NOVEL INTERFACE FOR AN ONLINE PUBLIC ACCESS CATALOGUE: A CITATION NETWORK APPROACH

by

N. Somasekhara Karnavar

A Master's dissertation, submitted in partial fulfilment of the requirements for the award of Master of Philosophy degree of the Loughborough University of Technology

November 1992

Supervisors :

Ann O'Brien and Alan J. Poulter
Department of Information and Library Studies

ABSTRACT

The conventional subject search strategy of querying with words and phrases has been creating a lot of difficulties for the users of Online Public Access Catalogue (OPAC) systems because of the matching problems with the system vocabulary. An alternative is to use search by browsing through related records. In the proposed novel interface for the OPAC, a citation network approach is employed for subject access by browsing. The users can scan the references in a document and also the citations for a document and they can select the relevant materials from these related records. Abstracts are provided for the evaluation of the relevance. HyperCard, a hypertext-oriented software package for the Apple Macintosh Computer is used for the design of this novel interface. The aim of this OPAC design is to achieve a user-friendly interface which supports browsing and navigation. Facility for cocitation search is also provided. Through cocitation search, relevant and recent documents in a particular subject area can be retrieved by checking whether two useful documents are cited together. The interface design procedure is accomplished by using the HyperCard programming language HyperTalk. An evaluation of the prototype design is done through task performance tests and interviews. The values of citation search and cocitation search were appreciated by the evaluators, particularly by the subject specialists. A touch screen interface would be more useful in public access catalogues; since some users are inexperienced in using a mouse as a pointing device. This interface is feasible for a small subject area. But how this interface can be incorporated into a large OPAC system for providing useful browsing through citation networks is a matter of further research.

ACKNOWLEDGEMENTS

I remember all by benefactors at this time of submitting this dissertation. Particularly I should thank my supervisors, Ms. Ann O'Brien and Mr. Alan Poulter for their able guidance and concern for us.

I thank my family back in Kerala, India, also for sharing my responsibility while I have been away from home.

I am thankful to Mr. S.C. Biswas and Mr. Inderdeo for their help in many ways. I should specially mention Mr. B.G. Sundersingh for the personal and professional friendship which I cherish most.

I extend my thanks Mrs. Mary Ashworth for helping me out in the shaping of this dissertation.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1 : INTRODUCTION

1.1 An introduction to Online Public Access Catalogue (OPAC)

It was at the turn of the 1980's that Online Public Access Catalogues (OPACs) started replacing the card catalogues in the libraries of North America and Europe in a big way, as a natural sequel to the ongoing automation ventures. The society as a whole is being influenced by the computer revolution, and libraries have to keep pace with the technology to cater to the increased expectations of the users. A lot of literature on OPAC is now available. Efthimiadis and Neilson have listed 1226 unique items on OPAC in the second edition of *A Classified Bibliography on Online Public Access Catalogues* (1989)[1,2]. This clearly shows the importance shown to OPAC studies by the library profession.

The earlier online catalogues opened to public were just crude replications of the card catalogues. Because of the developments in Computer technology and OPAC research, there are a variety of online catalogue systems available now. Some of them are developed by commercial vendors (eg. Geac), while others are developed by either cooperative library networks (eg. Libertas of South West Automated Library Cataloguing Project - SWALCAP) or by individual library systems (eg. MELVYL of California University). With so many OPAC systems around, Hildreth has arrived at a generation-wise classification of OPACs[3]. This classification actually does not reflect any chronological delineation, but only 'tries to identify the qualitative stages of evolution in the design and production of online catalogues'[4].

The first generation Opacs were mostly automated circulation control systems opened for the general users as an afterthought. They provided only a very few access points, such as author, title, and class number. Subject access was very much minimal.

The second generation OPACs have been designed to improve search facilities. The techniques followed in conventional online information retrieval systems, like DIALOG, MEDLINE, etc. were heavily incorporated into OPAC design to attain this. They support search by Boolean logic and use keywords. The other capabilities of the second generation online catalogues include restriction of the searches by specified fields, taking care of truncation of words and limiting the results by date or publisher. Different choices for display formats are also available. Most of the vendor- supplied systems correspond to this generation.

But, the highly structured and command-driven online retrieval mechanism available with the second generation OPACs is suitable only for the experienced users. In fact, OPAC is meant for a much wider user population with varying competence and experience in using computers as well as in expressing search strategies. In such systems, matching of the user-input query terms and the system vocabulary creates problems for retrieval and the users feel unhappy at the terminal confronted with a cluttered screen with too many instructions.

The modern or the third generation OPACs are to be more user friendly and are expected to give the users satisfying results in their quest for information sources. Many research projects in different parts of the world are trying to tackle the problems of online catalogues. To quote examples, there are the ongoing research projects, such as OKAPI funded by the British Library Research and Development Department[5] and the HYPERCATalog project at the LIBLAB (Library and Information Science Research Laboratory) at Linkoping University, Sweden[6]. The CITE project at the National Library of Medicine[7] and DDC online project funded by Council on Library Resources (CLR), Forest Press (the Publisher of DDC) and Online Computer Library Centre (OCLC)[8] were some of the major initiatives in USA in this direction.

