What if There Were Desktop Access to the Computer Science Literature?

Dennis J. Brueni, Edward A. Fox, Lenwood S. Heath,
Deborah Hix, Lucy T. Nowell, and William C. Wake

TR 92-42

Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

August 12, 1992

What If There Were Desktop Access to the Computer Science Literature?*†

Dennis J. Brueni, Edward A. Fox, Lenwood S. Heath, Deborah Hix, Lucy T. Nowell, William C. Wake ‡

Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061-0106

Abstract

What if there was an electronic computer science library? Consider the possibilities of having your favorite publications available within finger's reach. Consider project Envision, an ongoing effort to build a user-centered database from the Computer Science literature. This paper describes our first year progress, stressing the motivation underlying project Envision, user-centered development, and overall design.

CR Categories and Subject Descriptors: H.3.1 **[Information Storage and Retrieval]:** Content Analysis and Indexing; H.5.2 **Information Interfaces and Presentation]:** User Interfaces — *Evaluation / methodology, Screen design*; I.7.2 **[Text Processing]:** Document Preparation — *Standards*

Keywords: electronic libraries, SGML, HyTime, database, object-oriented, user interface, user-centered

1 AN ELECTRONIC C.S. LIBRARY

Many believe the effectiveness of education and the efficiency of research in the future will rely on comprehensive electronic libraries [6, 12]. What *if* you had a complete library of the computer science literature containing journal articles, books, technical reports, videos, and audio clips? Imagine pinpointing the documents that best match your current interests, using an intuitive and efficient user interface. What *if* you could browse the library by following links relating its constituent documents, thus navigating along many dimensions? Imagine the literature in this library coming alive—an algorithm presented in a paper executed and animated without the drudgery of your implementing it (again). Imagine seeing the video of the keynote speaker at that conference you missed. What *if* this library was available to you?

*This work has been funded by the National Science Foundation under Grant IRI-9116991.

†Reprint of paper submitted to 1993 ACM Computer Science Conference.

‡Authors' present electronic mail addresses: brueni@csgrad.cs.vt.edu, fox@vtopus.cs.vt.edu, heath@vtopus.cs.vt.edu, hix@vtopus.cs.vt.edu, nowell@csgrad.cs.vt.edu, and wakew@csgrad.cs.vt.edu.

Project Envision, a research effort at Virginia Tech, is aimed at prompting computer scientists and practitioners to *envision* this new type of reality, to give their ideas and needs, and to help with our user-centered development and testing efforts. We hope to stay focused on the future, on the users, and on practical tasks and needs.

An electronic library is a database of literature, along with enhanced capabilities for accessing that literature, carrying forward all of the advantages of modern libraries and adding new ones, facilitated by new technology and related advances. For the initial version of Envision, the ACM has agreed to allow any of its publications to be loaded into the database. We are also securing access to other literature for Envision.

The above scenarios can only be realized if such a large collection of computing literature is supported by a rich representation of the contents of that literature. Hypertext links connect interesting document features to related points of interest in the database. Documents will be stored in a manner intended to maximize and simplify reuse of their contents. HyTime and SGML provide a standard framework for the descriptive markup required to achieve such goals [17, 18]. The content elements of articles may be thought of as nodes in a large information graph, connected by structural, referential, and hypertext links. Efficient storage and retrieval from such a semantic network will be enabled through capabilities of the "graph of objects" database LEND [7].

The capability to access the database builds on our previous experience with an advanced information retrieval system (CODER) based on artificial intelligence and expert system techniques [10, 14, 15]. CODER makes use of LEND for storage of data, information, and knowledge bases, including multimedia and hypermedia.

Once a user identifies one or more documents as being relevant, he or she may view the document or may move to related documents via links in the database. In Envision, viewing a document is more than just reading a display; it is an interactive experience. The user may call on *Mathematica*[1] to investigate the mathematical content of the document, or an algorithm animator to explore the algorithmic content. The ability to explore the literature rather than just read it will serve as the basis for future laboratories to enhance education.

This paper is organized as follows. Section 2 describes the principle of user-centered design and how it is employed in Envision. Section 3 lists some of the materials planned to be made available in the Envision database. Section 4 motivates the design of Envision by discussing how users want to use computer science literature. An overview of this design is given in Sections 5 and 6 with special attention paid to the user interface in Section 7.

