A VECTOR-PRODUCT INFORMATION RETRIEVAL SYSTEM ADAPTED TO HETEROGENEOUS, DISTRIBUTED COMPUTING ENVIRONMENTS

Mark E. Rorvig
Software Technology Branch
NASA Johnson Space Center
Houston, TX 77058

ABSTRACT

Vector-product information retrieval (IR) systems produce retrieval results superior to all other searching methods but presently have no commercial implementations beyond the personal computer environment. (NELS) NASA Electronic Library System, provides a ranked list of the most likely relevant objects in collections in response to a natural language query. Additionally, the system is constructed using standards and tools (i.e., UNIX, X-Windows, Motif, TCP/IP) that permit its operation in organizations that possess many different hosts, workstations and platforms. There are no known commercial equivalents to this product at this time. The product has applications in all corporate management environments, particularly those that are information intensive, such as finance, manufacturing, biotechnology, and research and development.

INTRODUCTION

The field of information retrieval (IR) has always advanced unevenly. Even fundamental theoretical insights have occaisonally required a generation or more of developments in electrical engineering to enjoy commercial practice. The NASA Electronic Library System (NELS) is a good example of hardware and software dependent intellectual advances. In this case, two discoveries from the 1960's regarding indexing system performance and retrieval ranking, have been implemented within the context of standard operating systems, network protocols, languages, and display tools. Figure 1 below illustrates the interactions among these electrical and software standards and components.

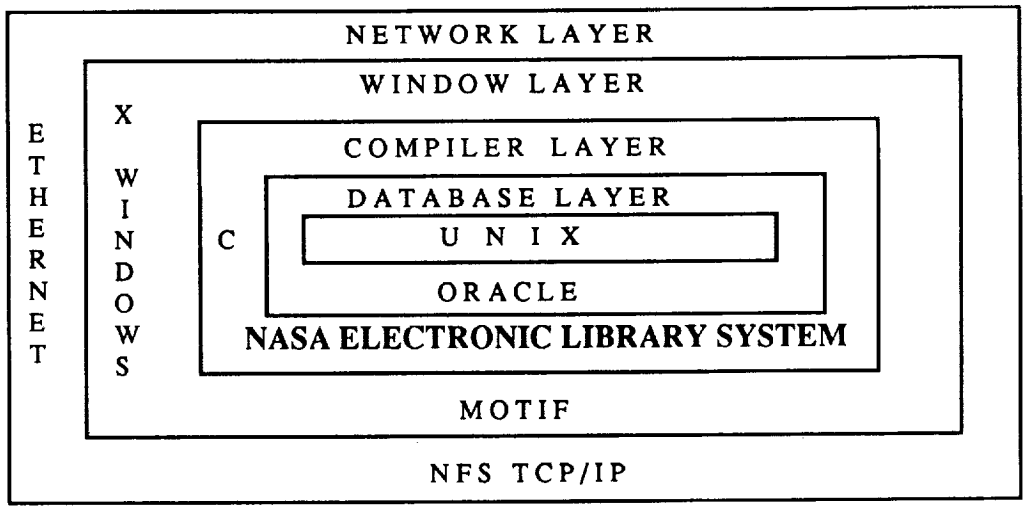

Figure 1: NELS environment variables expressed as layers of interacting electrical and software standards, and their specific implementations. For example, NELS is written in "C", displayed through Motif, shares resources with other devices by NFS TCP/IP, and depends on Oracle and UNIX for system support.

This article will describe the intellectual history of searching supported by NELS 1.0 and the its method of use on wide area networks. A few basic screen layouts will be examined to provide a "feel" of the system for the reader. FIinally, some hardware platforms and configurations currently supported by NELS will be discussed.

VECTOR-PRODUCT SEARCHING AND RELEVANCE FEEDBACK

One of the great counter intuitive discoveries of IR research in the early 1960's was that the precision and depth of indexing did not result in improved retrieval performance when compared to the simple strategy of providing access to objects by every word in the title [1]. This research finding directly influenced the development of the NASA RECON system in the late 1960's, and subsequently the system development of Knight-Ridder's DIALOG, a large commercial distributor of database information. Although the available entry points were expanded to include all words in the abstract of an object, as well as the title and author information, subject indexing diminished in importance in online systems, except as an aid for quicly isolating a specific body of intellectually similar objects for further searching.

