Users, User Interfaces, and Objects: Envision, a Digital Library

Edward A. Fox, Deborah Hix, Lucy T. Nowell, Dennis J. Brueni, William C. Wake, and Lenwood S. Heath
Department of Computer Science, Virginia Tech, Blacksburg, VA 24061-0106

Durgesh Rao
National Center for Software Technology, Juhu, Bombay 400049, India

Project Envision aims to build a "user-centered database from the computer science literature," initially using the publications of the Association for Computing Machinery (ACM). Accordingly, we have interviewed potential users, as well as experts in library, information, and computer science—to understand their needs, to become aware of their perception of existing information systems, and to collect their recommendations. Design and formative usability evaluation of our interface have been based on those interviews, leading to innovative query formulation and search results screens that work well according to our usability testing. Our development of the Envision database, system software, and protocol for client-server communication builds upon work to identify and represent "objects" that will facilitate reuse and high-level communication of information from author to reader (user). All these efforts are leading not only to a usable prototype digital library but also to a set of nine principles for digital libraries, which we have tried to follow, covering issues of representation, architecture, and interfacing.

Introduction

Computer and information scientists should be among the *first* to experiment with digital libraries. In the spirit of this recommendation, the Association for Computing Machinery (ACM), as well as other associations and publishers, are becoming involved in Project Envision, a research effort supported by the National Science Foundation to build "a user-centered database from the computer science literature" (Brueni et al., 1993). Starting with users of Project Envision at Virginia Tech and spreading to Norfolk State University and other groups and individuals across the Internet, testing will proceed regarding the applicability of digital library methods to Envision's scientific domain of computer science literature.

A goal of Project Envision is to solve some of the important research problems relating to digital libraries, especially those relating to information storage and retrieval, human–computer interaction, and electronic publishing (Fox & Lunin, 1993). Accordingly, we have identified and tried to apply a set of principles that we believe should be the basis for future national, and later international, digital libraries. The next section explains these principles.

From the proposal stages through its current prototypes, Envision is being created as a user-centered system, as specified later in Principle 8. Therefore, users are closely involved in the development of Envision, through a structured interviewing process that guided decisions about system functionality as well as through formative usability evaluation. In the third section below, "Interviews with Users," we discuss some of the more interesting aspects of our task analysis (Principle 7), based on user interviews. The fourth section describes the innovative Envision user interface design that evolved from this task analysis, and the results of usability evaluation of our user interface design for both the Envision query screen and search results screen.

In the fifth section, "Objects and Document Type Definition Development," we consider how working with "objects" (see Principles 2 and 9) can help improve the overall scientific communication process, and encourage reuse of the fruits of scholarship. This has real implications regarding representation (Principles 1–3), system architecture (Principles 4–6), archiving, and use of digital library information. Finally, we conclude by highlighting some important challenges, and summarize our plans for future work.

Principles for Digital Libraries

In reviewing early work on electronic libraries, we noted the influence of current practices in traditional libraries and publishing operations. In particular, if we consider the spectrum of representations illustrated in Figure 1, we see that common practice (that is, using paper-like page images as in Elsevier's TULIP project) may be the least useful approach for the next generation of digital libraries. Page images have all of the limitations of regular paper (problems with resizing, arbitrariness of "chunking" into pages, and so on),

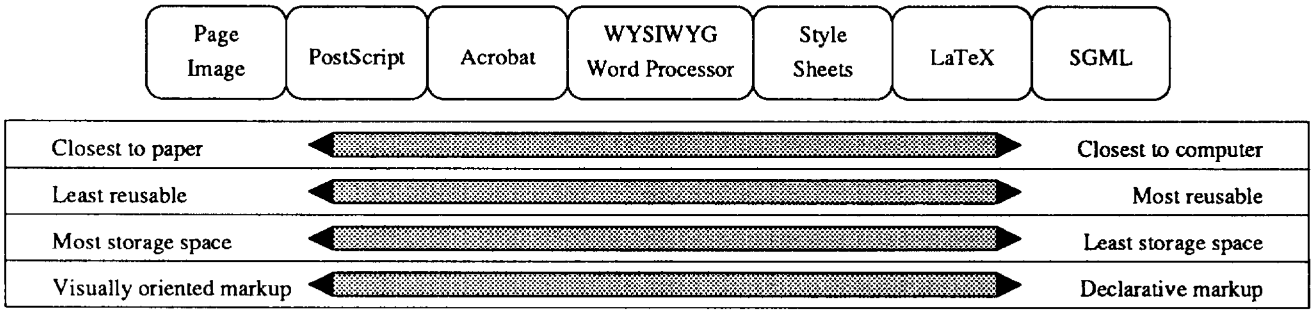


FIG. 1. Spectrum of document representations.

can only be reused by copying, require enormous amounts of storage space, consume natural resources, and are marked up in a way that virtually excludes computer exploitation of a document's organization to help in searching.

Once we go beyond our historical focus on pages, an enormous range of electronic products and services become feasible. Beginning in 1988, we worked with ACM to enlist the involvement and creativity of many individuals and groups interested in electronic publishing (Fox, 1988a). A variety of CD-ROM, hypertext, book, videotape, and online database products emerged. Then the time seemed right to take the next step, specifically, designing and building a prototype electronic archive or digital library (Fox, 1990).

Consequently, we began to reconceptualize the idea of digital libraries, as we *envision* their next generation. We aimed to harmonize and integrate concepts from a variety of interrelated fields: Artificial Intelligence (AI); distributed systems; electronic publishing; human–computer interaction; hypertext; hypermedia; information storage and retrieval; object-oriented approaches (analysis, databases, design, development, programming); and open systems. This focus led to the following set of nine principles for constructing electronic archives, arranged in three areas: Representation, Architecture, and User Interface. An earlier explanation of these principles was presented in Fox (1992).

Representation

Principle 1: Declarative representations of documents should be used. Linguistics and communication theory teach us to be concerned both with the content and form of documents. Document form is represented using one or more "markup" schemes, and the most usable scheme for electronic publishing is called "declarative" or "descriptive" markup (Coombs et al., 1987), which is supported by an ISO standard, the Standard Generalized Markup Language of SGML (Goldfarb, 1991). This approach lets us model documents as a collection of ordered hierarchies of content objects (OHCOs) (DeRose et al., 1990). Thus, the guidelines developed as part of the Text Encoding Initiative suggest markup conventions for old manuscripts, poetry, dictionaries, and other literary works. Often there are multiple OHCOs, such as one for chapter and verse, and another for page and paragraph. Note that in Biblical scholarship, for example, the former, rather than the latter page-oriented, approach is preferred. Further, the many important links (see also Principle 3) inside or among documents can be flexibly captured for increased portability using the declarative ISO standard HyTime, which is a hypermedia standard based on SGML (Newcomb et al., 1991). In summary, declarative representations of documents are feasible, and standards now exist which facilitate easy interchange.