2 USER-CENTERED DEVELOPMENT

User-centered development is a process consisting of a number of steps. In Project Envision, the first step was to prepare an interview questionnaire for a sample of typical and not-so-typical users of computer science literature. The questionnaire surveys the users' current use of information sources, their future information needs, and their wish lists for the electronic library of the future. The second step was to analyze the responses of the users to guide the identification of Envision's capabilities, particularly of its user interface. The third step was the design and implementation of a prototype to support usability testing with representative users of Envision. The fourth step is iterative design; returning to step two and re-evaluating

[1] *Mathematica* is a trademark of Wolfram Research Inc.

the suggestions made by users during evaluation sessions. In short, users are involved from inception to installation of Envision.

Over a four month period, we interviewed twelve professionals in the areas of computer science and information retrieval. The subjects were carefully chosen to broadly represent the type of user we expect for Envision. Five interview subjects are active industry researchers in the areas of information retrieval and human-computer interaction, including a corporate librarian. One is a corporate research department head, responsible for document retrieval supporting the work of others. One is a faculty member specializing in information storage and retrieval at another research university. Five VPI&SU faculty members, three from the Department of Computer Science and two from Industrial and Systems Engineering, were also interviewed. In one to two hour interviews, interviewed subjects 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.

User task analysis [23] followed our interviews and was based partly on interview results. Our first design work is focusing on the major tasks of targeted search and retrieval, and use of the search results. Refinement of task analyses is proceeding iteratively with user interface design, prototyping, and usability testing.

3 CONTENTS OF THE DATABASE

Interview subjects indicated a reliance on journals and conference proceedings as major sources of information, with additional attention to conference attendance. They ranked correspondence with colleagues as equal in importance to journals as a source. Textbooks and trade magazines also ranked high as information sources. Some use network bulletin board services or news/discussion groups. Only one uses a CD-ROM system. Interestingly, none mentioned using video or audio recordings.

Guided by these priorities, the data to be used for Envision will include all data accessible from ACM that can be processed and prepared for loading and use. A number of resources will be provided, including a large quantity of data of various types.

algorithms derived from *Collected Algorithms* (CALGO);
bibliographic entries including *Guide to the Computing Literature*;
full text of *Communications of the ACM* (CACM) as included in the *Computer Library* CD-ROM, and the full set of texts from Design Automation conferences and newsletters that are included in the *DA Library* CD-ROMs;
dictionary and thesaurus which is the Oxford English Dictionary;
hypertexts included in the *Hypertext Compendium* covering the literature regarding hypertext and hypermedia;
journals including *ACM Transactions on Mathematical Software* (TOMS) that will be linked with CALGO, and *ACM Transactions on (Office) Information Systems* (TOIS);

multimedia including the SIGGRAPH and SIGCHI video reviews, the *Interactive Digital Video* video documentary, and other image, video, and audio resources;
raw data from published studies, when available;
page images including coverage of the Design Automation area as included in the *DA Library* CD-ROMs and others to be scanned as needed for complete coverage of important topics;
reviews including those published in *Computing Reviews* and made available on the *Computing Archive* CD-ROM;
technical reports such as from VPI&SU;
taxononomy which is the Computing Reviews Classification System and supports mapping between the several versions approved over the years;
timeline information on ACM conferences and related events.

The above data will be supplemented by data collected at VPI&SU in connection with a variety of research projects, including Virginia Disc One [11]. These contain rich resources in the area of information retrieval, as well as images and digital video. Other data of particular value will also be included as they become available. The data will be valuable to computer scientists; we plan realistic user studies to explore how well the tasks of practicing scientists can be supported by electronic databases. In addition, the data will be current—updated at least quarterly.

4 USES OF INFORMATION

Special attention is being paid to how people want to use the information available in the computer science literature. For the purposes of authoring, interview subjects cite a need to reuse document features, such as selecting citations for quoting, or obtaining bibliographic information. Subjects wanted an electronic metaphor for "yellow sticky notes" to annotate documents browsed in the database. Other potential users expressed an interest in gathering information into personal subsets of the database, or studying the structure of webs contained in the database. Interview subjects want to see various visualizations of data in order to understand the information therein.