Traditionally library catalogues were monograph-oriented. Journal articles, conference papers, etc. which are very much in demand were generally not included in the catalogues. Access to these information sources have been through the commercial Abstracting and Indexing (A&I) databases. Even now, most of the OPACs give information only about the monograph collection. Thus, only a fraction of the materials available in the library gets represented in the OPAC. The online environment is fast changing this outlook. Advocating the inclusion of the records on journal articles, Barbara Quint has written, 'without good, widespread access to journal articles, however, a library is like an information palace without door knobs'[9]. She was airing a general feeling of the time. That was in 1987. The same year CLR (Council on Library Resources) and IBM (International Business Machines Corporation) supported an experiment at University of Illinois at Urbana-Champaign on incorporating direct online database search facilities through the OPAC[10]. A pioneer in this venture, MELVYL of the California University Library included Medline files in their system and made them available at selected terminals in November, 1987 for user access[11]. By 1988, several other libraries in USA started mounting commercial A & I database files to their OPACs. Online reference materials, such as dictionaries, encyclopaedias, etc. are now being mounted along with the OPAC. Online text retrieval is the next appealing option for the library users. As the technology develops further, the constraints of storage and other hardware-software problems may disappear to promote the wide-spread use of augmented OPACs.

The advances in information technology and communications technology has changed the library scene drastically. Thus, present day OPAC can give details on a variety of information sources. With networking facility, it is not confining its services or coverage to a single library. It can also contain information on documents which are not available in the particular collection. Networking facilitates speedy interlibrary loan which can be negotiated at the OPAC interface itself. In such an environment, OPAC needs some re-definition. Reflecting on this issue, Culkin acknowledged the relevance of the words 'online', 'public' and 'access' in 'Online Public Access Catalogue', but expressed some reservations on the 'Catalogue' part of the name because it connoted the traditional idea of confining the coverage to a single library system[12]. Meanings change over time to imbibe the new trends. Gregor and Mandel are of the opinion that 'the Catalogue no longer functions only "to show what the library has" (Cutter's second objective) but to show what the library can obtain for the user'[13]. Buckland views OPAC as 'the umbrella for the totality of bibliographical records *linked to holdings records* that a given library makes available'[14]. OPAC, of course gives status information and location information for documents in a single library or for a system of participating libraries. The idea of OPAC has considerably expanded over the years to represent its new role as an enhanced information retrieval system aspiring to become the 'library users' window on the world'[15].

With the proliferation and popularity of computerisation a new discipline, Human-Computer Interaction (HCI) has emerged and the end-user is getting the utmost attention. User interface design for ease and usability is the watchword. Since OPAC is an interactive system for the library users, ease in conducting a search, efficiency in retrieving the right documents and browsing facility are some of the attributes demanded at the user interface. Hypertext systems and WIMPS (Windows, Icons, Mouse, and Pointers) environment can be meaningfully used to achieve such user interface designs. Even the advanced techniques of artificial intelligence (AI) and expert systems are also being experimented at the design of user interfaces.

1.2 The Project

The proposed project is largely influenced by the following premises of Hildreth:

> Librarians have long understood the differences between 'known-item' searches and 'subject' searches in the catalogue. Now, we need to equally appreciate different kinds of subject searches and searching requirements. Alternative design models and retrieval methods which support these separate needs and modes of searching behaviour should be integrated into improved subject OPACs[16].

An alternative approach to an interface should take care of search by browsing. A citation network is a useful model for such a browsing system. Thus it has been decided that the objective of this project is to design a novel interface for OPAC to facilitate search by browsing through the references and citations. Since it is a citation network approach a cocitation search search strategy is also attempted.

1.3 The Dissertation

Chapter 2 of this dissertation reviews the issues in subject access at the OPAC with special reference to the use of browsing. Chapter 3 is on the importance of references, citations and cocitation search in subject access. Chapter 4 is on the characteristics of the proposed novel interface. Chapter 5 reviews the developments in user interface design and Chapter 6 describes the features of HyperCard as a User Interface Design Tool. Chapter 7 describes the design procedure of the novel interface. Chapter 8 is on evaluation of the developed interface and the results and Chapter 9 is for recommendations and conclusions.

REFERENCES

1. **Efthimiadis, Efthimis N. and Colin Neilson.** *A classified bibliography on online public access catalogues,* 1989.

2. **Efthimiadis, Efthimis N.** Online public access catalogues: characteristics of the literature. *Journal of Information Science,* 1990, 16(2), 107.

3. **Hildreth, Charles R.** Beyond Boolean: designing the next generation of online catalogues. *Library Trends,* 1987, 35(4), 647-667.

4. *Ibid.,* p. 648.

5. **Mitev, Nathalie N, Gillian M. Venner and Stephen Walker.** *Designing an online public access catalogue,* 1985.

6 **Hjerppe, Roland.** HYPERCAT at LIBLAB in Sweden: a progress report. *In* : Charles R. Hildreth, ed. *The online catalogue: developments and directions,* 1989, pp. 177-209.

7. **Doszkocs, T.E.** Natural language processing in an online catalogue. *Information Technology and Libraries,* 1983, 2(4), 364-380.

8. **Markey, Karen.** Subject searching strategies for online catalogues through the Dewey decimal classification. *In* : Charles R. Hildreth, ed. *The online catalogue: developments and directions,* 1989. pp. 61-83.