Principle 2: Document components should be represented using natural forms, namely "objects" that can be manipulated by users familiar with those objects. When we think of documents in their most general form, specifically as multimedia "bundles" of information, it becomes clear that object-oriented representations are essential. String matching systems like PAT view documents as substrings, and basic retrieval (e.g., simple Boolean or vector or probabilistic) systems concentrate on vectors of features, so it is infeasible to ask context-dependent questions or to inquire about structure as might arise in a question about inclusion relationships. As we move to multimedia documents, which are becoming more common as multimedia technologies are refined and multimedia systems become more available (Fox, 1991), the weaknesses of these models become even more evident. In particular, multiple media must be synchronized or coordinated as well as interrelated. An ISO committee, the Multimedia Hypermedia information coding Experts Group (MHEG), deals with input, output, and interaction objects and their relationships in real-time multimedia systems. Describing and processing these documents becomes so complicated that object-oriented programming, where savings arise through inheritance, is essential. User interaction is also complicated, unless various document parts each can be manipulated as a separate object: Video or audio is played or stopped; animations are run; spreadsheets are executed with new data; simulations are tried with different parameters; algorithms are executed or animated; and three-dimensional images are rotated to provide different perspectives. Mathematical objects present unique challenges and possibilities, so that derivation or proof objects can be analyzed step by step, or formula objects can be visualized in various ways (Wolfram, 1991). Clearly, representations using objects that are convenient for users will allow authors to communicate more directly with readers, rather than going through the awkward, low-level medium of paper.

Principle 3: Links should be recorded, preserved, organized, and generalized. As we integrate documents into very large collections covering an entire scientific domain or professional area, links among those documents become increasingly important to help with search and browsing. Groupings of those links into paths, threads, tours, and webs are essential for organizing, personalizing, sharing, and preserving the structural, interpretational, and evolutionary connections that develop. We are beginning to see the emergence of wide area hypertext systems (Yankelovich, 1990) like the WorldWideWeb (WWW), that carry this concept forward into a distributed environment. Clearly, we must coordinate hypertext and hypermedia linking with the various approaches to search and retrieval (Fox et al., 1991b). One approach is the idea of information graphs (including hypergraphs), where objects of all types are interrelated by links or arcs that capture not only citation (reference) but also inheritance, inclusion, association, synchronization, sequencing, and other relationships. By specializing object-oriented databases to this task, we are building a foundation for next-generation integrated retrieval systems (Chen, 1992). Our work with the Large object-oriented External Network Database (LEND) system and methods for querying information graphs (Betrabet et al., 1993) is along these lines, as are other efforts to build systems for managing information graphs (Glyssens et al., 1990; Paredaens et al., 1992). Clearly, adaptations of hypertext (link) and semantic network (AI) concepts are essential for digital libraries.

Architecture

Principle 4: There should be a separation between the digital library and user interfaces to it. To serve millions of users, with their diversity of backgrounds, talents, and needs (see "Interviews with Users"), a variety of user interfaces will be needed for digital libraries. With hardware limitations and variations, there are a host of other reasons for building user interfaces that are particularly suited to common environments. Thus, in Project Envision, we have development efforts underway for Macintosh (specifically, both 13-inch and megapixel displays), X/Motif, and NeXT-step user interfaces. Earlier reports (Nowell & Hix, 1992, 1993a, b) and the discussion in the section on interface design below explicate these issues. With all this necessary tailoring of interfaces, it is clearly much easier if the system architecture is such that the digital library itself can be decoupled and developed separately. Common parlance refers to a *client* system running the user interface, a *server* system managing access to the digital library itself, and a well-defined *protocol* organizing the communications required between the two. In the case of digital libraries it makes sense to begin with the Z39.50 protocol that was originally devised for communication between library catalog and bibliographic database systems. That is the approach taken in the popular Wide Area Information Server (WAIS) system (Kahle et al., 1993). We believe that further generalization is needed, so that information objects and their links can also be communicated, and we have been developing an Envision protocol to test that idea. In our case, then, we have Envision client software to manage the user interfaces, an Envision protocol, and the main (distributed) Envision system.

Principle 5: Searching should make use of advanced retrieval methods. At the heart of digital library systems like Envision, there must be support for searching, browsing, following links, presenting selected information, and other services. Regarding searching, our experimental studies, and others recently completed in connection with the 1992 Text REtrieval Conference (TREC), indicate that advanced retrieval methods can be more effective then conventional Boolean approaches. Our work with hundreds of thousands of library catalog records indicates that users prefer vector and feedback methods to standard Boolean searching (Fox, 1988b; Fox et al., 1993). These approaches can be further extended through the use of frames (Weaver et al., 1989) and other representations to get closer to "concept searching." On the efficiency side, advances in hashing (Wartik et al., 1992) can improve performance in ordered dictionaries (Fox et al., 1991a). In many indexing, linking, and other situations, guaranteed direct access to large collections, given a desired key, can be supported by rapidly finding minimal perfect hash functions (Fox et al., 1992a, b). With all these possible benefits, future digital libraries should certainly be designed to use the most advanced retrieval methods possible.

Principle 6: Open systems that include the user, and where (some of) the functions of librarians are carried out by the computer, must be developed. As digital libraries emerge, and become directly available to end-users, it is important not only to improve the user interfaces, but also to provide assistance to users like that offered by experienced librarians and search intermediaries. One approach is to develop distributed expert-based information systems, building upon studies of user-intermediary protocols (Belkin et al., 1987). Specifying the user's information need or problem, modeling the user, specifying the subject domain, and managing the overall dialog are of particular importance. Our COmposite Document Expert/extended/effective Retrieval (CODER) system was designed along these lines (Fox & France, 1987; Fox, 1987). Other efforts in this regard suggest that, while development is difficult and time-consuming, such an approach may be of value when large numbers of users are involved. We hope that experienced intermediaries will become involved in expert system projects to pass on their guidance to millions of end-users.

User Interface

Principle 7: Task-oriented access to electronic archives must be supported. Current efforts to build prototype digital libraries are often focused on a particular subject domain, in part because of support provided by associations or publishers. Thus, the CORE project (involving the American Chemical Society and Chemical Abstracts Service, as well as Bellcore, Cornell, and OCLC) deals with the chemical literature (Lesk, 1991). Part of the hope of that project is to have access to the chemical literature be a key feature of a "chemist's workstation." Supporting the research, referencing, writing, and educational activities of staff in a university chemistry department can be viewed as providing task-oriented access to information suitable for each of those types of activities. We believe that in addition to having user interfaces that support information access as a separate activity, with its aspects of searching, browsing, previewing, and so on, "embedded information access" must be enabled. For example, a chemist preparing a class or conference presentation should be able to escape from a tool like PowerPoint™, find a description of an important reaction, grab the registry number and structure diagram for one slide, extract a table showing yield for another slide, and return directly to the expanded presentation. Similarly, a programmer accessing the Envision archive should be able to interrupt a programming effort to find a useful algorithm from *Collected Algorithms,* and add it as a subroutine, along with capturing some of its documentation and pointers to more information. We hope that efforts of this type will proceed in similar fashion to how computers in cars, microwave ovens, and compact disc players now support rather than interfere with users' tasks.