The object-oriented design philosophy underlying Envision allows us to consider new ways to access and use information. Imagine querying the database to find reports on programming languages with examples of code for solving the dining philosopher's problem. Imagine viewing a graphical time-line of all documents related to your favorite research topic over the years. These and many other things are possible within the paradigm underlying Envision.

4.1 Education

One goal of Project Envision is to investigate the role a database can play in education. Instructors could query the database to obtain accurate and current materials for the classes they teach, as well as assistance in creating oral presentations. Students and researchers would naturally value the database as an aid in locating relevant information. Visualizations such as algorithm animations appear promising for accelerating the process of understanding algorithms [2]. In the yearly graduate course on Information Storage and Retrieval, the theme will be on how we could develop a national electronic library, with Project Envision as the case study. In addition, students will work with data, technology, tools, and interfaces

related to Envision, including accessing online literature from conferences of the ACM Special Interest Group on Information Retrieval. Visualization of IR algorithms, and testing of them with real data collections, will further assist in giving students a very thorough and fully grounded view of the the subject matter. Another early use of Envision in education will be in the algorithms classes at Virginia Tech. Initially, some number of computational geometry algorithms will be animated using X-TANGO [24, 25]. The journal papers from which these algorithms derive will be loaded into the Envision data base, along with the animations. Students in the algorithms classes will have online access to the original description of each algorithm along with a executable version in a linked environment. Students will be able to explore, in patterns that suit them, the knowledge web in this subarea of computer science in a way that will enhance their ability to make important connections.

4.2 Research and Visualization

In interviews, subjects expressed the need for mathematical capabilities such as statistical analysis and graphing. When working with algorithms our subjects often want to obtain source code in a language with which they can work. We also found strong interest in animations of algorithms. Other users were interested in direct manipulation of objects found in the database, such as rotating a three-dimensional cube or tracing the steps of a proof.

Heeding the advice of users, and as a proof of concept for the more general task of interfacing to existing tools, we are constructing a link called "MathAccess" between Envision and *Mathematica* [5]. MathAccess will allow Envision to use and control the mathematical, graphical, algorithmic, and symbolic manipulation capabilities of *Mathematica*:

- extracting a mathematical expression to evaluate or graph;
- implementing numerical and combinatorial algorithms;
- demonstrating the operation of certain algorithms;
- "filling in" details of algebraic manipulations; and
- generating the steps of a proof in symbolic logic.

The intent is that the mathematical and algorithmic content of papers may be more easily understood if the reader can work through the details.

Animations, timelines, flowcharts, maps, graphs, simulations, and browsing are all alternate visualizations of objects. The use of well-defined descriptive markup, i.e., SGML, to describe all document features makes it possible for these alternate document views to take place in an extensible fashion. Imagine authors writing custom visualization tools for database objects, and submitting them for publication in the database—another result of the richness of the object-oriented paradigm underlying Envision.

5 DATABASE DESIGN

In practice, one distinguishes two major levels of "objects" when considering technical literature. The dichotomy can be compared to an open book versus a closed book. At the inner (open book) level, one expects to see flavors of objects like mathematics, proofs, or figures. These are objects that authors normally work with to create documents and readers struggle

to understand. These objects, *document features*, are discussed in Section 5.1. On the outer (closed book) level we see the objects that are more familiar to librarians and researchers. Examples of outer level objects are titles, journals, books, authors, dates, and articles. A description of the object analysis for these objects is presented in section 5.2.

5.1 Document Features

As product features attract a person to purchase a particular computer or car, document features attract a person to read a document and are essential for effective scientific communication. Example types of document features include algorithms, program code, time complexities, lemmas, proofs, theorems, definitions, examples, dialogues, formulae, equations, animations, tables, graphics, 3-D objects, bibliographic entries, and citations [4]. In addition to the document features themselves, we are identifying semantic relationships between features such as "is the time complexity of", "is the author of", or "solves problem."

The results of interviews (see Section 2) provided insight into the usefulness of document features. In addition to traditional strategies, a user of an Envision database may want to use document features to form queries and navigate through the database of literature. For example, a user might be interested in seeing all algorithms in CALGO with a time complexity of $O(n \log n)$. Interview subjects expressed interest in reuse of document features. For example, a user may be interested in implementing an algorithm appearing in a journal article. There is no reason to start from scratch if the text of the algorithm is already presented in the paper. A simple text filter might convert it from the high-level algorithmic notation to a reasonably complete piece of code.