9. **Quint, Barbara.** Journal article coverage in online library catalogues: the next stage for online databases. *Online,* 11(1), 90.

10. **Mischo, William H. and Amy F. Moore.** Enhanced access to periodical literature in an online catalogue environment. *In* : Charles R. Hildreth, ed. *The online catalogue: developments and directions,* 1989, p. 110.

11. **Ritch, Alan.** Ten years of monitoring MELVYL: A librarian's view. *Information Technology and Libraries,* 1992, **11**(2), 179.

12. **Culkin, Patricia B.** Rethinking OPACs: The design of assertive information systems. *Information Technology and Libraries,* 1989, **8**(2), 172-177.

13. **Gregor, Dorothy and Carol Mandel.** Cataloguing must change. *Library Journal*, 1991, **116**(6), 43.

14. **Buckland, Michael K.** Bibliography, library records, and redefinition of library catalogue. *Library Resources and Technical Services,* 1988, **32**(4), 309.

15. **Fayen, Emily Gallup.** *The online catalog: Improving public access to library materials,* 1983, p. 110.

16. **Hildreth, Charles R.** General introduction; OPAC research: laying the groundwork for future OPAC design.
In : Charles R. Hildreth, ed. *The online catalogue: developments and directions,* 1989, p. 22.

CHAPTER 2 : SUBJECT ACCESS AT THE OPACS

2.1 Conventional subject access: problems and developments

The two maxims current about OPACs are:

1. Users prefer subject searches
2. Users experience difficulties in carrying out subject searches

The results of the first large scale study on the use of OPACs conducted in USA, supported by Council on Library Resources (CLR) corroborated these views[1]. Other studies also came out with the same conclusions[2,3].

The difficulties in subject search arise because of the person-to-person differences in describing documents and information needs. There can be three possible different perspectives on the same document in an information retrieval (IR) situation. The author describes the topic of the document through the title, of course, if it is not fanciful. The author's view may reflect a 'subjective' approach to the subject of the document. The indexer/cataloguer attempts to describe the subject of the document as objectively as possible using a controlled vocabulary of either a list of subject headings or a thesaurus. But the request for the same document may come up in an entirely different expression. The user's description depends on the experience in the subject field, the aspect of the subject in which he/she is interested, and the language skills. Or the user may envisage a new use and hence a valid new description for the document in question, which the author or the indexer has not even dreamed of. Information retrieval (IR) research right from the time of the introduction of online search services on Abstracting and Indexing (A&I) databases, has been trying to achieve the matching of the potentially divergent views on the subject of documents at the search interface.

IR research coined the terms precision and recall, to evaluate retrieval output after a search for a topic. The efforts to achieve precision may result in search failure and a liberal approach may result in recall of many documents resulting in information overload. Larson[4] has metaphorically described the situation as the danger of being between the Scylla of search failure and Charybdis of information overload. Related to this is the controversy on the use of free-text or natural language vocabulary as indexed from the title, abstract or even the full text, if available vs. the controlled vocabulary, which is still to be settled. In controlled vocabulary again there is the controversy on the use of pre-coordinated list of subject headings vs. thesaurus which warrants post-coordination of terms. The emergence of terms/keywords as the main form of access to information retrieval systems necessitated the introduction of Boolean logic. But, soon it was found that in the case of systems like OPAC where users are not trained in search strategies, Boolean operators became a problem[5]. First of all, there is a usual tendency among the users to confuse the Boolean "AND" with the "OR". Even if the user learns the correct meanings of the Boolean operators, there are still difficulties in applying the exact combinations to achieve the right results. Another problem with the Boolean operation on information retrieval systems is search failures or too little output when three or more search terms are combined using "AND".

Some other well-known problems at the search interface include typing errors, differences in spelling practices (British and American), abbreviations and acronyms, plural and singular forms, grammatical differences in presenting terms like the practice of using "ing" or the noun form, homonyms, synonyms, etc. In a review, Larson has listed the developments in IR research which were aimed to tackle these problems[6]:

1. Lead the user from free-text terms to the corresponding subject headings or class numbers associated with potentially relevant items.
2. Alleviate search failure through stem searching, spelling correction, and partial matching of queries.
3. Rank the retrieved items in order of their probability of relevance to the user's query.
4. Facilitate more open-ended searching and search expansion through the use of relevance feedback.

Some of these remedies are already being implemented in some of the operational OPAC systems. For example, CITE system at the National Library of Medicine (USA) has incorporated the concept of 'relevance feedback'[7]. At the CITE interface the user's query terms are automatically stemmed, the spellings are corrected, and term frequency weighting is done. The resulting terms are then mapped with the subject headings from MeSH.The user is then asked to confirm or alter the ranking of the terms. The approved terms are then input as the query. The retrieved records are displayed in a ranking order. The user is again asked to indicate the relevant records. The system uses this relevance feedback to formulate a new query adding the subject headings from the relevant records. The search becomes interactive. Likewise, Libertas of SLS (SWALCAP LIBRARY SERVICES) in UK has implemented some of the solutions from the prestigious OKAPI project[8]. In Libertas, instead of the use of Boolean operator "AND", each term in the search is assigned a weight, which is inversely related to the frequency of the term and the search output is presented in the form of a ranked list of documents with the most likely to be relevant coming first.