Principle 8: A user-centered development approach should be adopted. Since workstations are often devoted to individual users, we must make them serve those users. We should turn our system development efforts around to be centered on the users, rather than on the machine. Without this focus on the user, we may well produce digital libraries (and other interactive systems) that can compute perfectly and quickly, but cannot communicate effectively and efficiently with their users. As we learn more about design and development of interfaces (Hix & Hartson, 1993), a user-centered approach becomes more feasible. The next two sections explain our efforts in user-centered design of the Envision system.

Principle 9: Users should work with objects at the right level of generality. If we follow Principle 2, our digital libraries will represent information in terms of usable objects. With advanced search methods such as those called for in Principle 5, we can search, browse, and preview those objects. Further manipulation should be supportive of user tasks, as called for in Principle 7. We consider all of these issues further in the section on objects and document type definition development.

The following sections discuss many of these principles further, focusing on users, user interfaces, and objects. For more general information on Project Envision, the reader is referred to Brueni et al. (1993).

Interviews with Users

In accordance with Principle 8, we began by focusing on potential users of a digital library of computer science literature, such as Envision. Over a four-month period we interviewed 12 professionals in the areas of computer science and information retrieval. Interviewees were chosen carefully to broadly represent the type of user we expect for Envision. During intensive interviews lasting from one to two hours, interviewees responded to questions focused on four topics:

(1) Current information retrieval practices.
(2) Current information dissemination practices.
(3) Desired information retrieval and manipulation capabilities.
(4) Demographic data.

When seeking publications relevant to a particular topic, most of our interviewees have used electronic information systems of some kind. These include computerized library catalogs, CD-ROM systems, and online search services. However, our interviewees found existing systems difficult to use for a variety of reasons. Inadequate access to any electronic information system is one major problem. Indeed, the feature most requested by interviewees for a new information retrieval system is access from the workstation in their own offices.

Interviewees also complained about the difficulty of structuring queries, the number of diverse user interfaces, inadequacy of feedback about unsuccessful searches, and the amount of knowledge required before systems are really usable. Our interviewees generally disliked any requirement or need to consult a human intermediary, or search system expert, to access the literature.

Most interviewees specifically requested or implied the need for full text retrieval. Other features commonly requested include:

- Access to multiple forms of information (abstract, resume, brief description, full text, bibliographic entry) about each document retrieved;
- Print capability;
- User annotation facilities; and
- Ability to establish and work within a personal subset of the database.

A *usable* interface was mentioned often as a needed feature, and complaints about the user interfaces of existing electronic information retrieval systems were frequently cited reasons for not using those systems.

Our interviewees want the ability to explore patterns in the literature. One spoke at length about the "community of discourse," or invisible college of people carrying on conversations in print, all reading what the others have written. Others spoke of citation indexes, reference tools that reveal patterns of citation within the literature, so that works evolving from major articles may be identified. Ability to locate seminal documents, those which have been widely cited, is needed. Interconnections in the literature are of widespread interest. People want to use hypermedia linking to navigate among documents with common patterns of citation and to follow chains of reference among documents. In essence, they want to be able to follow on-going "conversations" in the literature.

Browsing was another common theme. Users want to be able to explore the literature along dimensions of their choosing, to home in on particular areas of interest and explore those in detail, then move on to broader views, or sometimes different views. For some, browsing includes the ability to examine the structure of documents, not just the citation or the abstract. Users want to identify a section of interest in a document and zoom in on it for closer examination and more details. Access to tables of contents provides part of this capability, but users want to move seamlessly between the table of contents and the body of a document. They want to see structure at a finer granularity than a table of contents allows. Capability to search document structures is wanted, so that chapters, illustrations, graphs, or sections of code might be located, not just whole documents by title, subject, or author.

For some, browsing is a luxury rarely permitted by pressures of time. These users want the ability to locate a few critical items of interest and be protected from the rest. They are especially interested in powerful filters to eliminate "junk" and allow them to easily locate only the most highly relevant materials. Offered the possibility of a system regularly scanning the literature for them and notifying them of new publications of probable interest, they were fearful of being overwhelmed. Information overload was cited as a reason for avoiding Internet discussion groups and bulletin boards.

Interviewees shared reliance on journals and conference attendance as major sources of information, with additional attention to conference proceedings. Talk with colleagues was ranked as equal in importance to journals as a source. Colleagues are especially helpful in providing pointers into the literature, that is, specific references to works likely to be helpful in solving a particular problem or to be of particular interest. One interviewee indicated that colleagues serve as valuable filters; they point to the few best works in an area without providing an exhaustive list of less valuable materials. A few interviewees make use of network bulletin board services, but most do not.

Interviewees indicate that they rarely use videos, because of the inability to browse or skim video, which is seen instead as an "all-or-nothing" experience. Users are frustrated by the difficulty of locating particular segments of video that are of special interest. They would like to see a "video table of contents" and to be able to create hyperlinks to and from specific video frames.

We asked our interviewees about objects of interest in the computer science literature. They spoke of the obvious entities: books, journals, articles, videos, bibliographies, even figures and tables. People are also objects of interest, as authors, as researchers, as colleagues. Research projects, funding sponsors, conferences and workshops, and various types of institutions are also objects of interest. Additionally, programs, data structures, algorithms, animations, programming languages, hardware devices, interesting problems, and concepts are entities the users wish to manipulate. Users want access to source code, ideally in a choice of languages. They want to be able to embed the code in their own programs for testing and use with their own data, without rekeying the code. They would like access to analytical data about algorithms, to explanations by experts, and to animations that increase comprehension.

Design and Evaluation of the Envision User Interface

Responding to interviewees' concern that an information retrieval system must be accessible from their offices, our design is based on the premise that the Envision user interface will run as a client process on a user's desktop computer, communicating with the Envision retrieval system via network. Our user interface designs provide flexible use of varying configurations of monitors, both in size and number of displays. The lowest configuration supported uses a single 13-inch gray-scale display. With larger or more monitors, tiling of windows becomes feasible, and it is easier to work with full-text or page-image retrieval.

Our interface specification calls for separate windows or groups of windows for each of the major phrases or types of interaction with the Envision system. These include:

- Query Window (with four query fields and a query history);
- Search Results Windows (Graphic View, Item Summary, Item Preview); and
- Browsers.