Our representation for document features attempts to maximize the document user's ability to access and reuse the features. To this end, literature in the Envision database is represented in SGML (Standard Generalized Markup Language) [18] and HyTime (an international multimedia standard based on SGML) [17, 22]. These standard formats, where possible, permit us to represent documents in the Envision database as ordered hierarchies of content objects capturing much of the semantic nature of the writing [9]. Furthermore, this permits applications that interface to Envision to transform received data into a usable internal format. A proof of concept is the *Mathematica* interface to Envision, which will convert SGML representations of mathematical formulae to the *Mathematica* representation, allowing the user to directly use it with *Mathematica*. Another example could be converting bibliographic entries into BibTeX format for use with the LaTeX document preparation system [20]. Well-defined features also enable the authoring of "meta-objects" that operate on document features and are stored in the database itself (such as a program producing a finite automaton corresponding to a given regular expression).

Literate programming [19] is being used to facilitate the design of SGML document type definitions (DTDs) for document types and features. The use of literate programming has allowed us to write documents describing the design, semantics, and intended use of DTDs in parallel with the actual composition of the DTDs. The high level literate programming notation (itself defined by an SGML DTD) supports modular design, enforces DTD design guidelines, and permits translation to other notations [3].

5.2 Object-Oriented Analysis

The database must support querying, browsing, import, and export. We plan to store information about people, projects, timelines, and organizations, as well as articles and bibliographic information. We are using the Coad-Yourdon object-oriented analysis notation for object specification [8]. The analysis is divided in five separate layers:

1. *Subjects*: identifies groups of related classes. Our analysis includes three high level subjects: Date, Legal Person, and Publication.
2. *Classes*: shows the inheritance structure and all identified objects grouped by subject. Our classes include Person, Organization, Conference, and Journal.
3. *Attributes*: describes the values and types of various objects. For example: a Person's name and electronic mail address.
4. *Structure*: shows relationships between objects such as containment and inheritance. For example: a Journal contains Volumes, a Corporation is a type of Legal Person.
5. *Services*: addresses the actions that objects are required to perform and the services they are required to provide.

6 ARCHITECTURE

The feature that interview subjects most wanted in an information retrieval system is access from their offices and workstations. Responding to this concern, we designed the Envision user interface to run as a client process on the user's workstation, communicating with an Envision server via a network, as illustrated in Figure 1.

The protocol between the user interface and the database system will use a modification of the standard Z39.50 protocol. This approach is similar to that used by WAIS (Wide Area Information Server) and is compliant with ISO standards.

6.1 LEND

LEND (Large External-storage object-oriented Network Database) is designed to solve some of the problems faced by researchers in the fields of artificial intelligence, computational linguistics, hypermedia, and information retrieval, as well as problems of users of such systems [7]. LEND supports various classes of objects, as is normally the case for object-oriented databases, and provides persistent storage and rapid access. New algorithms for minimal perfect hashing and order preserving minimal perfect hashing were specially developed and give near optimal performance in terms of time and space for access to objects [13, 16]. LEND offers special classes for nodes, links, and anchors to support hypertext and hypermedia collections. LEND also supports path-based inference providing many of the benefits of semantic networks. New graph layout heuristics provide efficient storage and access to the graph structured database [21].

A graph based query language provides a powerful means of retrieving data from LEND [7]. This language assumes that the database is stored as an information graph G with nodes and arcs; the result of a query is a subgraph of G. Query language statements can be used to ask about both the content of G and aspects of the structure of G. For example, if a node represents a word and an arc represents a synonym relation between the two words it connects, then it

