Integrating Access Methods to Text Data Files[1]

Michael A. Shepherd and Carolyn Watters
Computing Science Division
Department of Mathematics, Statistics & Computing Science
Dalhousie University, Halifax, Nova Scotia, Canada B3H 3J5

ABSTRACT

DalText is a prototype information retrieval system that permits the user to select access methods based on the task at hand. The system integrates multiple access techniques to textual data that are not maintained in traditional database management systems or information retrieval systems. The access methods available include viewing the data as a sequence of records for browsing, generating sets of records through string matching and Boolean combinations, and through generating a table schema and instantiating attribute values dynamically. As the access methods are all based on the same underlying data access model, the user can flip back and forth between the access methods in order to best accomplish the task at hand.

Introduction

Traditionally, information systems have been developed to provide effective and efficient use of given data sets. The models used for the development of these systems have been based on characteristics of the data rather than on characteristics of the information needs of the users (Kuhlthau, 1991). For example, information retrieval (IR) systems (Salton, 1989) have been developed to provide users with access to relatively unstructured textual data based on the properties of word occurrences, while database management systems (DBMS) (Date, 1990) have been developed to provide users with access to more structured data, such as employee or corporate data, based on properties and values of well defined attributes or fields within certain domains.

Typically, users map their information needs into the query language of the specific system and then interpret the results within the framework of their needs. This is the data-centered paradigm for information access (Watters and Shepherd, 1992) and is based on the premises that information needs can be expressed or reformulated as distinct queries and that all needs from a data set can be satisfied by one form of output (i.e., records of text or tables of values).

However, the access method to data should be determined by the user's task, not by the structure of the data. For example, assume a database of announcements of conferences has been downloaded from a newsgroup. If mounted in a DBMS, various attributes have to be predefined and announcements can only be retrieved if the user's information needs are reflected by the schema. For instance, the user would not be able to retrieve information

[1]This research has been supported in part by the Natural Sciences and Engineering Research Council of Canada Operating Grant OGP0009163.

about conferences with papers on fuzzy sets unless the *titles_of_papers* to be presented was an attribute field. Presumably, the user would be able to retrieve such information if the database were mounted in an information retrieval system that indexed the entire announcement. On the other hand, if an attribute in the DBMS schema were *program_chair*, then the user could more easily generate a list of names of program chairs from the DBMS system than from the information retrieval system.

The information requirements for many tasks, however, cannot be predicted nor can every use of a given data set be anticipated. Sometimes it is appropriate for the user to browse structured data for a given task and sometimes it is appropriate for the user to see a table or a list generated from textual data. Consequently, there is a shift in the focus of information systems from the data-centered models to user-centered, or perhaps, "information-need-centered" models (Hartson et al, 1990; Shepherd & Watters, 1989). A user-centered paradigm refers to information access that is driven not only by the structure of the database or its primary access system, but also by views related to the task of the user. Thus the user defines the type, amount, and structure of the data required to complete an information task. Expert system frontends, natural language frontends, hypertext, and multimedia information systems are examples of this movement towards user-centered information access system.

Cognitive Tasks and Information Access

The cognitive engineering approach to the design of information systems (Rasmussen, 1992) is to produce systems that support users in their work situations based on an analysis of the required cognitive tasks. Users tend to switch between strategies and tools in performing a task to find the best match between these strategies and tools and the required task. In addition, the formats in which the information is retrieved should match the intent of the user.

Other cognitive studies have also revealed that the required form of the output of a system is tightly coupled with the intent of the user (Crow, 1993; Palvia & Gordon, 1992). For graphical responses, i.e., pie charts, bar charts, line graphs, etc., the appropriateness of various formats has been established for a variety of tasks over the same data (Chappel, 1993). A relationship of task to output form can also be seen when dealing with textual data. For example, when the intent of the task is to *see what's going on* then browsing of full text may be appropriate. When the intent is to *see which items are about x* then perhaps a set-oriented approach but when a list of *x's with value y* is the intent of the task then a list of values or a table of related attribute values may be what the user needs to most easily fulfil the intent of the given task. In any event, what data users need and how they are presented depends largely on the current task.