The problems discussed above are encountered usually at the conventional form of search using search terms and phrases. The classic model of IR, as shown in Figure 1, depends on the match of the document representation and the query[9]. Palay and Fox have termed this kind of search as 'Parameterised Search' and characterised it as 'strictly focused, in the sense that the user must specify exactly the set of attributes that the records must contain and/or satisfy' as different from the search by browsing[10]. Hildreth also differentiated these two forms of search and called them "query searching" and "browse searching"[11]. Nowadays, browsing as a search procedure is getting a lot of attention.

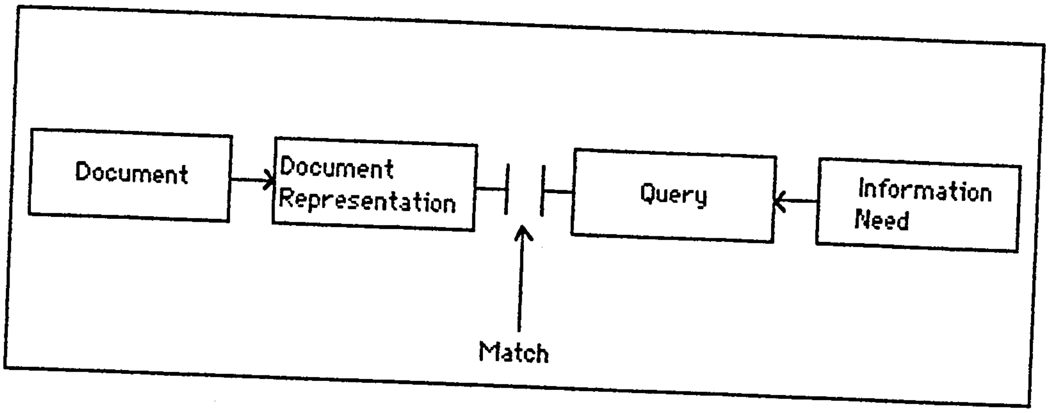


Figure 1 Classic information retrieval

2.2 Browsing for subject access

Browsing in the library context is an acknowledged form of access to the collection to find out useful materials. According to Apted[12] browsing through books and journals is 'general purposive browsing' and browsing through library catalogues or bibliographies is 'specific browsing'. Specific browsing is rather done with a particular purpose in mind and it is not done in a haphazard way.

Serendipity is one of the products of browsing. More than the possible serendipitous discovery of useful materials, browsing is credited to foster creativity[13]. Commenting on browsing of mass media, Davis wrote:

> The human brain can store and retrieve billions of bits of information, combine them in often almost unbelievable ways, and create concepts and knowledge that never before existed. To do this, the most diverse and detailed input is needed.[14]

The quality of browsing as enhancing creativity is well accounted. Martin[15] pleaded for end-user searching of A&I databases since it would lead to browsing, thus promoting conducive atmosphere to get innovative ideas.

With the provision of abstracts and meaningful linking of records, browsing of OPAC can be made a creative process. Browsing at the OPAC is considered a 'heuristic search'. Some of the browsing heuristics a searcher might follow, as adapted from Thompson and Croft[16], are described below.

If the current document is interesting, a searcher may like to see:

1. Other works by its authors.
2. References listed in the document.
3. Citations for the document.
4. Other papers in the same conference proceedings or other articles in the same journal issue, depending on the nature of the document.

Several such browsing links can be thought of. Without the difficulties encountered at the conventional search systems, subject access is possible by browsing through such meaningfully linked documents.

Another advantage of browsing is that user can evaluate the utility of the information source at the browse mode comparing the information provided with the document description with his own requirements and select the record. This conforms to the 'berrypicking' model of searching suggested by Bates[17]. A major aspect of the berrypicking model is giving enough facility for browsing. Bates has criticised the 'lingering tendency in information science to see browsing *in contrast* to directed searching, to see it as a casual don't-know-what-I-want behaviour that one engages in separately from regular searching'[18]. Ellis[19] has emphasised the role of browsing as a standard way of information searching.

The responsibility is on the user. The system, in this case, does not sit at value judgment for the relevance of the documents. But, the system should take care that the user is not lost in the maze of records. There should be proper navigational facilities.

The development of Hypertext systems is another factor which has stimulated added enthusiasm for browsing. Now linking of records has become an easy possibility, thus paving way for the design of OPACs with browsing facilities.

2.2.1 Hypertext systems for browsing

Any treatise on hypertext can not but mention that seminal article, 'As we may think' by Vannevar Bush[20], which was published in 1945. His ideas were very innovative. He identified the associational nature of human mind and wanted an information retrieval system based on the associational links. He even suggested a system called "Memex" for this purpose. The technology of the time was not so much conducive for the production of such a machine. Later Douglas Engelbart, acknowledged as the inventor of word processing, screen windows and 'mouse' device, worked on the principles of linking texts and developed a system called NLS (oN Line System) which had non-hierarchical links[21]. But the credit for popularising the concept of such links and coining the term 'hypertext' goes to Ted Nelson. He is pursuing the project 'Xanadu' for the fulfilment of his ambition to have a single system for all the world literary output which can be followed by associative links[22].