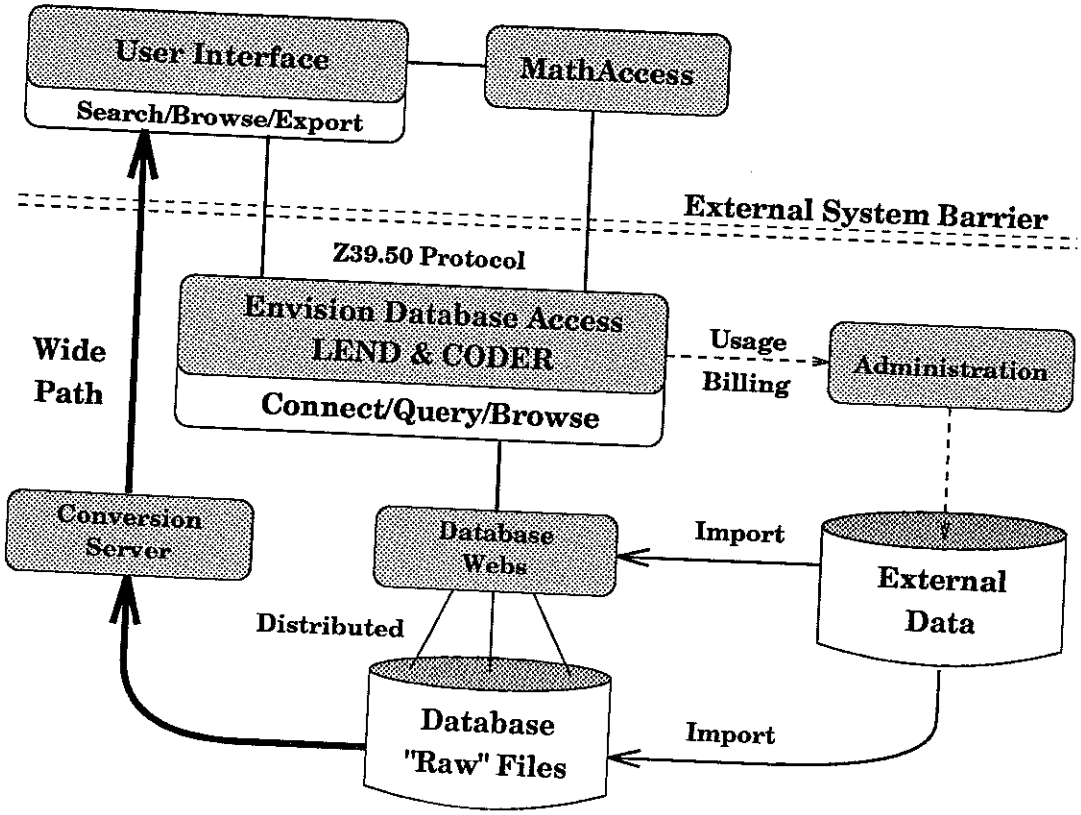

Figure 1: The Envision architecture

would be possible to find all the synonyms of a particular word that can be reached by following synonym arcs. We are working on optimizing execution of query language statements [1].

7 USER INTERFACE DESIGN

The first version of the user interface design for display and use of search results is complete, and a rapid prototype has been developed. An example of our search results screen is presented in Figure 2. The interface design provides flexible use of monitors, with configurations varying both in size and number of displays. The most basic configuration uses a thirteen inch monochrome display. Central to the design is the concept of viewing each document as a node (in Figure 2, the bubbles) within the Envision database graph and representing the node graphically as an icon. The user sees results as clusters of documents, similar to a scatter plot. The design provides a graphical, direct manipulation of search results and provides users with control over the semantics of six dimensions associated with the results display: icon placement along the x-axis and y-axis; the number associated with each icon; and size, color and shape of the icon itself. The icon number associates the icon with textual information about the document, which is displayed in the Item Summary window when the icon is selected. The user may view document content in the item preview window.

We prepared usability specifications and benchmark tasks related to use of the search results interface, and have completed the first cycle of iterative evaluation and refinement of the design. In our first cycle of usability evaluation, our task performance goals were