The next two subjections deal with the Query and Search Results Windows (see Fig. 2), respectively. Work on the Browsers will be reported in a later publication.

Envision Query Window Design

The Envision Query Window design gives users the benefits of natural language query formulation (i.e., no complex syntax or use of logical operators is required, nor is knowledge of an artificial indexing language), while also providing the means to restrict searches. The Query Window has two categories of use:

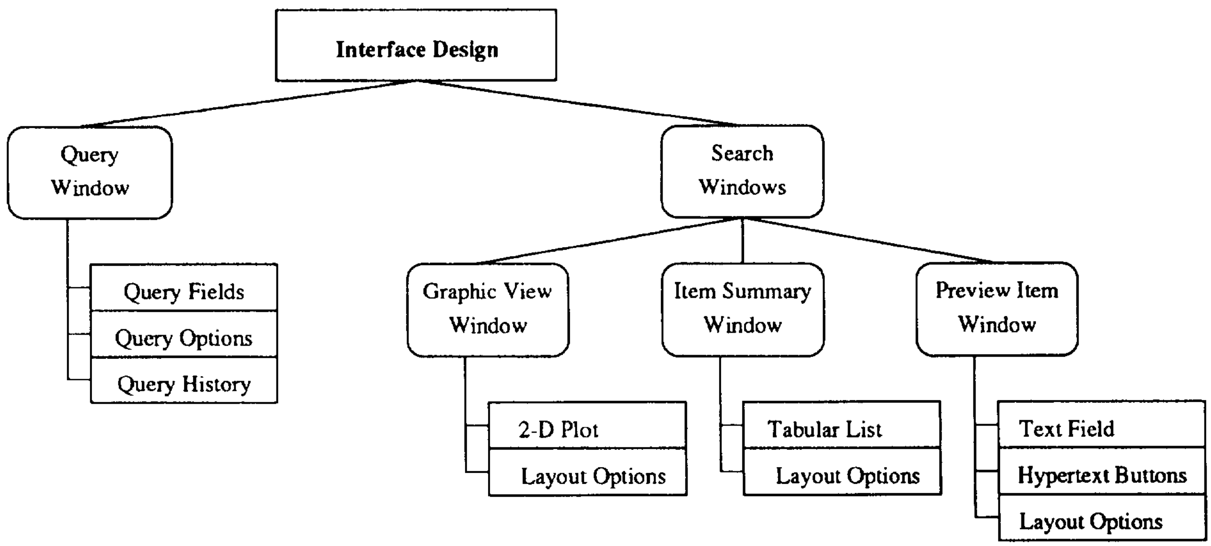


FIG. 2. Main components of Envision's user interface design.

- New queries are created and searches performed from this window.
- Access to previously completed (old) queries and the results of the related searches are provided. Old queries may simply be viewed or they may be revised and used for another search. Results of searches from old queries may also be redisplayed via a query history feature.

The Query Window offers a user three ways to create new queries:

- By entering document descriptors in four new query fields for authors, title words, words related to content, and words found in other parts of the document as specified by a pop-up menu labeled "Special Query."
- By editing earlier queries.
- By combining results of previously completed searches, using set operations.

Query Fields. The Query Window, shown in Figure 3, features four query fields for Authors, Words in Title, Content Words, and Special Query, plus a Query History field. The Special Query field has a pop-up menu control that allows users to specify searches of other document parts—abstracts, chapter titles, figure headings, and tables of contents, as well as to enter a complete bibliographic citation as a single block.

When creating a new query or editing an old one, the user may make changes in addition to or instead of simply editing the text in the four fields. Other options include changing the matching types (explained further below) used for each field, changing the relationship among fields, and changing filters that restrict search results. The filter controls—for publication year, publication language, number of items to be found, and type(s) of items desired—are at the bottom of the window.

As shown in Figure 3, matching type options are given to the right of the query fields, with text and related radio buttons. Users have control over whether terms within a given field are "ANDed" ("Match all...") or "ORed" ("Match any..."). In some cases, as for Word(s) in Title, and user may specify that the order of terms in the query must be matched as well. The relationship among the four query fields is also user-controlled via radio buttons below the group of fields.

Query History. As queries are stored or related searches are performed, the user establishes a history that is accessible through the Query History field across the top of the window, shown in Figure 3. In the Query History, a one-line summary form of each query is displayed in order by query number, along with the number of items retrieved by the related search. The Query History provides access to the results of previous searches, means to redisplay the full content of previous queries for possible revision, and a mechanism for combining the results of completed searches.

Formative Usability Evaluation of the Query Window Design. Prior to building an interactive rapid prototype, the Envision Query Window design was modified several times as a result of critique sessions with prospective users, using paper versions of the design. Sessions with the Human–Computer Interaction Research Group at Virginia Tech were particularly productive. An Aldus SuperCard^{TM} prototype was then created on a Macintosh and used for the formative usability evaluations described below.

The foremost goal in our usability evaluation of the Query Window was proof of concept: We needed to verify that users could understand how to formulate a complicated query using this window and also how to formulate revised queries based on previous queries. Usability evaluation was conducted with four participants (a reference librarian and a Computer Science Department undergraduate student, graduate student, and faculty member).