Work on hypertext created some operational systems. The first to come out was NoteCards developed by Xerox Pao Alto Research Center. This card metaphor was used in Apple computer's more famous HyperCard. The linking is between records in the form of cards. NoteCards has got extensive use of windows. It is possible to create displays of the NoteCards hierarchy for browsing.

NoteCards was developed as a general purposive hypertext system. HyperCard also uses linking and browsing mechanism through its buttons. An extensive description of HyperCard is given in Chapter 6.

Another earlier hypertext system is 'Guide' developed at the University of Kent. Later Office Workstations Limited (OWL) marketed this product. Guide database consists of a hierarchically structured text with cross reference connections. Text and graphics can be used in Guide database creation. Guide uses a scrolling method as in word processing systems. A guide document is a single scrolling field. Within text there are buttons which when selected gives the associated text.

'A hypertext system consists of *nodes* ('chunks') of information and links between them'[23]. Database management systems also can develop various links, as in relational and object-oriented links. But such systems emphasise on search by some criteria and report production. Hypertext systems, on the other hand, rely on reading and browsing.

Conklin commended the browsing capacity of hypertext systems and at the same time he was anxious about the user disorientation in complex hypertext systems[24]. Navigation is a problem in such cases. But when the organisation of the system is made clear instances of getting lost in the network can be made minimal. By giving proper messages as to where one is, the system can give the much needed guidance. Graphical browsers and maps with nodes as documents can also help in navigation. But such representations is possible only with small sets of information.

REFERENCES

1. **Matthews, Joseph R., G.S. Lawrence and D.K. Ferguson.** *Using online catalogs: a nation wide survey,* 1983.

2. **Markey, Karen.** Subject searching experiences and needs of online catalog users: Implications for library classification. *Library Resources and Technical Services,* 1985, **29**(1), 34-51.

3. **Lipetz, B and P.J. Paulson.** A study of the impact of introducing an online subject catalog at the New York State Library. *Library Trends,* 1987, **35**(4), 597-617.

4. **Larson, Ray R.** Between Scylla and Charybdis : Subject searching in the online catalog. *Advances in Librarianship,* 1990, **15**, 225.

5. **Cooper, William S.** Getting beyond Boole. *Information Processing and Management,* 1988, **24**(3), 244.

6. **Larson,** ref. 4, 218.

7. **Hildreth, Charles R.** *Intelligent interfaces and retrieval methods for subject searching in bibliographic retrieval systems,* 1989, p.89.

8. **Hancock-Beaulieu, Micheline.** Userfriendliness in online catalogues. *Program,* 1992, **26**(1), 33-34.

9. **Bates, Marcia J.** The design of browsing and berrypicking techniques for the online search interface. *Online Review,* 1989, **13**(5), 408.

10. **Paley, Andrew J. and Mark S. Fox.** Browsing through databases. *In* Oddy, R.N. et al. *Information Retrieval Research,* 1981, p. 310.

11. **Hildreth, Charles R.** General introduction: OPAC research: laying the groundwork for future OPAC design. *In*: Hildreth, Charles R. ed. *The online catalogue: developments and directions,* 1989, p. 10.

12. **Apted, S.M.** General purposive browsing. *Library Association Record,* 1971, 73, 228.

13. **Bawden, David.** Information systems and the stimulation of creativity. *Journal of Information Science,* 1986, **12**(5), 203-216.

14. **Davis, Watson.** The universal brain: Is centralised storage and retrieval of all knowledge possible, feasible or desirable. *In:* Kochen, Manfred, ed. *Growth of knowledge: readings on organization and retrieval of information,* 1967.

15. **Martin, Peter.** The innovation process and the online information channel. *In:* Stern, B.T., ed. *Information and Innovation,* 1982, p. 159-168.

16. **Thompson, R.H. and W.B. Croft.** Browsing in text retrieval systems. *International Journal of Man-Machine Studies,* 1989, **30**(6), 639-668.

17. **Bates,** ref. 9, 414.

18. *Ibid.*

19. **Ellis, David.** A behavioural approach to information retrieval system design. *Journal of Documentation,* 1989, 45(3), 187.

20. **Bush, Vannevar.** As we may think, 1945. *Reprinted in:* Kochen, Manfred, ed. *Growth of knowledge: readings on organization and retrieval of information,* 1967.

21. **McKnight, Cliff and Andrew Dillon.** *Hypertext in context,* 1991, p.9.

22. **Nelson, Theodor H.** Managing immense storage. *Byte,* 1988, 13(1), 225-238.

23. **McKnight,** ref. 21, p. 2.

24. **Conklin, Jeff.** Hypertext: an introduction and survey. *Computer,* 1987, **20**(9), p.38.

CHAPTER 3 : REFERENCES, CITATIONS AND COCITATION SEARCH

3.1 References and citations for subject access

Browsing through footnotes, references and bibliographies in books and articles has been an accustomed method pursued by the researchers for getting information sources. INFROSS study on social scientists revealed a clear preference for this kind of search[1]. Likewise the study conducted by Van Styvendaele among the engineers, scientists, and social scientists in an academic environment also reported the popularity of footnotes and lists of references in documents as the sources for materials of interest[2].