Qiu (1991) developed stochastic models to study search behaviour in a hypertext information retrieval system and found that the search task has the strongest impact on search strategies among all the factors investigated. This was substantiated in a recent experiment (Watters, Shepherd, and Qiu, 1993) using the DalText system described in this paper. In this experiment, it was found that the selection of access method was determined by the task at hand and was not determined by the user's background. Nielsen (1989) compared 92 published benchmark measurements of various usability issues related to hypertext and found the two most important issues to be individual differences among users (different people will

perform very differently) and the effect of different tasks (people with different tasks will use hypertext systems in different ways).

An information retrieval system is made up of many subsystems, including the subsystem which provides an access method and the subsystem which provides a display of the retrieved items. While most information systems provide only one version of each subsystem, Frants, et al (1993) have shown that there is a need for multiversion information retrieval systems. A multiversion system provides, as its name implies, multiple versions of one or more of the subsystems of the overall information retrieval system. These systems are capable of choosing the most appropriate version of each subsystem (where multiple versions are available) to best satisfy the "problem oriented information need" as reflected by the user query. The particular subsystem investigated by Frants, et al (1993) was that of the construction of the query formulation, with feedback, and it was found that it was preferable to have a choice of different methods for construction of the query formulation.

DalText, the prototype system developed at Dalhousie University, permits the user to select the access method and display format appropriate to the task at hand. The user can employ any (or all) of three basic access methods (browsing, set creation, table generation) as needed to fulfil a given information task. The results of an access can be displayed as a data stream, a table of extracted attribute values, and/or as sets of items. In addition to allowing the search strategy to be task-driven, the highly integrated access methods and display formats of DalText allow the user to flip back and forth among these methodologies during the exploration and formulation stages of the Information Search Process (Kuhlithau, 1991).

DalText Integrated Access Methods

The DalText system provides some of the features of DBMS and some of the features of an IR system in a less rigid environment that also permits browsing of the original data file. This means that information can be found in textual datasets for which the user does not, at least initially, wish to invest the resources or time to manipulate into structured retrieval systems. Such datasets may include e-mail, newsgroups, reports in preparation, downloaded files, etc.

DalText provides the user with browsing capabilities as well as set-oriented retrieval and attribute-oriented retrieval to data streams, although with much less rigor and much more *caveat emptor* than would managed IR, DBMS, or hypertext systems. When the user drives a session the user takes responsibility for the degree of completeness and precision required with respect to a given task or purpose.

The DalText system has been designed so that the user can match the intent of the task to multiple access techniques and display formats during the searching of almost any textual data that is available. Given a data stream and an information task, the user formulates, at least abstractly, some view or concept of the organizational structure of the data within that data stream with respect to that information required to accomplish his or her information task. The more the user knows about the structure of the data within the data stream the more exact can be the view describing that structure. The user orients a specific data view towards a recognition of some task requirement.

The views considered essential in the first DalText prototype system are a sequential view, a set-oriented view, and an attribute or tabular view. The access views are truly integrated in that set queries can be made on tables and tables can be generated from the results of set queries. As the access methods are all based on the same underlying data access model (Watters and Shepherd, 1990), the user can flip back and forth between the access methods in order to best accomplish the task at hand. These access views have been described in some detail elsewhere (Hartzman and Watters, 1990; Shepherd and Watters, 1989; Watters and Shepherd, 1991; Watters and Shepherd, 1992). The following description emphasizes the integration of these views and shows how users can flip among the three access methods as they develop their search strategies for the various tasks.

The DalText architecture can be viewed as having two layers; the access layer and the data layer. The data layer consists simply of one or more unindexed files. As depicted in Figure 1, the access layer provides three integrated access and display methods. These consist of browsing one or more records in the data stream, generating sets of records meeting some criteria, and generating a universal relation table through feature extraction.