This concept has been implemented in NELS in two mutually reinforcing ways. First, although keywords may be assigned by indexers and searched as part of a natural language query, they are not an integral consideration. Rather, objects are assigned to classes, which may consist of intellectual concepts, organizational units, or in NASA's research engineering environment, system, subsystem, and sub-assembly hierarchies. A system user may then navigate to any class body of interest, search further, or simply list the objects. Second, the lexical components of object descriptions in the author, keyword, title, and abstract fields are all parsed, stripped of suffixes, and placed in an object attribute table for searching by natural language.

Searches conducted directly by natural language have long been a cornerstone of IR. However, in nearly all commercially available systems (excluding those dependent on devices such as array processors and their like), searching has been restricted to boolean logic queries; a form difficult to learn, subject to return of null sets in long queries, and in shorter ones, return of objects as an undifferentiated lump of knowledge. NELS has rejected this approach by implementing another important discovery of the 1960's, vector-product searching as defined at Harvard and Cornell by Gerard Salton and his various students and proteges [2].

Salton's method, expressed as the cosine vector approach, was a revolutionary discovery. Implicit within it and years ahead of the field was the first natural language interface. The formula below defines this method, simply stated as the cosine coefficient of commonality between QUERY terms and DOCument terms, or, between a query and all documents sharing at least one term in common with the query.

$$\text{COSINE}(\text{QUERY}i,\text{DOC}j) = \frac{\sum_{k=1}^{t} (\text{TERM}ik \bullet \text{TERM}jk)}{\sqrt{\sum_{k=1}^{t} (\text{TERM}ik)^2 \bullet \sum_{k=1}^{t} (\text{TERM}jk)^2}}$$

The result of applying this formula to a query is a list of objects retrieved by their degree of lexical closeness to the query. Further, the degree of closeness desired may be set by a user as a cut-off point associated with the user's sign-on identification. To initiate a search, from anywhere in a class structure, even at the very top, requires of the user only to type a description of his or her information requirement, in as general or specific a form as desired.

Wide implementation of this approach never occurred, though the focus of research in IR itself has indicated conclusively that it is superior to boolean search techniques, regardless of users, data, or system platforms [3]. Though there are many reasons for this phenomenon, chief among them was the basic limitation of memory required to store and search the vector spaces in a timely manner for massive data files. Since the recent arrival of virtual memory and RISC architectures, however, this consideration has diminished significantly in importance and has led NASA to implement this concept in complete and final form.

However, even with relevance information appearing as a cosine "score" beside each item, users may have further difficulty identifying some items and redesigning their query based upon it due to typing errors, misunderstanding of the vocabulary of an abstract of a retrieved document, or inability to learn enough about a document to conveniently restate their queries. Therefore, a system of relevance feedback is further available for users. To use this feature, a searcher pulls down a menu labelled "options" and selects the command "like". A simple click on one of the previously

retrieved items with a mouse causes the system to initiate another search, in this case substituting the abstract of the selected item for the earlier natural language query.

By iteratively applying this method, the user is able to bring his query closer and closer into conformance with the language of the system, constantly raising the level of relevance. Indeed, it is possible for a user to find objects with this method with extremely little prior knowledge of the search domain. Figure 2 below illustrates this procedure with a simple flowchart.

