechanism for achieving this is to use cross referencing. Users can increase the variety in their search by following links between terms. Another technique is to have users expand queries by selecting additional terms from lists suggested by the system. Belkin and Croft [17] noted that this technique was not effective. The reasons for these differences are not obvious, although it appears that using only system suggestions is too restrictive and does not make full use of the users domain knowledge.

Traders with poorly structured offer spaces will provide inadequate triggering information, and may result in the user prematurely stopping their search even though there are many relevant resources available. Thus using type management of service type facilities like relationships and subtyping to provide syndetic structure (*see also* entries) should be strongly encouraged. In addition, tools like morphological analysis should be a capability of the query interface. Milosevic et al [5] characteristic of accuracy in exporting is symptomatic of failing to meet this criterion. Hence following these design principles leads us to:

Usability Criteria 4 *the trader search interface/offer space must respect the principle of variety.*

2.2 Querying and browsing

There is a tendency among resource discovery tool developers to see browsing a casual, don't-know-what-I-want behavior that one engages in separately from "regular" searching

[6]. This emphasis that browsing is not "regular" leads to a bias held by many information retrieval tool developers including trader UI developers and that is: a bias towards querying as a main search interface.

This bias is unrealistic as browsing is a rich and fundamental part of human information seeking behavior – one may even argue that it is queries that are irregular. The notion that a well defined query is possible was challenged by Nicholas Belkin [18]. Belkin claimed that for a user to state their information need, they have to describe what they do not know. In effect users do not naturally have "queries"; but, rather they have what is Belkin calls an "anomalous state of knowledge"[18, page 62].

Browsing is a fundamental behavior as it is the users attempt to deal with what Bates calls the PRINCIPLE OF UNCERTAINTY: "Document description and query development are indeterminate and probabilistic beyond a certain point." This principle reflects the difficulty users have in formulating queries. Essentially, if a user is not looking for something they already know exists; then, the more specific the query is the more likely that it will fail. A related example would be to ask two people to describe a very specific resource; the principle of uncertainty manifesting itself as them using very different terms. Whereas, if you were to ask them to describe a fairly general resource then they are more likely to use more similar terms.†

Focusing on tools which just support browsing behavior only is not the answer either. Conklin [20] described that early hypertext tools lacked support for indices, which became a problem for users of large hypertexts. Users often complained that they became "lost in space" – not knowing how to get to a node that they know, or think, exists. It wasn't until the ability to query an index of the hypertext was possible that being lost became less of a problem. To help users perform both general and specific searches the tool needs both querying and browsing facilities. Queries can help users get closer to their desired information faster, but only by browsing will they be able to locate the most specific items, unless of course they already know where it is. Acknowledgment that querying and browsing are both necessary leads to:

Usability Criteria 5 *the trader UI must seamlessly provide for both querying and browsing.*

The designers of trader UIs cannot expect users to identify their specific resources by performing just queries, to do so would be violating the principle of uncertainty. In addition, it is too much to expect the users to find their resources by browsing all the time. Thus there should be a certain degree of search flexibility provided by the trader UI. Rather than supporting only queries, as has been popular in the past, both querying and browsing facilities should be supported in an integrated fashion.

†As an aside, one interesting effect of the uncertainty principle is it allows authors like Salton to make rather counter-intuitive statements like [19]: "Evidence from available studies comparing manual and automatic text-retrieval systems does not support the conclusion that intellectual content analysis produces better results than comparable automatic systems."

2.3 The role of strategy in searching

Developers maybe tempted to interpret the interconnected search undertaken in subsection 2.1 as simply a series of queries of the classical sort. This interpretation is common in traditional information retrieval systems, as they tend to exploit their file structure in only one way. In contrast, users conduct searches using many different techniques in endless variation. From the standpoint of effectiveness in searching, as Bates [6] pointed out, the searcher with the widest range of search strategies available is the searcher with the greatest retrieval potential. Bates [21] provides a system of classification of different search strategies:

Moves are identifiable thoughts or actions that are part of information searching and are the atoms considered in this model, in much the same way as the entity is an atom in conceptual modeling. Moves can be part of a plan with a specific goal in mind, or part of a formless effort by a user who doesn't know what they are doing. An example move may remove unwanted elements from a retrieved set by introducing the `and not` operator. The moves identified by Fidel [22] provide many more examples.