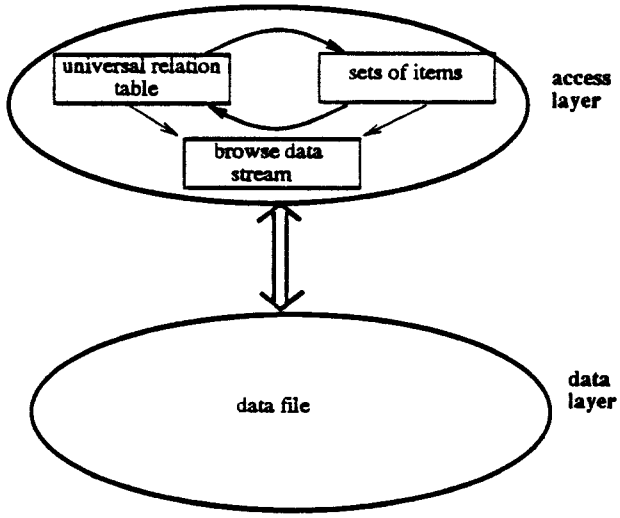

Figure 1. DalText Access Architecture

The data base for the following example is one day's worth of *The Wall Street Journal.* The example data base is 550 kilobytes, although the DalText system can handle data bases of 50 megabytes with no serious degradation in performance. The file is part of the data base distributed as part of TREC-2 (Text Retrieval Conference) which is being conducted by the National Institute of Standards and Technology and sponsored by the Defense Advanced Research Projects Agency.

Set-Oriented Access Methods

Figure 2 shows the results of both the browse and set-oriented access methods. Window **wsj_0404** is the browsing window. The user can browse the entire file from start to finish as a data stream, and can always browse forward or backward from the current position. The user has generated a set of items by doing a search for those items containing the character string, "securities". The system allows the generation of such sets and logical operations on such sets through Boolean operators. The results of set generation operations are shown in the **sets of items** window.

The items in a set may be listed at any time as shown in the **set1** window. The user may view a specific item in the **wsj_0404** window by selecting it in the **set1** window. This moves the data stream pointer to the beginning of that item in the data stream, and this change would be reflected in the contents of the **wsj_0404** window. The user may then select for display the next item (or any item) in the set, or browse forward or backward in the data stream from that position.