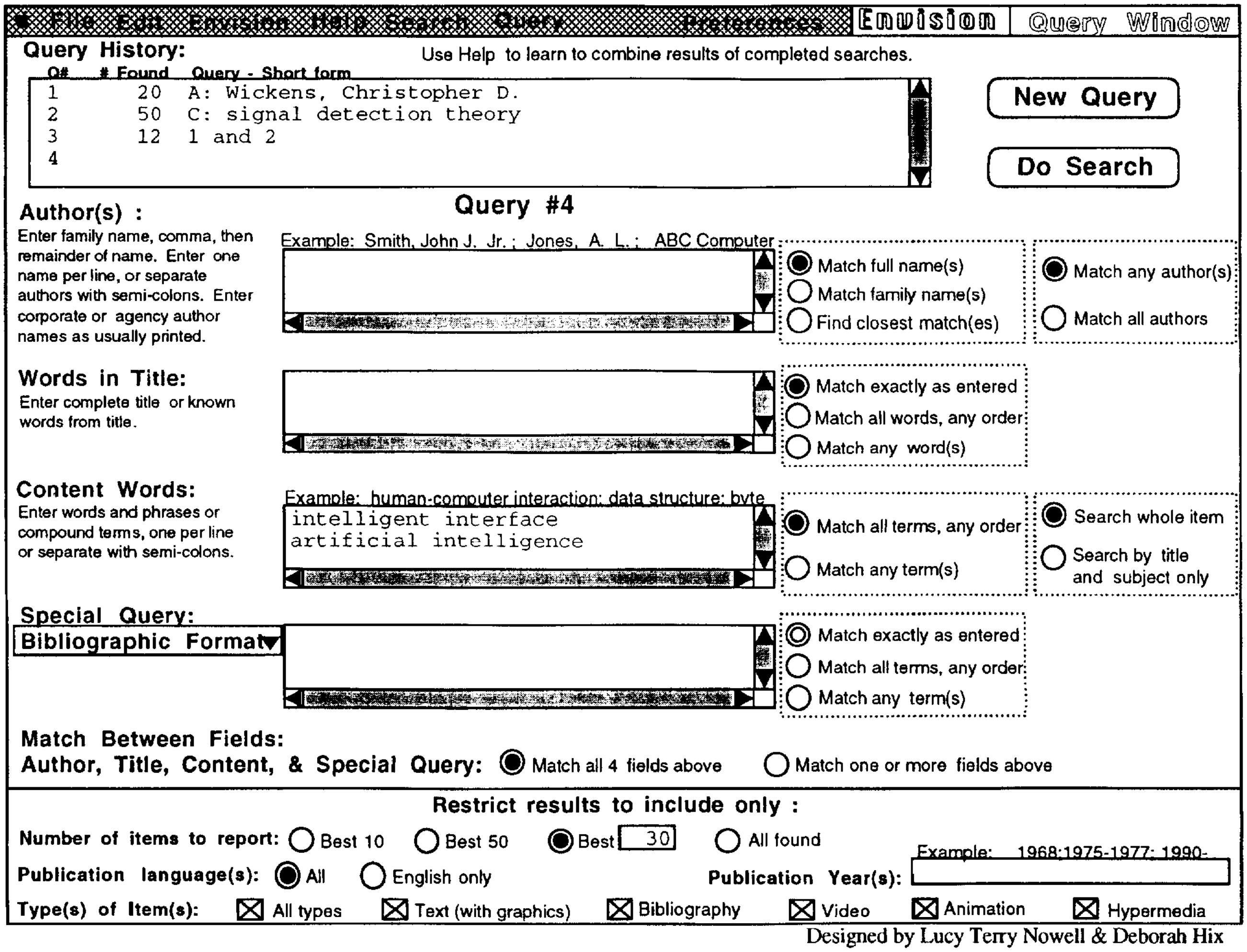


FIG. 3. Query Window screen.

We formulated eight benchmark tasks, each with several subtasks. The tasks revolved heavily around formulating new queries, modifying previous queries, and changing parameters on searches. Prior to evaluation, we established 63 usability goals based on objective measures of time for task completion, number of errors, and number of uses of Help. We had 35 usability goals based on subjective measures from a questionnaire adapted from QUIS, the Questionnaire for User Interaction Satisfaction (copyright 1988–1991 Human–Computer Interaction Laboratory, University of Maryland at College Park).

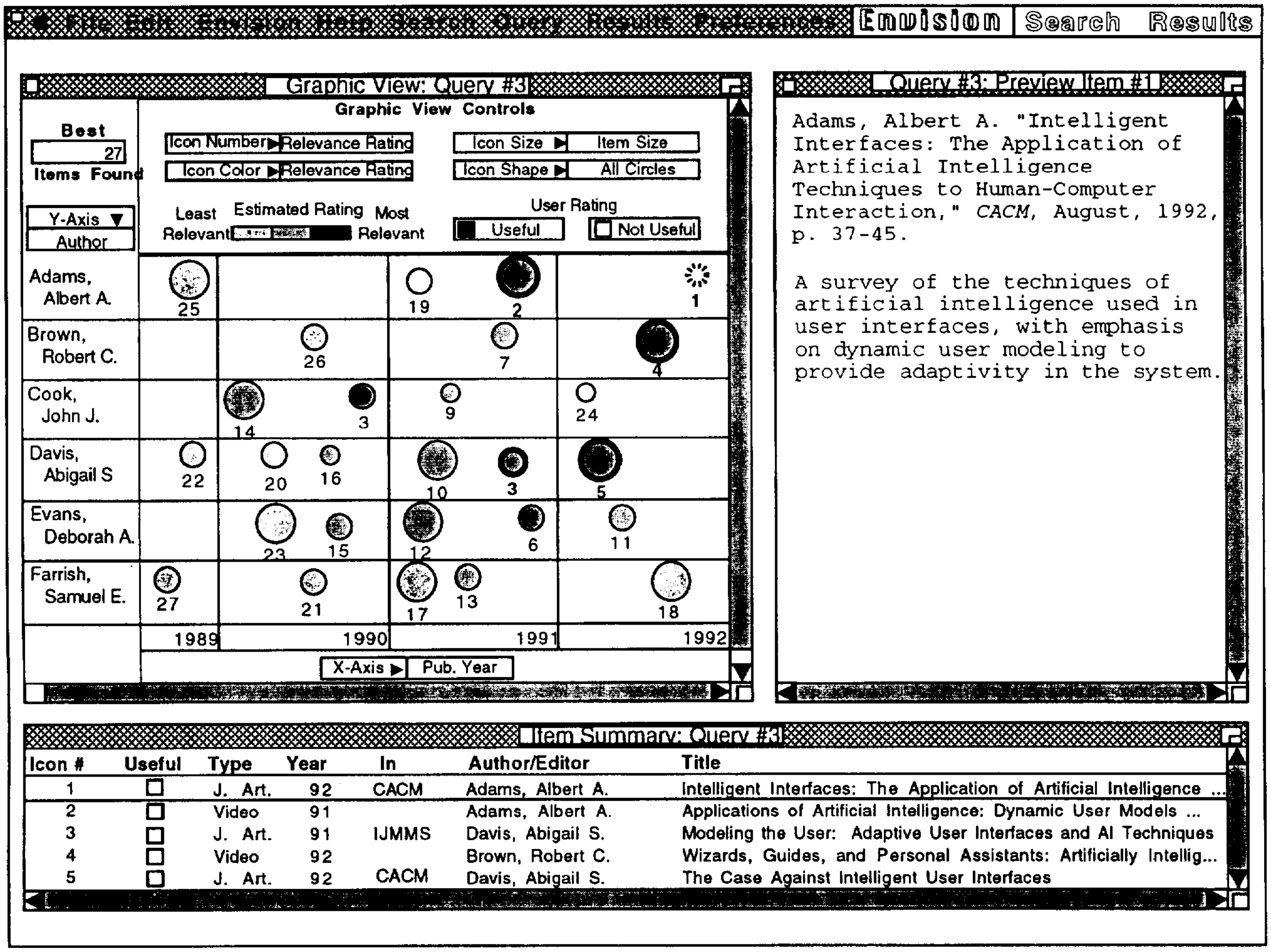
Across all participants, quantitative data were very encouraging. Average times for task completion equaled or exceeded our planned usability goals for ten subtasks and our worst acceptable case for an additional 11 subtasks; ten average times did not meet our worst acceptable case. Also, all 16 counts of errors and 16 uses of help met or bettered our usability goals for worst acceptable case. (Note that 16 counts of errors and also 16 uses of help is coincidental.) For the 35 subjective questions, responses were on a scale from a worst of −3 to a best of +3. Across all participants, there were no negative average responses.