Tactics are one or more moves made to further a search. A tactic represents the first level at which strategic considerations are primary. Tactics are utilized with the intention of improving search performance in some way, either in anticipation of problems, or in response to them. An example tactic for dealing with a retrieved set that is too large maybe to reject unwanted elements when reformulating the query. Bates' [23, 24] work on tactics provides many more examples.

Stratagems are larger, more complex set of thoughts and/or actions than tactics. A stratagem consists of multiple tactics and/or moves, all designed to exploit the file structure of a particular search domain thought to contain the desired information. Tools which allow footnote chasing and citation searching are examples of systems which exploit stratagems. The search methods identified by Ellis [25, 26] and Bates [6] fall into this category.

Strategies are plans which may contain many moves, tactics and/or stratagems, for an entire information search. Another analogy for a strategy is an "overall plan of attack" [27, page 408]. A strategy for an entire search is difficult to state in any but the simplest of searches, because most real-life searches are influenced by the information gathered along the way of the search. The solution to a given search problem may be characterized by a single encompassing search strategy, but many require application of interim tactics and stratagems to accomplish the overall goals of the searcher. The research done by O'Day and Jefferies [9] was completed at the strategic level.

Currently the majority of information retrieval systems communicate with the user at the move level. Strategic behavior must almost always be exercised by the human searcher. Little or no operational capability of a strategic nature is provided by current systems to the user. Lack of strategic support can make systems very difficult to use. For example, Belkin and Croft [17] noted that users find boolean queries extremely difficult to generate. This is because the user is thinking at a strategic level and the system is demanding that the user operate at a move level. These points lead to the next criterion:

Usability Criteria 6 *The trader UI should allow the user to easily exercise many different strategic search choices.*

This criterion highlights the need for trader UI developers to go beyond simple attribute searching of the offer space and provide a variety of search strategies. The most novel approaches to this problem of providing alternative search strategies have come from the information exploration tools research area [4, pages 395-438]. Famous examples include hypertext and multimedia. Other less well known examples are things like graphical boolean queries [4, pages 423-428], graphical fisheye views [28], information crystals [29], and document landscapes [30]. The limitation of the new search strategies is of course how easy the new techniques are to understand. Features which are difficult to understand will quite often be under-utilized by users, thus reducing the possible benefits they may offer.

2.4 Support for searching

Support for searching can come in many forms: no support at all, decent online help and meaningful system messages, intelligent decision support systems for information retrieval, and intelligent agents. Each of these options describes the amount of user involvement in the search and is complemented by the involvement of the system – that is to say, the more system involvement, the less the user has to do in the actual search process. Waterworth and Chignell [31] described this spectrum of involvement as the "degree of mediation".

Riddle [32] identified an important assumption about mediation that many developers like to make. Software developers often believe human time and resources are extremely scarce and costly resources, much too expensive to be wasted upon resource discovery problems. Consequently, developers try to solve every problem by automating it. Bates likes to describe the same belief as [21, page 575]: "if part of the information search process is not automated, it is only because we have not yet figured out how to automate it."

While effective systems will doubtlessly be produced, not all users will want this kind of response from an information system. For example Shniederman [4, page 66] points out that differing levels of user expertise require different UIs. Novice users prefer hand-holding menu driven interfaces and expert users prefer command line systems. In this case the degree of mediation is decreasing as the user expertise increases.