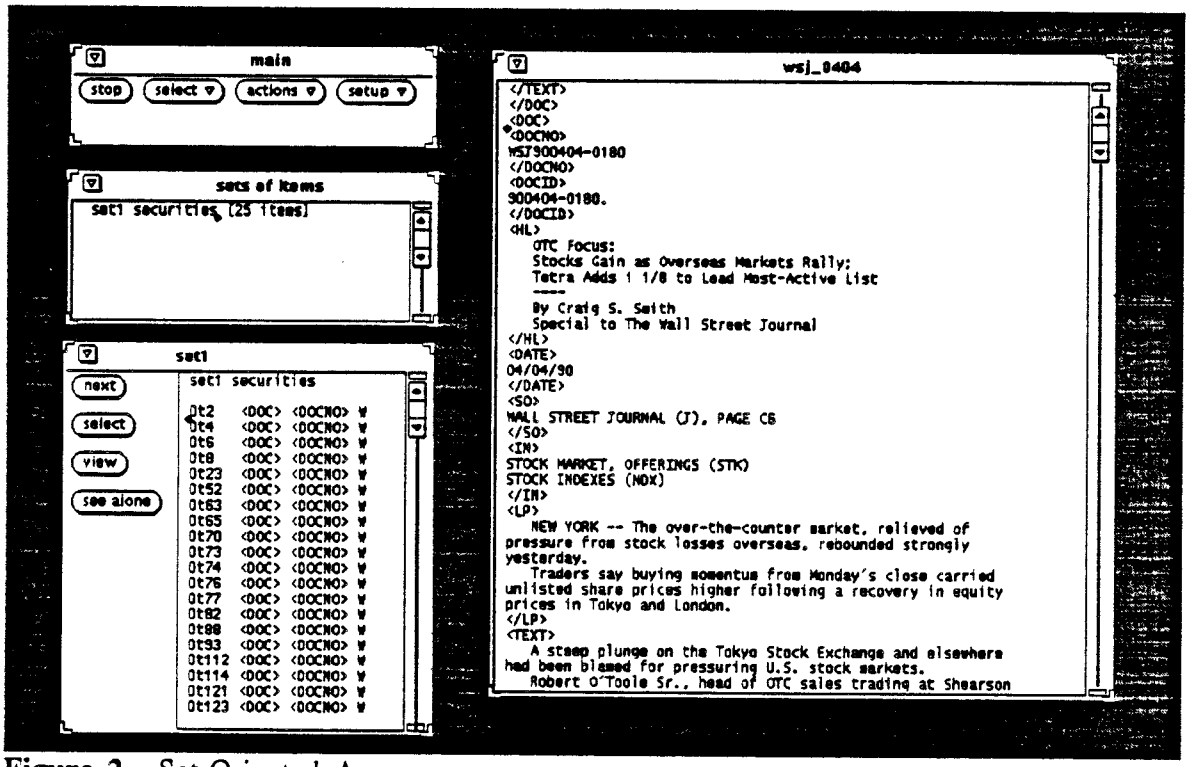

Figure 2. Set-Oriented Access

Attribute-Oriented Access Method

The user may decide that, rather than generating sets of items, it might be more appropriate to use a strategy that generates lists or tables of information. In Figure 3, the user has defined and instantiated, dynamically, a set of attributes over the entire data base. This is shown in the **UR TABLE** window. The attributes are *headline*, *gov-dept*, and *exchange*. The attribute, *DB article*, is instantiated automatically with the item identifier. As the attributes of the objects in this table are defined and instantiated dynamically, the structure of the objects will also change or evolve dynamically. The structure of these objects may be hierarchical in nature and can be shown in the **UR objects** window.

The user may view a specific item in the **wsj_0404** window by selecting it in the **UR TABLE** window. This moves the data stream pointer to the beginning of that item in the data stream, and this change would be reflected in the contents of the **wsj_0404** window. The user may then select for display the next item (or any item) in the table, or browse the data stream from that position.