The two terms 'references' and 'citations' are sometimes used interchangeably in some contexts by some authors. But, in this project the difference between the two terms are succinctly delineated as follows. The items given in the list of references appended to a document are called 'references' in that document. When that particular document is referred to by a number of other documents, these documents are called 'citations' to that document. Fig. 2 shows the relation between a citing and cited document. Document D1 is cited by Document D2. In turn it can be told Document D2 is citing Document D1. D1 is the reference in D2 and D2 is the citation for D1. There is a time factor also involved in the citing - cited relationship. Since the references given in a document belong to a date prior to that document, the following up of references is termed 'backward chaining'[3].

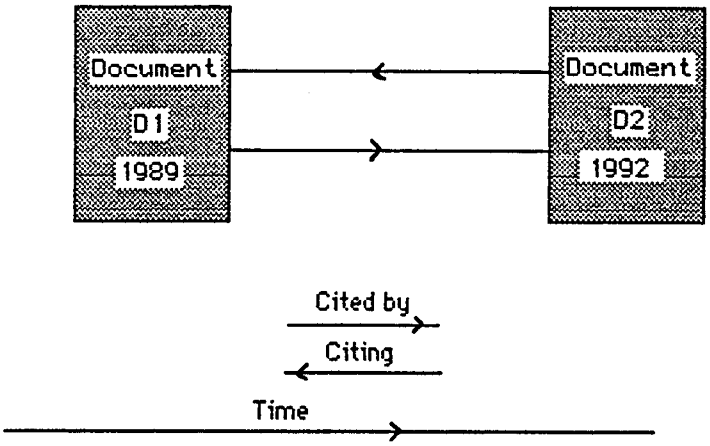


Figure 2 Citing-cited relationship

If a document is relevant, it is presumed that its citations are also relevant. Following up citations is called "forward chaining'[4], since the citations for a document can happen at any point of time in the future. The information retrieval effectiveness of citations are used in the citation index services offered by the Institute for Scientific Information (ISI), Philadelphia. The commercial success of the citation indexes by ISI proves the value of citations. Eugene Garfield, the founder of ISI commenting on the usefulness of citation notes that it is 'a precise, unambiguous representation of a subject that requires no interpretation and is immune to changes in terminology'[5].

The citation is always related in some way to its subject content or methodology adopted for the work. Brooks identified seven motivational factors for the citing of documents[6].

1. Currency

   Current documents are cited for the prestige factor involved in the up-to-dateness of the research.

2. Negative credit

   Authors have to criticise, correct and disclaim and dispute other works citing the questioned documents. This kind of error detection is considered useful contribution to the field.

3. Operational information

   Methodologies and information about apparatus, tools are borrowed from the cited paper.

4. Persuasiveness
Researchers refer to documents to establish the correctness of their methods and results.

5. Positive credit
Documents by pioneers and other such related works are cited giving due recognition to the earlier works in the field.

6. Reader alert
References are sometimes given for background reading for those who are interested in the developments in the field.

7. Social consensus
Some works are cited just because they are well established papers and they are expected to be quoted.

Garfield has reported a study by Vinkler in which the motivations for citing are classified in to two main groups, 'professional motivations' and 'connectional motivations'[7]. Professional motivations refer to the contributions of theoretical and practical aspects acknowledged by citing the documents. Connectional motivations evolve from the relationship between the citing authors and cited authors. But, most of the references are for professional motivations only.

Searching through references and citations brings to the fore the multidisciplinary approach, which the traditional search by query can not do. Another major advantage of references and citations is that there is no semantic problem which usually affects the query search by keywords and phrases. The citation link does not depend on the changing of the names of concepts and techniques in subject fields.

Redundant and perfunctory citations can cause some problems, since retrieval of such materials does not serve any purpose. But Egghe and Rousseau[8] are of the opinion that redundant citations may be signalling simultaneous and independent discovery.

3. 2 Cocitation search

As a measure of relation between two documents Kessler[9] proposed Bibliographic coupling between two records. If there is one common reference between two records, it is considered that the two records are related with a bibliographic coupling strength of one. Innovating on this connection Small[10] proposed a new measure of relation called cocitation which is considered to be more strong (It is reported that Marshakova also arrived at this concept independently[11]). If two documents are cited together by a document, they are said to be cocited. The three documents thus connected must be having subject similarity or associated by the co-occurrence of ideas. Figure 3 illustrates a cocitation context. Documents A and B are two cocited documents. X and Y are two documents cociting them. Therefore, cocitation is recommended for information retrieval purposes. The documents retrieved by cocitation search by checking if two pertinent documents are cited together are found to be useful and recent. Chapman and Subramanyam in a study has noted that a cocitation search by specifying documents retrieves fewer, but highly relevant

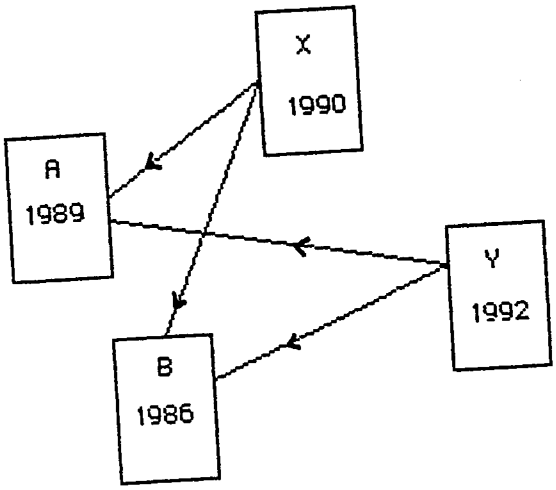


Figure 3 Cocitation relations

documents and suggested cocitation search where precision is at a premium[12]. While the bibliographical coupling strength of two documents remains static, the cocitation of two documents can give newer and newer documents over the years until the cocited documents become obsolete.