Qualitative data were equally pleasing. All participants expressed their satisfaction with the design, stating that it made query formulation easy and fast. Participants commented that they particularly liked the control over the search provided by the interface, including the restrictions they could place, and the large amount of information displayed. They suggested some minor changes to the interface: wording of messages on the display, use of a larger font for some text, and small improvements for a couple of buttons.

Only minor design changes were made to the Query Window as a result of usability evaluation. These changes are already reflected in Figure 3, including use of a larger font for some example and prompt text. Labels for some buttons have been reworded (notably for buttons controlling matching rules for each query field), and one control button (Clear Fields) was eliminated.

Envision Design for Search Results

In designing the Envision user interface, we wanted to get away from the conventional approach of present-



Fictitious data shown.

Designed by Lucy Terry Nowell and Deborah Hix

FIG. 4. Search Results screen.

ing search results via scrolling text-only lists, which are cumbersome at best and of low information content at worst. Thus, the Envision design attempts to use innovative graphical presentations of search results to users.

Graphic View Window. Central to the search results display design is the concept of viewing each document (item) as a node within the Envision database graph and representing the document graphically as an icon. Results of a search are presented in a Graphic View Window as a scatterplot of icons, as shown in Figure 4. This sample screen presents the results of a search for which the query has been issued in the Query Window as described in the previous sections. The search results design provides a graphical, direct manipulation presentation of documents found by the search, and provides users with control over the semantics of six attributes of the Graphic View icons:

- Icon placement along the x-axis and y-axis;
- A number associated with each icon; and
- The size, color, and shape of the icon itself.

By manipulating the icons, users may perform a variety of functions:

- View basic bibliographic information in the Item Summary Window;
- Examine the item abstract or other short description in the Preview Item Window;
- Use the document represented as the basis for a feedback search; or
- Print or save selected information pertaining to the item.

A single icon is selected by clicking it. Groups of icons may be selected by depressing the shift key and successively clicking on icons, or by using a drag-selection technique—depressing the mouse button while enclosing the icons of interest within a "rubberband" selection box.

Item Summary Window. Textual summary information is displayed in the Item Summary Window, shown at the bottom of Figure 4, when an icon in the Graphic View Window is selected. The icon number at the beginning of each summary line provides a mechanism for associating each summary line with its icon bearing the same number in the Graphic View Window. Users have control over the content and layout of the Item Summary Window through a dialog box accessed from the Results menu at the top of the screen.

Preview Item Window. The user may view a retrieved document's content within the search results screen, through the Preview Item Window shown in the upper right of Figure 4. The Preview Item Window displays full bibliographic citation data, as well as an abstract or short description. However, users may opt to display other information in the Preview Item Window, including *Computing Reviews* categories, table of contents, a storyboard for video, and so forth. The full content of text items may be displayed in the Preview Item Window, but specialized document viewers are planned for display of videos, animations, and other document formats.

Formative Usability Evaluation of the Search Results Design. Like the Query Window design, the first version of the user interface design for the display and use of search results has been prototyped on a Macintosh using SuperCard, and we also have performed extensive usability evaluation on it as described below. Similarly, the Envision Search Results Window design was modified several times before the interactive prototype was built. Again, prospective users and members of the Human–Computer Interaction Research Group at Virginia Tech critiqued paper versions of the design.

The foremost goal in our formative usability evaluation of the search results windows was—as with evaluation of the Query Window—proof of concept: We needed to verify that users could understand relationships among the three windows and the graphic and textual objects within them. We also needed to establish user interest in this type of graphical, direct manipulation interface for viewing search results, which have traditionally been presented as lists of text.

Usability evaluation was conducted with three computer scientists (one of whom is also a library automation programmer) as participants. We formulated eight benchmark tasks, with a number of subtasks. Included were such tasks as locating the title and author of the most relevant work and finding three works published by a given author in a specified year. We established 42 usability goals based on objective measures, specifically measures of time for task completion, numbers of errors, and number of times Help was used. We also had 28 usability goals based on subjective measures (again adapted from QUIS).

Across all participants, quantitative results were very good. Average times for task completion equaled or exceeded our planned usability goals for 13 of the 16 measures of time. For 14 counts of errors and 12 counts of Help usage, all measures met our usability goals. For 26 of the 28 subjective questions, the average user was positive, on a scale of -3 to $+3$. Both negative responses pertained to the Help system, which was present in only a rudimentary form.

Qualitatively, all three participants were strongly positive in their evaluations of the design concept. They liked the variety of information presented visually in the Graphic View Window and the possibilities offered for customizing the presentation to their own needs. All commented positively on the power the interface provides the user. None of the participants had any difficulty in recognizing relationships among the windows and the objects in them.

Only minor changes in the design were required as a result of usability evaluation. These included changing the font for one label from plain text to bold and changing the layout of the control buttons in the Graphic View Window.

Future Work on the Envision User Interface Design

Participants evaluating the Envision Search Results Window told us they want access to the query while they look at the search results. They want to be able to revise or refine the query, in addition to examining it in relation to the results. Work is in progress to develop a smaller version of the Query Window for simultaneous display with the Search Results Windows. Users will be able to toggle between the full-sized Query Window, with the complete set of features described above, and the smaller window, which will provide only basic functionality. Other features still under design include user annotation facilities, extensions to support personal subsets of the database, hypermedia capabilities, and browsers for both the database as a whole and individual documents.

Objects and Document Type Definition Development

One of the focuses of the Envision project has been defining and using the objects present in computer science, in accord with Principles 2 and 9. As shown in Figure 5, we have taken a three-pronged approach to identifying objects: interviewing users, using those known from previous systems such as bibliographic databases (e.g., people, publications, and so forth), and a bottom-up approach wherein we examine items such as journal articles to try to identify other significant objects.

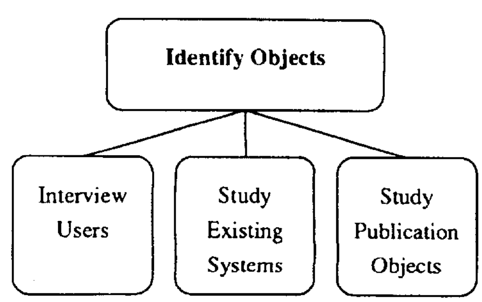


FIG. 5. Approaches to identifying objects.