Many developers are tempted to increase the degree of mediation because they can see the apparent order in user's searching. While on the surface searches are ordered; deep down they are chaotic systems. Researchers who try to mimic human searching on computers discover that the implementation rapidly becomes too complex [3]. Bates reflects this problem in her PRINCIPLE OF COMPLEXITY [16]: "Entry to and use of an information system is a complex and subtle process." Thus we cannot make any assumptions about a user's current searching preferences and this leads to:

Usability Criteria 7 *the trader UI must respect the principle of complexity by being flexible*

Flexibility for customization is a key issue for trader UIs. The trader UI needs to provide options to let users customize the service to their own needs. Not providing potential for customization assumes that user will want to use the trader in a particular fashion; and thus would violate the principle of complexity.

3 CONCLUSION

The purpose of this paper was to identify some of the key issues in making trader user interfaces easier to use for resource discovery. Existing research has a problem in that it has been conducted largely in isolation from usability issues, particularly with regard to resource discovery. By using established knowledge from the information science field various assumptions that trader UI developers make about usability can be explored and in some cases made more realistic. To aid trader UI developers in examining their assumptions a usability evaluation scheme was developed establishing seven usability criteria.

Problems with current trader UIs stem from the assumption that searches by users upon a trader are independent of each other. But, as noted in section 2.1, searches are interconnected, as each search provides the user with new ideas and directions to follow in their next search. Thus, user satisfaction is linked to the response time of the search interface; faster response times allow users to perform interconnected searches more fluidly (see usability criteria 1). Also, it is the accumulation of search results that matters to the user and not the final retrieved set. As a result the UI should support results management and query refinement (see usability criteria 2). In addition the trader UI should not stifle the fluidness of an interconnected search, as a result the UI should be simple to understand (see usability criteria 3).

More problems stem from the way resources are cataloged in trader offer space. The trader search interface should aid the user in navigating thru the offer space by providing morphological analysis of queries and judicious use of typing capabilities to provide cross-referencing (see usability criteria 4). A popular bias among trader UI developer is to only provide a query interface; such a bias limits the users ability to effectively explore the offer space. Provision for both querying and browsing should be a feature of the trader UI (see usability criteria 5). Querying and browsing should not be the only search strategies available to the user - infact the more alternative search interfaces available to the user the more effective they can be at exploring the offer space (see usability criteria 6). The last assumption overturned by this paper is that UI developers will try to automate as many parts of the search process as possible; such an assumption is open to creating many problems as the searching process is known to be far more complex than we currently understand. As a result the trader UI should provide a great deal of flexibility and customization to allow users to search in their own peculiar ways (see usability criteria 7).

As an aside, due to the general nature of the work presented in this paper, it is not surprising to observe that the criteria are not limited in application to trader UIs, but they are versatile enough to be applied to other resource discovery tools. For example, they have been applied to Virtual Libraries on the World Wide Web [33].

ACKNOWLEDGMENTS

The author would like to thank Bob Colomb, Ashley Beitz, Renato Iannella and the anonymous referees for their helpful comments and suggestions. In addition, the author would like to acknowledge the assistance provided by the staff of the Distributed Systems Technology Centre (DSTC) and the Research Data Network Collaborative Research Centre (RDN-CRC). The author is supported by an Australian Postgraduate Award (APA).

REFERENCES

[1] C. Bowman, P. Danzig, and M. Schwartz. Research problems for scalable internet resource discovery. Technical Report CU-CS-643-93, University of Colorado at Boulder, March 1993.

[2] M. Bearman. Odp-trader. In J. De Meer, B. Mahr, and S. Storp, editors, *Open Distributed Processing, II*, IFIP Transactions C-20, pages 37–51. North-Holland, 1993.

[3] D. Ellis. The physical and cognitive paradigms in information retrieval research. *Journal of Documentation*, 48(1):45–64, March 1992.

[4] B. Shniederman. *Designing the User Interface.* Addison Wesley, second edition, 1992.

[5] Z. Milosevic, A. Lister, and M. Bearman. New economic-driven aspects of the odp enterprise specification and related quality of service issues. In *DSTC Trader Workshop*, 1993.