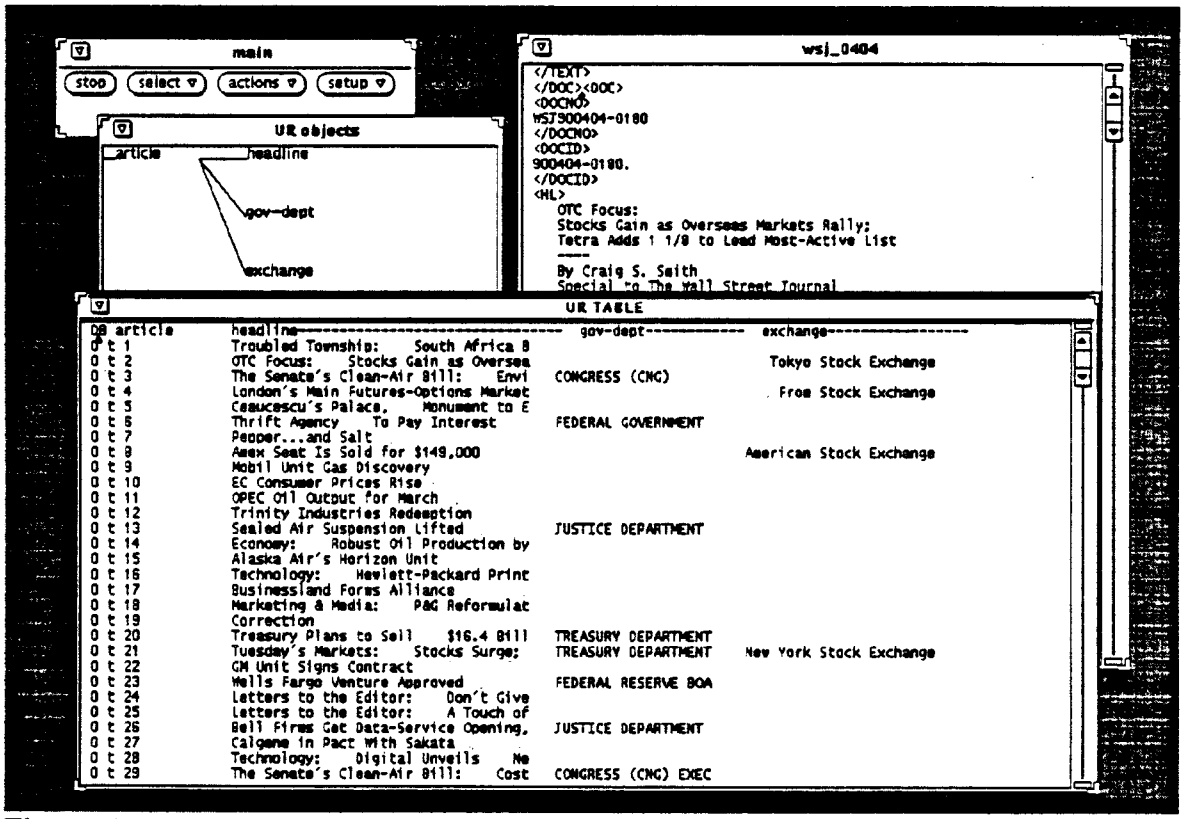

Figure 3. Attribute-Oriented Access

Imposing Attributes on Sets

Figure 4 shows that the user has chosen to impose the table view (shown in Figure 3) on the previously generated set of items containing the character string, "securities". The results of this are shown in the **set1 UR table** window. This may also be thought of as restricting the items in the UR TABLE to only those items appearing in set 11. Note that the article numbers (0t2, 0t4, etc.) are the same in the **set1** window and in the **set1 UR table** window.

The user may view a specific item in the **wsj_0404** window by selecting it in the **set1 UR table** window. This moves the data stream pointer to the beginning of that item in the data stream, and this change would be reflected in the contents of the **wsj_0404** window. The user may then select for display the next item (or any item) in the table, or browse the data stream from that position.