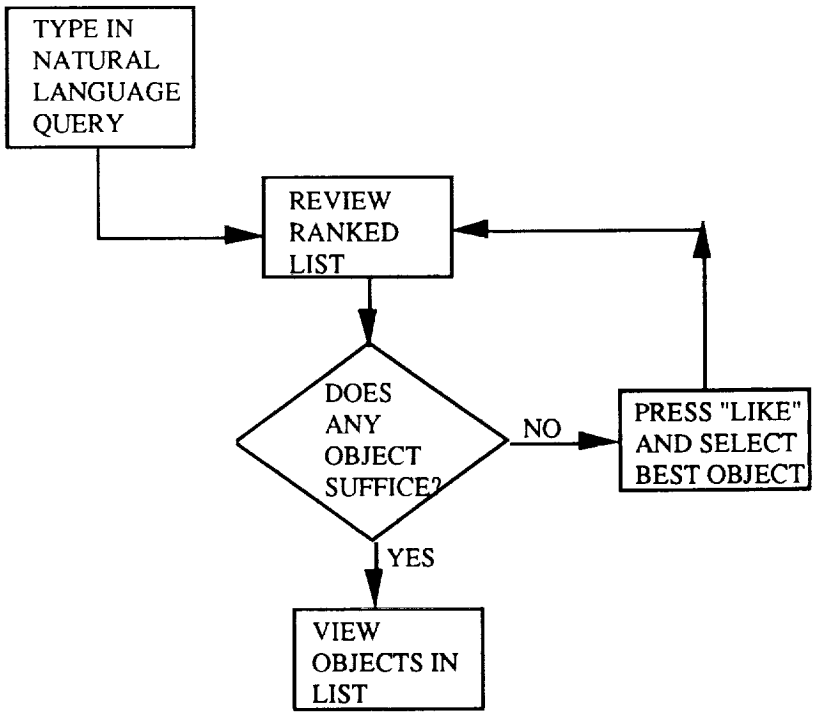

Figure 2: Method for using relevance feedback in NELS to improve system retrieval performance, that is, to move iteratively more closely to fit a natural language query to the language of the system.

NELS IMPLEMENTATION ON A WIDE AREA NETWORK

As noted in Figure 1 above, NELS is built to operate over ethernet networks, running Network File Server (NFS) and Transaction Communication Protocol Internet Protocol (TCP/IP). This permits operation of NELS to be distributed on a local, a regional, or a national basis. Moreover, although NELS itself must run on a UNIX based operating system platform, files may be referenced and accessed anywhere that an ethernet connection is maintained. For example, it would be possible to have a system hosted on an IBM RISC 6000 that loaded an Interleaf viewer from a SUN Sparc and a document file from a DEC VAX. Both tools and files may be distributed on heterogeneous devices and drawn to a common platform for display. In this manner, data from many systems may be used from a single NELS host, with response times and access slots limited only by the power of the host device.

All data on all referenced systems is recorded in NELS through a "metadata" record. A metadata record is information about a file of documents, images, graphics, or drawings. The specific fields of the metadata may be customized by a librarian for specific collections. One of the fields in all metadata records is the "path" field which describes for NELS the directory locations of files on all other devices at all other locations. A typical metadata record for an image of the earth from space appears below as Figure 3.