[6] M. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–424, 1989.

[7] G. Salton. *Automatic Information Organization and Retrieval.* McGraw-Hill, 1968.

[8] M. Hancock. Subject searching behaviour at the library catalogue and at the shelves: Implications for online interactive catalogues. *Journal of Documentation*, 43(4):303–321, December 1987.

[9] V. O' Day and R. Jefferies. Orienteering in an information landscape: How information seekers get from here to there. In *INTERCHI' 93 "Human Factors in Computing Systems"*, pages 438–445, 1993.

[10] S. Allen and J. Matherson. Development of a semantic differential to assess users' attitudes towards a batch mode information retrieval system (eric). *JASIS*, 27(5):268–272, September 1977.

[11] A. Rushinek and S. Rushinek. What makes users happy? *Communications to the ACM*, 29(7):594–598, 1986.

[12] National Information Standards Organization. *Information retrieval application service definition and protocol specification for open systems interconnection : an American National Standard.* NISO Press, 1993.

[13] C. Kuhlthau. A principle of uncertainty for information seeking. *Journal of Documentation*, 49(4):339–355, December 1993.

[14] P. Bysouth. *End User Searching: The effective gateway to published information.* Aslib, The association for information management, 1990.

[15] M. Bates. Factors affecting subject catalog search success. *JASIS*, 28(3):161–169, May 1977.

[16] M. Bates. Subject Access in Online Catalogs: A Design Model. *JASIS*, 37(6):357–376, 1986.

[17] N. Belkin and W. Croft. Information Filtering and Information Retrieval: Two sides of the Same Coin? *Communications to the ACM*, 35(12):29–38, 1992.

[18] N. Belkin, R. Oddy, and H. Brooks. ASK for information retrieval: Part I. Background and theory. *Journal of Documentation*, 38(2):61–71, 1982.

[19] G. Salton. Another look at automatic text-retrieval systems. *Communications to the ACM*, 29(7):648–656, July 1986.

[20] J. Conklin. Hypertext: An introduction and survey. *IEEE Computer*, 20(9):17–41, September 1987.

[21] M. Bates. Where should the person stop and the information search interface start? *Information Processing and Management*, 26(5):575–591, 1990.

[22] R. Fidel. Moves in online searching. *Online Review*, 9(1):61–74, 1985.

[23] M. Bates. Information search tactics. *Journal of American Society for Information Science*, 30(4):205–214, July 1979.

[24] M. Bates. Idea tactics. *Journal of American Society for Information Science*, 30(5):280–289, September 1979.

[25] D. Ellis. A behavioural approach to information retrieval system design. *The Journal of Documentation*, 45(3):171–212, September 1989.

[26] D. Ellis, D. Cox, and K. Hall. A comparison of the information seeking patterns of researchers in the physical and social sciences. *Journal of Documentation*, 49(4):356–369, December 1993.

[27] S. Harter and A. Peter. Heuristics for online information retrieval: a typology and preliminary listing. *Online Review*, 9(5):407–424, 1985.

[28] M. Sarker and M. Brown. Graphical fisheye views of graphs. In *CHI'92 "Human Factors in Computing Systems"*, pages 83–91, May 1992.

[29] A. Spoerri. Infocrystal: A visual tool for information retrieval and management. In *CIKM'93 "Information and Knowledge Management"*, pages 11–20, 1993.

[30] M. Chalmers. Visualisation of complex information. In *EWHCI'93*, pages 152–162, 1993.

[31] J. Waterworth and M. Chingell. A model for information exploration. *Hypermedia*, 3:35–58, 1991.

[32] P. Riddle. Library culture, computer culture, and the internet haystack. `http://is.rice.edu/~riddle/dl94.html`, 1994.

[33] A. Goodchild. Issues in building virtual libraries. Technical Report 310, Department of Computer Science, The University of Queensland, Australia., September 1994.