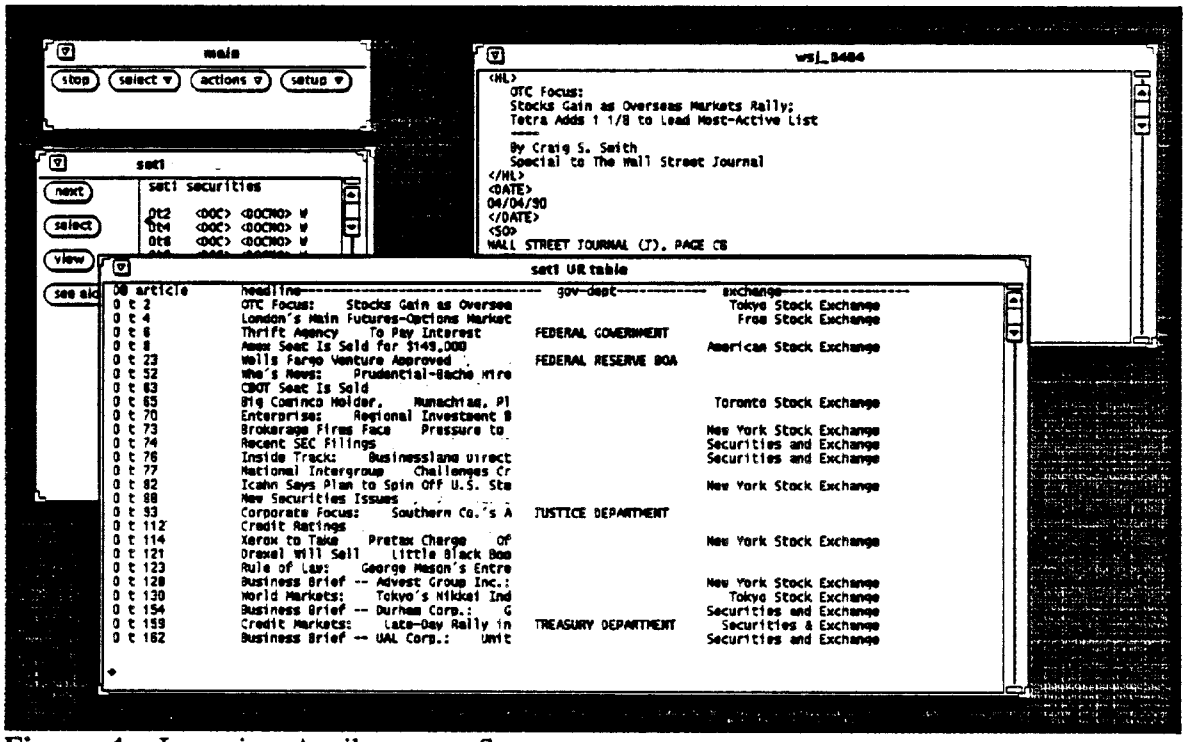

Figure 4. Imposing Attributes on Sets

Generating Sets from Attribute Values

Figure 5 shows that the user can generate a set of items in which each item has a specified attribute and value. The string, "Market", has been highlighted in the **UR TABLE** window. This has generated set 2 as shown in the **sets of items** and **set2** windows. Set 2 consists of all items with the string, "Market", occurring in the *headline* attribute or field. Again, any of these items may be browsed or displayed in the **wsj_0404** window.