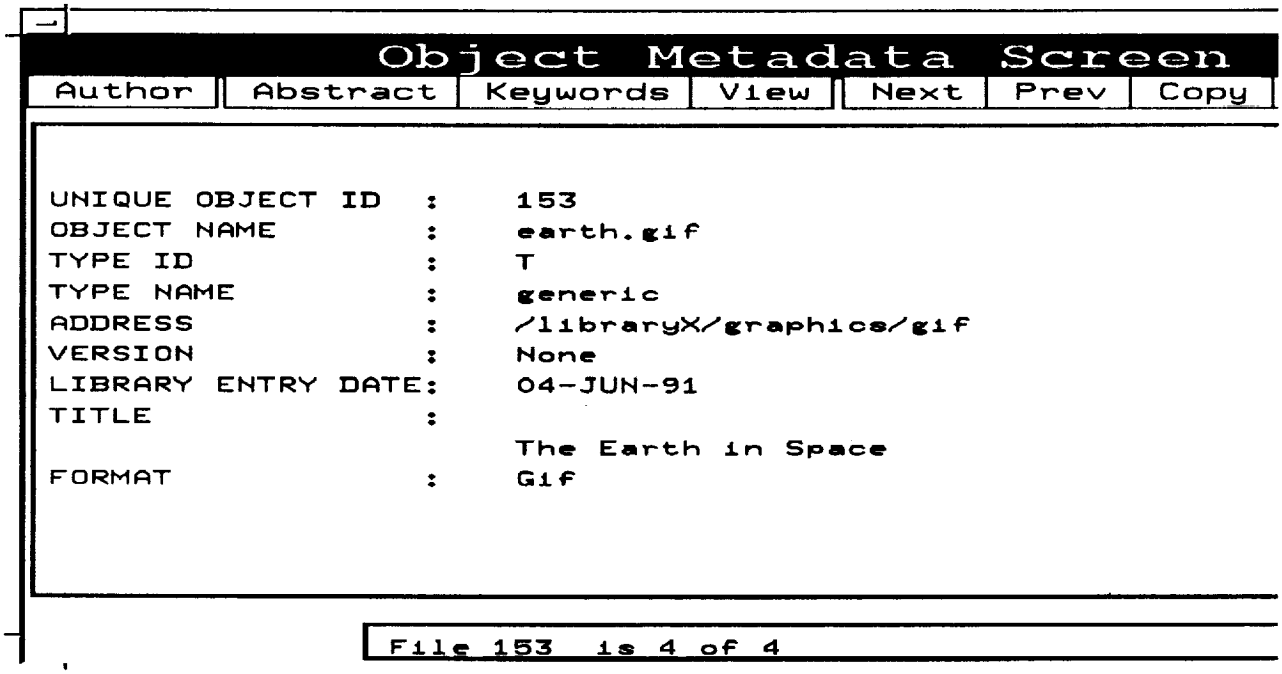

Figure 3: Metadata screen for a NELS entry.

Additonally, a number of devices hosting NELS may work in concert to optimize network topographies. For example, one network node may specialize in large numeric data files, another in image analyses, and another in engineering drawings. Although each node would be the primary user of its own data, however, by maintaining a complete metadata set at all three nodes, all data of all types could be accessed from any node, although more slowly at more remote locations. The diagram of Figure 4 below illustrates the distributed NELS concept.

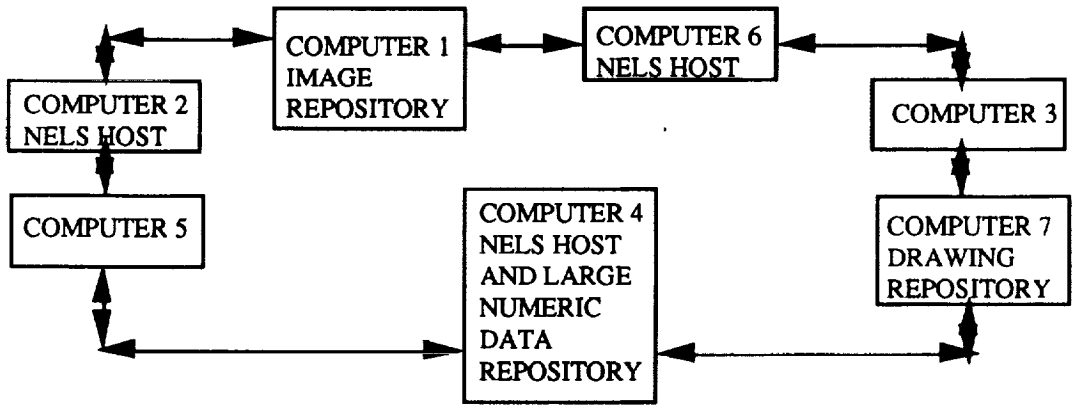

Figure 4: NELS distributed data base concept over a wide area network of computers.

NELS SCREEN EXAMPLES

NELS employs a Graphical User Interface (GUI) for screen display. With this type of interface, a user need only point with a mouse and click to select menu hierarchies, or operate screen buttons. Further, Motif window management software allows windows to be resized, or data to be viewed through the use of scroll bars. Finally, X-window/Motif software permits the inclusion of a number of file viewers. A viewer in this sense is a routine called by the window manager and passed the requested file as a parameter. Because of this, viewers may be added easily without disturbing the central application. Viewers presently supported in NELS 1.0 are ASCII, TIFF, GIF, raster, VE (a viewer for display of multiple images or bitmapped pages), and Interleaf. Viewers planned for implementation are autoCad, DECImage, and Post Script.

Figure 5 below displays the initial screen of NELS, which presents a list of the top hierarchies of the various sample libraries. A mouse click on any line would transfer the user to the next level of of organizational units, projects and offices. For known item searches, simply moving to the bottom of the list and displaying whatever objects are found there may remain the most efficient method of retrieval. The Function menu provides options for organizing the hierarchies by object class or aphabetically. Searching methods provided include natural language, query by example, and boolean logic. Collection defines searching depth, or the degree to which related collections are to be included in a given search. Admin lists functions available to Librarians such as adding and deleting collections and records, establishing users and user privileges and other functions. The screen overlayed in Figure 5 is the NELS search screen for natural language. The search request may be entered in simple English.