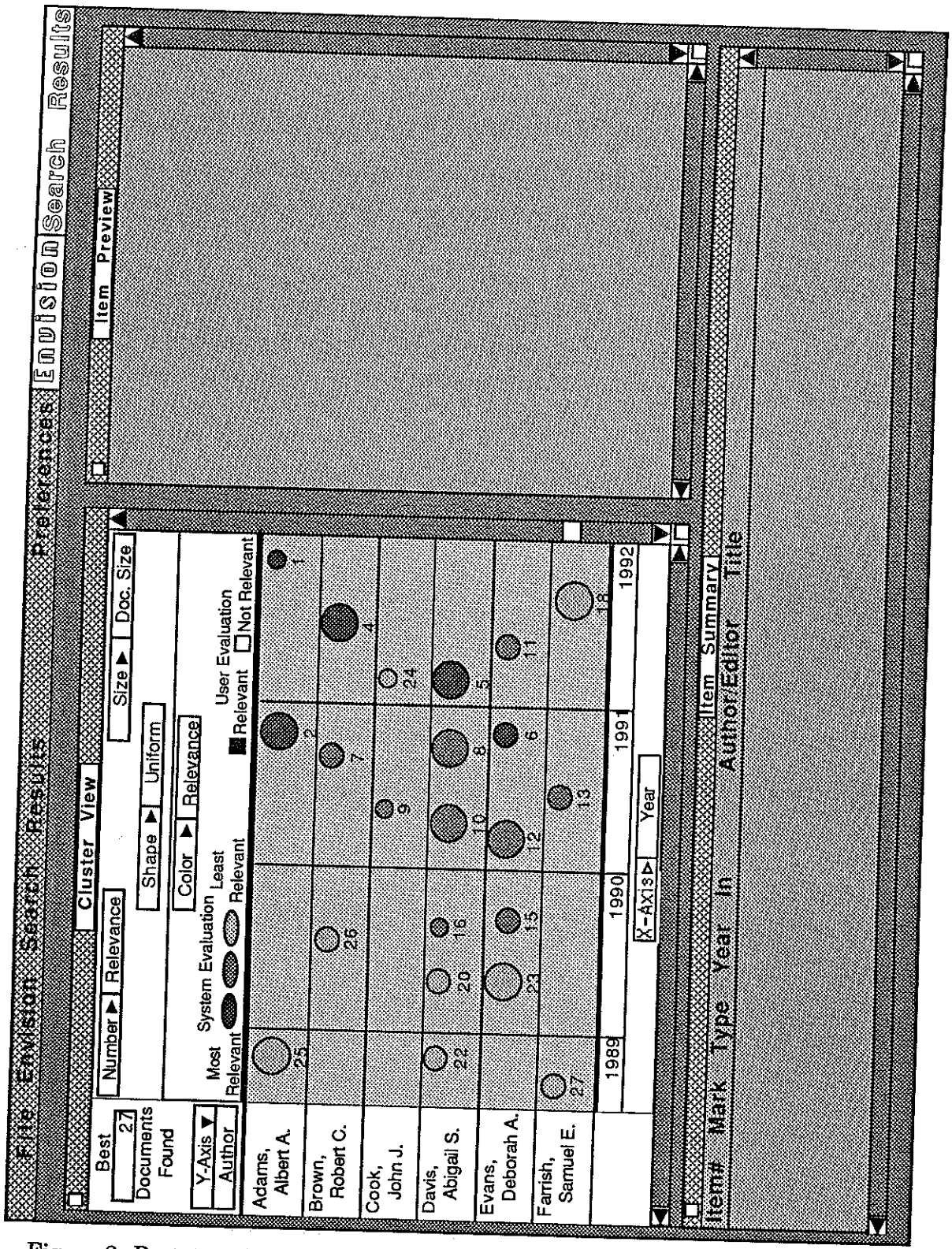

Figure 2: Prototype Envision user interface for display and use of search results

almost completely met. Users reaction to the "bubble design" were very positive. In less than 10 minutes from the time they first saw the screen, even seasoned DOS users were able to understand and manipulate the bubbles and associated documents. All users said the layout is intuitive and visually represents a large amount of information. Revisions to the design will include additional features, but no changes to the design concept. Design is continuing on search query screens, zooming in and out on results, and viewing of individual items.

8 CONCLUSION

Project Envision involves development of an archive for computing, using literature provided by ACM, and other sources. Through interviews and study of existing publications, we are identifying the document features that should be managed to allow users to more efficiently carry out the tasks they desire, including those where information retrieval is an embedded application. We hope to develop ways so that authors can assist in preparing documents that can be easily converted into SGML form, with descriptive markup that will facilitate automatic handling of key objects. Conversion efforts will allow us to develop an archive of sufficient size to enable testing by faculty, students, and staff. Our distributed expert-based information system, CODER, and the graph-of-objects database engine, LEND, should enable flexible access to relevant data.

As efforts proceed, and we gain experience with the problems and successes of electronic archives, larger efforts will no doubt be launched. Of particular interest, and of urgent importance for our nation's future progress, is construction of a national electronic archive. What *if* that could be in place by the turn of the century?

References

[1] BETRABET, S. C. Query Optimization for a Graph Based Query Language. Master's thesis, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1992. In progress.