The relevance of the cociting documents is acknowledged by the use of these documents to enhance indexing. The titles of the documents derived from a cocitation search are used to add more index terms, on the premise that they will be relevant[13].

Cocitation search is not offered by any operational systems. Even in citation index services offered by ISI, cocitation search can be done only by expert searchers using the Boolean operation of intersecting (using the operator 'AND') two sets of citation searches.

REFERENCES

1. **Bath University.** *Information requirements of researchers in the social sciences. Vol.1 Text.* 1971, p. 216.

2. **Van Styvendaele, B.J.H.** University scientists as seekers of information: sources of reference to periodical literature. *Journal of Librarianship,* 1977, 9(4), 271.

3. **Ellis, David.** A behavioural approach to information retrieval system design. *Journal of Documentation,* 1989, 45(3), 183.

4. *Ibid.*

5. **Garfield, Eugene.** *Citation indexing - its theory and application in science, technology, and humanities,* 1979, p. 3.

6. **Brooks, Terrence A.** Private acts and public objects: An investigation of citer motivations. *Journal of American Society for Information Science,* 1985, **36**(4), 223-229.

7. **Garfield, Eugene.** Citation behaviour an aid or a hindrance to information retrieval. *Current Contents,* 1989, 18, 4.

8. **Egghe, Leo and Ronald Rousseau.** *Introduction to informetrics: quantitative methods in library, documentation and information science,* 1990, p. 252.

9. **Kessler, M.M.** Bibliographic coupling between scientific papers. *American Documentation, 1963, **14**, (1), 10-25.*

10. **Small, Henry G.** Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science,* 1973, **24**(4), 265-269.

11. **Egghe,** ref. 8, p. 239.

12. **Chapman, Janet and K. Subramanyam.** Co-citation search strategy. *In:* Williams, Martha E. and Thomas H Hogan, comps. *Proceedings of the National Online Meeting,* 1981, p. 102.

13. **Kwok K.L.** On the use of bibliographically related titles for the enhancement of document representations. *Information Processing and Management,* 1988, **24**(2), 123-131.

CHAPTER 4 : THE NOVEL INTERFACE: PROPOSITION

4. 1 The characteristics of the Novel Interface

The novel interface is based on the presumption that browsing through the citations and references of relevant documents can help in retrieving useful materials. There is no provision in this OPAC for conventional search by keywords or subject headings. The access to the catalogue is directed by the usual information seeking patterns of researchers. Ellis has found that researchers in social sciences usually start the literature search by seeking the help of experienced people in the subject area and asking them for references to introductory works, key references and key authors[1]. This kind of starting the work on information search is true for any branch of knowledge. Taking the cue, the novel interface provides a list of authors in the subject area for access to the catalogue.

Another access point is through the highly cited documents in the subject area. 'Highly cited documents can be considered as 'exemplars' or 'concept symbols': illustrations of methods, or theories which comprise the essential repertoire of techniques for practitioners of a speciality'[2].

Once access is established user can go back and forth through citation network comprising of references in documents and citations to documents. Figure 4 illustrates the citation network. An important point to be considered is to provide navigational aids so that the user is not lost in the network of citations and references. The user can evaluate the usefulness of the record by

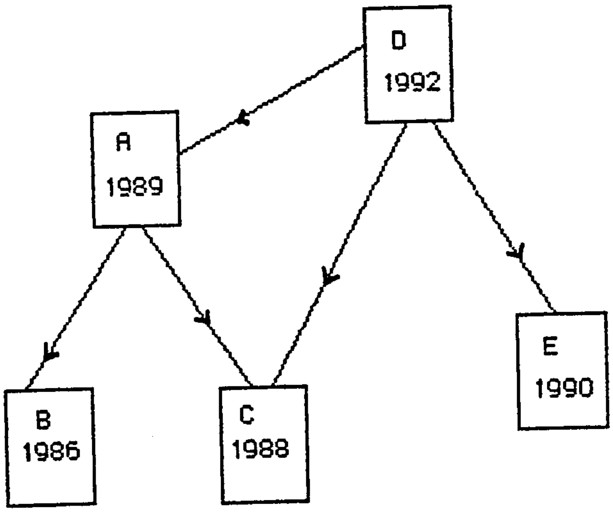


Figure 4 A citation network

going through the abstract provided. Once the usefulness is ascertained the document can be selected to be included in a list which can be printed at the end of the session. Information on the status of the document as to whether it is available in the library or whether it is on loan is to be provided. Ideally reservation and interlibrary loan request also should be possible at the novel interface.

An important feature of the novel interface is the facility for cocitation search. A list of cited works in the subject area is presented to the user so that a cocitation search is possible as easily as possible by selecting the documents.