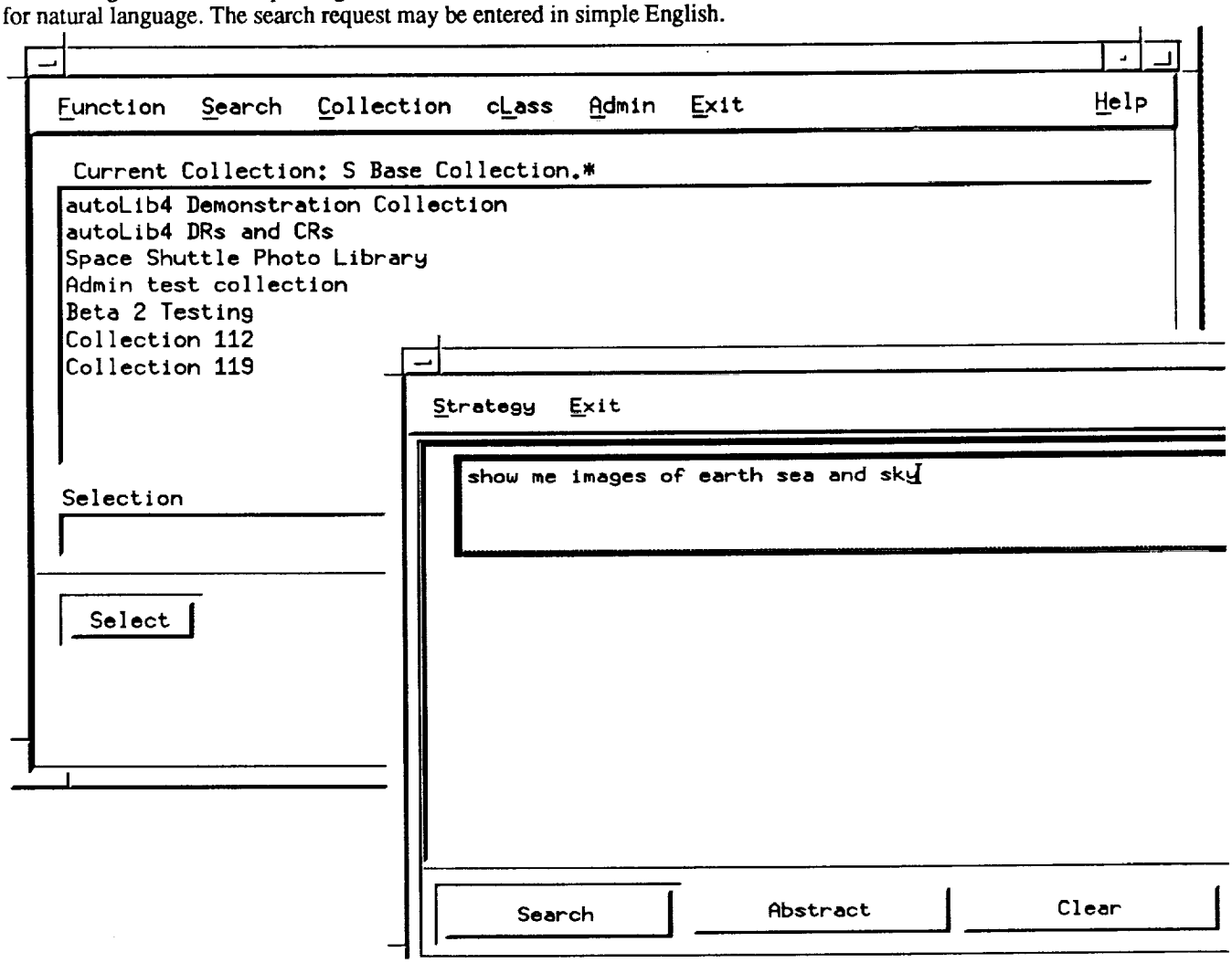

Figure 5: NELS main sign-on screen and natural language search screen.

Figure 6 below displays the ranked list of items resulting from the search for images made in Figure 5, with the retrieved image of the earth displayed in the lower left image corner. The numeric value shown to the left of the title on the object browser screen is the coefficient of correlation defined in the vector cosine formula shown earlier.