This overall process, involving object-oriented analysis, is iterative. As we obtain and use more real-world objects, we learn more about the objects, and revise our model. Object-oriented analysis is used to define the external representation of objects in the form of SGML document type definitions (DTDs). Also, it leads to an internal representation in the form of persistent objects in LEND, as described below. Figure 6 illustrates this process of object-oriented analysis that leads to both external and internal object representations.

External Representation of Objects

An object-oriented analysis produces a theoretical model of real-world objects. This theoretical model is first mapped onto a concrete syntactic model for representing instances of objects. For Envision, we have chosen to represent objects in an external form using SGML and HyTime. These standards permit us to represent objects as ordered hierarchies of content objects (OHCOs) using descriptive markup as recommended in Principle 1. Principle 3 is supported by HyTime's hypermedia linking. As widely adopted international standards, the SGML formats act as an interchange mechanism for Envision documents.

The design of SGML document type definitions is facilitated by literate programming (Brueni, 1992). Literate programming allows the task of writing documentation describing the design, semantics, and example use of DTDs to parallel the actual construction of DTDs. Our approach centers on the task of writing reports that describe context-free languages using grammars. The high-level literate programming notation (itself defined by an SGML DTD) supports modular design and enforces DTD design guidelines. Accompanying software produces typeset reports, similar grammar notations (such as BNF), parsers, and summary reports, each derived from the original grammar notation.

Internal Representation of Objects

The document type definition (DTD) provides an external representation of the hierarchical structure of a document suited for flexible presentation to a user. To place the document in the larger context of the computer science literature, we embed this external representation in an internal representation of an information graph (Lavinus, 1992). An information graph is a database of persistent objects organized as a graph, as discussed in regard to Principle 3. The graph structure allows for explicit representation of the relationships present in the computer science literature, such as

- A person *is an author of* an article;
- One article *is cited by* another article; and
- A particular computer science subject area *is germain to* an article

We have pursued theoretical investigations regarding the most effective ways to store an information graph (Lavinus, 1992) and the most efficient ways to access its objects (Fox et al., 1992a).

Chen (1992) has implemented the LEND database model for representing an information graph. Objects in LEND are implemented as C++ objects. To provide an efficient and flexible capability to query the database, the Graph Object Access Language (GOAL) has been developed as a part of LEND (Bertrabet et al., 1993). Queries in GOAL return subgraphs of the information graph; that is, the results are structured data. The result is a query language that makes the most of the powerful information graph representation.

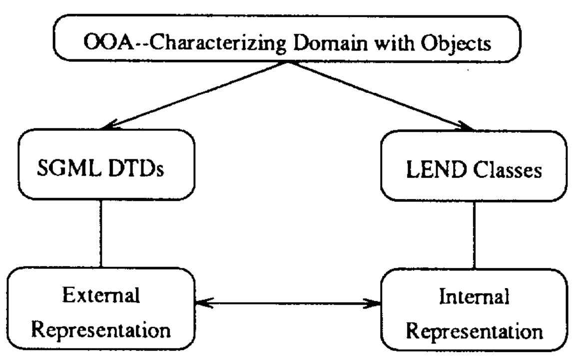


FIG. 6. Object-oriented analysis of computer science domain.

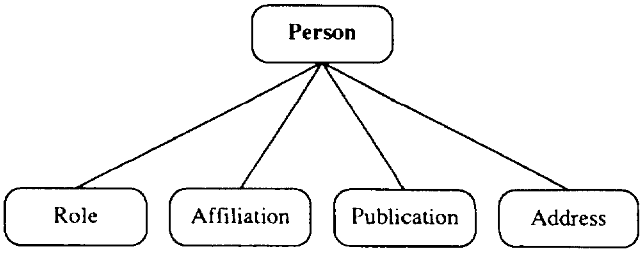
Example

To get a complete picture of the mapping between the internal and external representations for an object instance, consider a Person object as in Figure 7. The external (SGML) representation parallels the structure provided by an object-oriented analysis, which (although not shown) is paralleled in the internal (LEND) representation. Data available from real-world sources include the ASIS member directory, ACM and IEEE member files, X.500 directories, as well as other information either obtained or derived from existing data. Object-oriented analysis produces a Person object including attributes such as name, areas of interest, current affiliation(s), preferred contact address(es), and relationships including affiliation to organization(s), papers published, edited, or reviewed, projects involved in, and so on.

Development Status

We have analyzed objects like:

- Full-text journal articles,
- Bibliographic citations,
- Mathematical formulas,
- Theorems and proofs,
- Algorithms, and
- Program code.




```
<person>
   <role>Program Chairman</role>
   <fnm>Edward</fnm>
   <snm>Fox</snm>
   <affil>Department of Computer Science
          Virginia Polytechnic Institute and State University
   <title>SIGIR '95</title>
   . . .
<person>
```

FIG. 7. A Person object.

We have begun to represent them concretely as SGML and C++/LEND objects. Future DTD development will include the analysis of various multimedia objects, ranging from PostScript figures to bitmap images to MPEG video. We will also focus upon hypermedia functionality and fine-grained analysis of computer science objects.

Summary

Project Envision is providing us with an extremely rich testbed to investigate some of the very interesting problems relating to digital libraries. In this article we noted nine principles that we have followed and that we believe are important for next-generation digital libraries. Further, we explored some of those principles as they relate to user-centered design, user interfaces, and use of information objects within Envision. Results of user interviews were given which continually guide our research and development. Two of the innovative screens used in the Envision interface design were discussed, and details of our formative evaluation were given. Finally, we presented issues relating to our work with information objects, and how they can be represented externally for interchange and internally for efficient processing.

Many challenges remain for Project Envision. As we pass the halfway mark of our original three-year study, we have almost enough material converted into our database representational form to achieve critical mass in making Envision interesting to use. We are also close to having the key screens for several versions of Envision's user interface developed, and to being able to release a Macintosh and X/Motif client. Our next challenge is to let local (at Virginia Tech) users try out the Envision system, and then, based on their comments and reactions, to iteratively refine the user interface. We will also continue to expand the content of the database and further our work on document type definitions for the various types of objects. As these stabilize we expect ACM to begin distribution of parts of the Envision database.

After initial testing of Envision at Virginia Tech, we will release Envision to other sites, and eventually make it accessible over the Internet. In parallel with basic work to extend the Envision software and database, we will investigate how digital libraries such as Envision can be used to improve education. In particular, we will explore how the computer science curriculum can be revised and how learning can improve when a large digital library is available to all students and instructors.

Our investigations on Envision and investigations by others on similar systems in other domains promise to move us closer to an era of comprehensive, integrated, worldwide digital libraries.

Acknowledgments

This work has been funded by the National Science Foundation under Grant IRI-9116991, Dr. Maria Zemankova, Monitor. The project is also supported by the ACM and by Virginia Tech. The authors gratefully acknowledge the contributions of Robert K. France, Kevin A. Mayo, Eric Labow, and the Human–Computer Interaction Research Group at Virginia Tech, as well as our interviewees and usability evaluation participants.