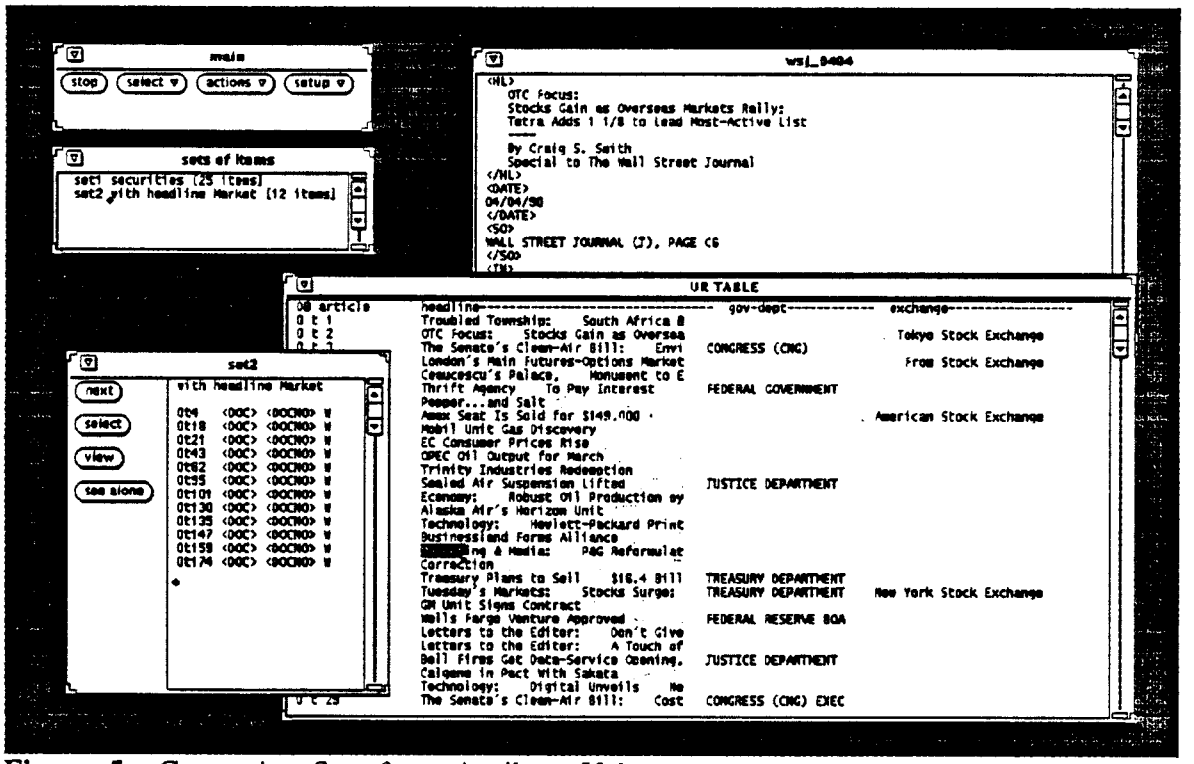

Figure 5. Generating Sets from Attribute Values

Summary

In this paper, we have described the integration of three access and display methodologies as implemented in the DalText system. An evaluation of this system found that such an integration was feasible and useful. It was found that users selected the access method most suitable to a given task when multiple access methods were available to them, independently of their backgrounds.

References

Cappel, H. & Wilson, M. (1993). Knowledge-based design of graphical responses. *Proceedings of the 1993 International Workshop on Intelligent User Interfaces*, Orlando, Florida, 4-7 Jan, pp. 29-36.

Crow, D. & Smith, B. (1993). The role of built-in knowledge in adaptive interface systems. *Proceedings of the 1993 International Workshop on Intelligent User Interfaces*, Orlando, Florida, 4-7 Jan, pp. 97-106.

Date, C.J. (1990). *An introduction to database systems*, Vol. 1, 5th ed. Reading, Massachusetts: Addison-Wesley.

Frants, V.I., Shapiro, J. & Voiskunskii, V.G. (1993). Multiversion information retrieval systems and feedback with mechanisms of selection. *Journal of the American Society for Information Science*, 44, 19-27.

Hartson, H.R., Siochi, A.C. & Hix, D. (1990). The UAN: A user-oriented representation for direct manipulation interface designs. *ACM Transactions on Information Systems*, 8, 181-203.

Hartzman, C.S. & Watters, C.R. (1990). A relational approach to querying data streams. *IEEE Transactions on Knowledge and Data Engineering*, 2, 401-409.

Kuhlthau, C.C. (1991). Inside the search process: Information seeking from the user's perspective. *Journal of the American Society for Information Science*, 42, 361-371.

Nielsen, J. (1989). The Matters that Really Matter for Hypertext Usability. *Hypertext'89: Proceedings of the Second ACM Conference on Hypertext*, Pittsburgh, Pennsylvania, 5-8 November, pp. 239-248.

Palvia, Shailendra C. & Gordon, Steven R. (1992). Tables, trees and formulas in decision analysis. *Communications of the Association for Computing Machinery*, 35(10), 104-113.

Qiu, L. (1991). *Probabilistic Models of Search State and Path Patterns in Hypertext Information Retrieval Systems*. Ph.D. Dissertation, The University of Western Ontario, London, Ontario, Canada.

Rasmussen, J. (1992). Cognitive engineering approaches to the design of information systems. *Proceedings of the 15 International Conference on Research and Development in Information Retrieval, ACM SIGIR'92*, Copenhagen, Denmark, 21-24 June.

Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Reading, Massachusetts: Addison-Wesley.

Shepherd, M.A. & Watters, C.R. (1989). Hypertext: user-driven interfaces. In *Interfaces for Information Retrieval and Online Systems: The State of the Art*, Martin Dillon (ed.) (1991). New York: Greenwood Press, 159-168.

Watters, C.R. and M.A. Shepherd. (1990). A Transient Hypergraph-Based Model for Data Access. *ACM Transactions on Information Systems*, 8(2), 77-102.

Watters, C.R. and M.A. Shepherd. (1991). Hypertext Access and the New Oxford English Dictionary. *Hypermedia*, 3(1), 59-79.

Watters, C.R. & Shepherd, M.A. (1992). *Shifting the information paradigm from data-centered to user-centered.* Technical report CS92-05, Computing Science Division, Dept. of Math., Stats & CS, Dalhousie University, Halifax, Nova Scotia, Canada.

Watters, C.R., Shepherd, M.A. & Qiu, L. (1993). *Task-Oriented Access to Data Files: An Evaluation.* Technical report CS93-01, Computing Science Division, Dept. of Math., Stats & CS, Dalhousie University, Halifax, Nova Scotia, Canada.