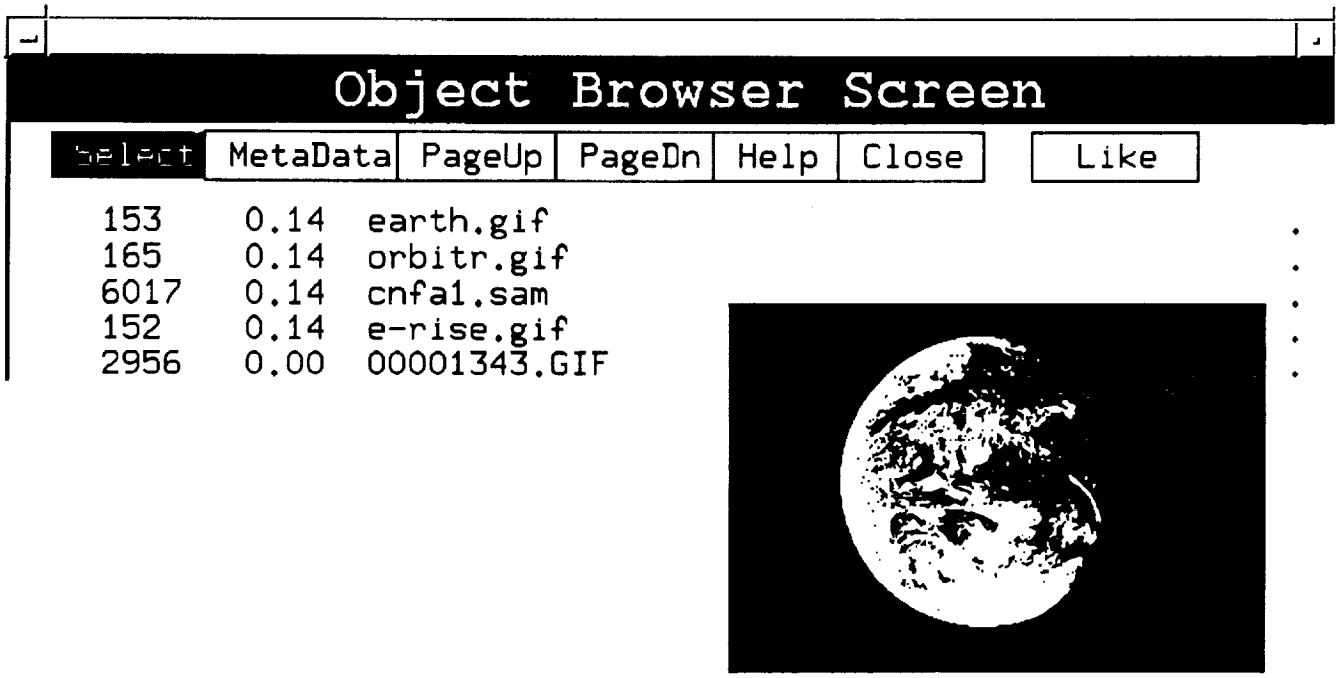

Figure 6: NELS object browser screen and retrieved image.

NELS SUPPORTED PLATFORMS

NELS is presently supported on the IBM RISC 6000 running under AIX, the DEC 5500 running under Ultrix, and the Sun 470 running under UNIX. Hewlett-Packard and Data General platforms are planned for future support. System access is supported for both general PC's and for Apple Mac II devices. Device classes for PC's are recommended to begin at the 386 level at 20 MHZ. Additionally, a minimum of 6MB of RAM memory is recommended. MAC II products are recommended to run System 6.05 minimally, with a minimum of 8MB RAM for best results. Both classes of devices require ethernet boards, however, these may obtained from a number of sources. X-software for PC's and Macintoshes has proliferated recently, and the market now offers a wide range of choices for this software as well [4].

REFERENCES

1. CLEVERDON, CYRIL W.; MILLS, J.; KEEN, MICHAEL. 1966. Factors Determining the Performance of Indexing Systems, Volume 1. Design. Volume 2. Test Results. Cranfield, England: ASLIB.

2. SALTON, GERARD; MCGILL,MICHAEL J. 1983. Introduction to Modern Information Retrieval. McGraw-Hill, New York.

3. RORVIG, MARK E. 1988 "Psychometric Measurement and Information Retrieval," In Annual Review of Information Science and Technology (ARIST), 23:157-189.

4. MCCOY, DANIEL J., 1991 "X Servers for PCs & Macintoshes," LibraryX, SoftwareTechnology Branch, NASA/Johnson Space Center, Loral Space Information Systems. NASA-JSC-PT4, Houston, Texas 77058. 1991