References

Belkin, N. J., Borgman, C., Brooks, H., Bylander, T., Croft, W. B., Daniels, P., et al. (1987). Distributed expert-based information systems: An interdisciplinary approach. *Information Processing & Management, 23,* 395–409.

Betrabet, S. C., Fox, E. A., & Chen, Q. F. (1993). *A query language for information graphs.* (Technical Report TR-93-03). Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA.

Brueni, D. J. (1992). *Literate DTD design.* Unpublished technical report. Virginia Polytechnic Institute and State University, Blacksburg, VA.

Brueni, D. J., Cross, B. T., Fox, E. A., Heath, L. S., Hix, D., Nowell, L. T., & Wake, W. (1993). What if there were desktop access to the computer science literature? In S. C. Kwasny (Ed.), *Proceedings of 21st Annual ACM Computer Science Conference: CSC '93* (pp. 15–22). Indianapolis, IN. New York: ACM Press.

Chen, Q. F. (1992). *An object-oriented database system for efficient information retrieval applications.* Ph.D. thesis, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA.

Coombs, J. H., Renear, A. H., & DeRose, S. J. (1987). Markup systems and the future of scholarly text processing. *Communications of the Association for Computing Machinery, 30,* 933–947.

DeRose, S. J., Durand, D. G., Mylonas, E., & Renear, A. H. (1990). What is text, really? *Journal of Computing in Higher Education, 1,* 3–26.

Fox, E. A. (1987). Development of the CODER system: A testbed for artificial intelligence methods in information retrieval. *Information Processing & Management, 23,* 341–366.

Fox, E. A. (1988a). ACM Press database and electronic products—new services for the information age. *Communications of the Association for Computing Machinery, 31,* 948–951.

Fox, E. A., (1988b). Testing the applicability of intelligent methods for information retrieval. *Information Services and Use, 7,* 119–138.

Fox, E. A. (1990). How to proceed toward electronic archives and publishing. *Psychological Science, 1,* 355–358.

Fox, E. A. (1991). Advances in interactive digital multimedia systems. *IEEE Computer, 24,* 9–21. Reprinted in Baecker, R. (Ed.) (1993). *Readings in groupware and computer supported cooperative work* (pp. 342–354). San Mateo, CA: Morgan Kaufmann.

Fox, E. A. (1992). Building a user-centered database from the ACM literature. In *Proceedings of the Symposium on Document Analysis and Information Retrieval* (pp. 235–246). Las Vegas, NV.

Fox, E. A., Chen, Q. F., Daoud, A. M., & Heath, L. S. (1991a). Order preserving minimal perfect hash functions and information retrieval. *ACM Transactions on Information Systems, 9,* 281–308.

Fox, E. A., Chen, Q. F., & France, R. K. (1991b). Integrating search and retrieval with hypertext. In E. Berk & J. Devlin (Eds.), *Hypertext/Hypermedia handbook* (pp. 329–355). New York: McGraw-Hill.

Fox, E. A., Chen, Q. F., & Heath, L. S. (1992a). A faster algorithm for constructing minimal perfect hash functions. In *Proceedings of the 15th International Conference on Research and Development in Information Retrieval, SIGIR '92* (pp. 266–273). Copenhagen, Denmark.

Fox, E. A., & France, R. K. (1987). Architecture of an expert system for composite document analysis, representation and retrieval. *International Journal of Approximate Reasoning, 1,* 151–175.

Fox, E. A., France, R. K., Daoud, A. M., & Cline, B. E. (1993). Development of a modern OPAC: From REVTOLC to MARIAN. In *Proceedings of the 16th International Conference on R&D in Information Retrieval, SIGIR '93,* Pittsburgh, PA. Accepted for publication.

Fox, E. A., Heath, L. S., Chen, Q. F., & Daoud, A. M. (1992b). Practical minimal perfect hash functions for large databases. *Communications of the Association for Computing Machinery, 35,* 105–121.

Fox, E. A., & Lunin, L. (1993). Introduction and overview to Perspectives on digital libraries. *Journal of the American Society for Information Science, 44,* 441–445.

Glyssens, M., Paredaens, J., & Van Gucht, D. (1990). A graph-oriented object model for database end-user interfaces. In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data,* Atlantic City, NJ. *ACM SIGMOD Record, 19,* 24–33.

Goldfarb, C. F. (1991). *The SGML handbook.* Oxford: Oxford University Press.

Hix, D., & Hartson, H. R. (1993). *Developing user interfaces: Ensuring usability through product and process.* New York: Wiley.

Kahle, B., Morris, H., Goldman, J., Erickson, T., & Curran, J. (1993). Interfaces for distributed systems of information servers. *Journal of the American Society for Information Science, 44,* 453–467.

Lavinus, J. W. (1992). *Heuristics for laying out information graphs.* Master's thesis, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA.

Lesk, M. (1991). The CORE electronic chemistry library. In *Proceedings of the 14th Annual International ACM/SIGIR Conference on R&D in Information Retrieval* (pp. 92–112), Chicago, IL.

Newcomb, S. R., Kipp, N. A., & Newcomb, V. T. (1991). The "HyTime" Hypermedia/Time-based document structuring language. *Communications of the Association for Computing Machinery, 34,* 67–81.

Nowell, L., & Hix, D. (1992). User interface design for Project Envision database of computer science literature. In *Proceedings of the 22nd Virginia Computer User's Conference,* Blacksburg, VA.

Nowell, L. T., & Hix, D. (1993a). Query composition: Why does it have to be so hard? In *Proceedings of the East-West International Conference on Human–Computer Interaction.*

Nowell, L. T., & Hix, D. (1993b). Visualizing search results: User interface development for the Project Envision database of computer science literature. In *Proceedings of HCI International '93, 5th International Conference on Human–Computer Interaction* jointly with *9th Symposium on Human Interface (Japan),* Orlando, FL.

Paredaens, J., et al. (1992). An overview of GOOD. *ACM SIGMOD Record, 21,* 25–31.

Wartik, S., Fox, E. A., Chen, Q. F., & Heath, L. S. (1992). Hashing algorithms. In W. Frakes & R. Baeza-Yates (Eds.), *Information retrieval: Data structures and algorithms* (pp. 293–362). Englewood Cliffs, NJ: Prentice-Hall.

Weaver, M., France, R., Chen, Q., & Fox, E. (1989). Using a frame-based language for information retrieval. *International Journal of Intelligent Systems, 4,* 223–257.

Wolfram, S. (1991). *Mathematica: A system for doing mathematics by computer* (2nd ed.). Reading, MA: Addison-Wesley.

Yankelovich, N. (1990). Three pieces of the puzzle: Wide-area hypermedia, information agents and on-line reference works. *Psychological Science, 1,* 350–352.