[2] BROWN, M. H. *Algorithm Animation*. The MIT Press, Cambridge, MA, 1988.

[3] BRUENI, D. J. Literate DTD Design, August 1992. Technical Memorandum.

[4] BRUENI, D. J., FOX, E. A., AND HEATH, L. S. Features of Computer Science Liturature. Tech. rep., Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1992. In preparation.

[5] BRUENI, D. J., HEATH, L. S., AND PARIPATI, P. MathAccess: The Envision Link to Mathematica. Tech. rep., Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1992. In preparation.

[6] BUSH, V. As we may think. *Atlantic Monthly 176* (July 1945), 101–108.

[7] CHEN, Q. F. *An Object-Oriented Database System for Efficient Information Retrieval Applications*. PhD thesis, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, March 1992.

[8] COAD, P., AND YOURDON, E. *Object-Oriented Analysis.* Prentice Hall, Englewood Cliffs, NJ 07632, 1990.

[9] DEROSE, S. J., DURAND, D. G., MYLONAS, E., AND RENEAR, A. H. What is Text, Really? *Journal of Computing in Higher Education 1* (1990), 3–26.

[10] FOX, E. A. Development of the CODER system: A testbed for artificial intelligence methods in information retrieval. *Information Processing & Management 23*, 4 (1987), 341–366.

[11] FOX, E. A., Ed. *Virginia Disc One.* VPI&SU Press. Produced by Nimbus Information Systems, Blacksburg, VA, 1990. ISBN 0-929900-00-6. Available from Dr. Edward A. Fox, Department of Computer Science, VPI&SU, Blacksburg, VA 24061-0106.

[12] FOX, E. A. Building a user-centered database from the ACM literature. In *Proc. Symposium on Document Analysis and Information Retrieval* (Tropicana Hotel, Las Vegas, Nevada, March 16-18 1992), pp. 235–246.

[13] FOX, E. A., CHEN, Q. F., AND HEATH, L. S. A Faster Algorithm for Constructing Minimal Perfect Hash Functions. In *Proceedings of the 15th International Conference on Research and Development in Information Retrieval* (1992), pp. 266–273.

[14] FOX, E. A., FRANCE, R., KOUSHIK, M., MENEZES, J.-L., CHEN, Q. F., DAOUD, A., AND NUTTER, J. CODER: A retrieval and hypertext system using SGML and a lexicon. In *Proc. ACH/ALLC '91* (Arizona State Univ., Tempe, AZ, March 1991), pp. 159–163.

[15] FOX, E. A., AND FRANCE, R. K. Architecture of an expert system for composite document analysis, representation and retrieval. *International Journal of Approximate Reasoning 1*, 1 (1987), 151–175.

[16] FOX, E. A., HEATH, L. S., CHEN, Q. F., AND DAOUD, A. M. Practical minimal perfect hash functions for large databases. *Communications of the Association for Computing Machinery 35*, 1 (January 1992), 105–121.

[17] GOLDFARB, C. F., AND NEWCOMB, S. R. *ISO/IEC DIS 10744:1991. Information Technology — Hypermedia/Time-based Structuring Language (HyTime).* ISO/IEC, October 1991.

[18] ISO. *Information Processing – Text and Office Systems – Standard Generalized Markup Language (SGML), First Edition.* ISO, October 1986. ISO 8879-1986.

[19] KNUTH, D. E. Literate Programming. *The Computer Journal 27* (1984), 97–11.

[20] LAMPORT, L. LaTeX *User's Guide and Reference Manual.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

[21] LAVINUS, J. W. Heuristics for Laying Out Information Graphs. Master's thesis, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, August 1992.

[22] NEWCOMB, S. R., KIPP, N. A., AND NEWCOMB, V. T. The "HyTime" Hypermedia/Time-based Document Structuring Language. *Communications of the Association for Computing Machinery 34*, 11 (November 1991), 67–81.

[23] NOWELL, L. T. Graphical User Action Notation: A Technique to Support Task Analysis in User Interface Design. Master's thesis, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1992. In progress.

[24] STASKO, J. T. TANGO, A framework and system for algorithm animation. *Computer 23*, 9 (September 1990), 23–39.

[25] STASKO, J. T. Animating algorithms with XTANGO. *SIGACT News 23*, 2 (Spring 1992), 67–71.