4.2 Similar systems and the novel interface

4. 2. 1 Citation index services from the Institute for Scientific Information (ISI)

Citation approach is almost synonymous with the well-known citation index services of the ISI. The Online search services for SciSearch, Social SciSearch and Arts & Humanities Search are available in UK through the Joint Academic Network (JANET). CD-ROM version of SCI and SSCI are also available. Menu-driven interfaces are used for all these. Cocitation search is not available in any of these indexes. It can be done by ANDing together the sets from two citation searches. But, for an untrained user it is a difficult procedure. In the CD-ROM version a 'related records' search is possible. This is done through bibliographic coupling and the retrieved set is presented in a ranking order according to the bibliographic coupling strength. Thus the records having the most number of common references with a particular document comes first. Only a maximum of twenty such records are shown. Tseng, et al. studied the CD_ROM version of SCI and found this offer of "Related Records" very convenient and useful[3].

The search on Science Citation Index through JANET is not particularly user friendly. When citation searching is requested, the following menu appears:

```
            R - Search cited references
            A - Inspect cited author index
            J - Inspect cited journal index
   or go to Search (S) Options (O) Citations (C) Issues (E)
   or type HELP or Exit
```

On entering R for searching cited references: a form filling type screen comes, where it is asked step by step the following questions:

```
            Enter citation author name,
            Enter year or a range of years,
            Enter a volume number,
            Enter a page number, and
            Enter a journal expression
```

Then if an author name is entered, a screen giving the citations for the author is given in the following form,

```
                                                               Sources
 [A]  SPIELBERGER_RK, 1984 VOL.7 P.193, IEEE T COMPON HYBR       10
```

If more than one cited article for an author is present, all the articles are shown with consecutive letter tags. The documents will be then arranged according to the year of publication. Documents can be selected by tagging. The citations for the tagged articles can be displayed or the output can be downloaded into one's E-Mail file.

With the cited author index option, the author name can be given for finding the total references for a particular author. Journal name also can be likewise checked for knowing the citations. But the entry of the Journal for checking is the biggest problem in SCI. There is no standard way of expression for any journal. It can be entered as truncated words in the title using the wildcard * and employing the Boolean expressions AND or OR. A check for the journal, 'IEEE Transactions on Components, Hybrids, and Manufacturing Technology' gave the following interesting results:

IEEE T COMP HYBR MAN
IEEE T COMP HYBR MFR
IEEE T COMP HYBRID M
IEEE T COMP HYBRIDDS
IEEE T COMPON HYBR
IEEE T COMPONENTS HY

Several other such combinations for the same journal can be seen. There is no standardisation for it. To be fair to the ISI, they acknowledge the inconsistency in Journal expression and warn that it is difficult to discover all the possibilities using it. But this discrepancy led to the splitting up of a total of 16 citations for the 1984 article by R.K. Spielberger into two different entries with ten and six citations, the difference being the rendering of the journal expression.

4. 2. 2 Intelligent Interface for Information Retrieval (I^{3}R)

Thompson and Croft[4] have reported setting up of a browsing system, called Intelligent Interface for Information Retrieval (I^{3}R) for text retrieval. For browsing a citation network is used. The system intelligently processes the query in natural language to extract terms and incorporates also the earlier domain knowledge level of the user if the user has previously used the system to reformulate the query. Then the documents retrieved are shown for approval. If a relevant document is selected a browse option is given. A graphical representation of nodes are used to initiate browsing. These are called neighbourhood and context maps. In the neighbourhood map the central node represents the document selected and the surrounding nodes are provided by the system as likely to be interesting depending upon whether they are related because they are referred to in the document or cited by the document. In context map the nodes are marked as to whether they are visited or judged relevant or recommended. The system is primarily window oriented and it is implemented in Common Lisp on a DEC VAX Station II. This is mainly a full text oriented system incorporating both query search and browsing. It is a highly evolved system.

4. 2. 3 Transient Hypergraphs for citation networks[5]

By this system browsing through citations and references is possible. Cocitation search is also attempted. HyperCard package was first used to construct 'hypergraphs' to graphically represent the document nodes in a citation network of citing and cited documents. But, the version of HyperCard available at the time was found to be not suitable for transient hypergraph system since the HyperCard links were to be made at the database itself and they were permanent fixes. Transient means, the links created should be dynamic and should last only for the particular session. It was later implemented on INGRES on a Sun work station.

HyperCard 2.0 version and the programming language HyperTalk version 2.0 are much more versatile than the earlier versions. The features of HyperCard and HyperTalk are discussed in Chapter 6. Creation of transient network views is now possible with HyperCard. The links can be made dynamic through scripting by the added facilities of HyperTalk version 2.0. Different views and linkings of the database can be shown without actually changing the internal structure. Thus the novel interface is implemented on HyperCard version 2.0.

4.3. Novel interface: Design considerations

Browsing links are to be created instantaneously for the session and on ending the session these links must vanish. The database should not be permanently affected by the creation of the links. The browsing is aided by navigational cues so that the user may not get lost in the network links. Always one may come back to the original position and start new searches. For example, both forward chaining from a document through its references and backward chaining through its citations should be possible in a single session.

Usual user interface design guidelines are to be followed for the creation of an easy-to-use system. Novel interface should be aesthetically and functionally attractive. Screen design, presentation of information, and designing of the objects meant for direct manipulation of the interface are all important facets to be considered. User interface design itself is now a major domain attracting the attention of the computer world.

REFERENCES

1. Ellis, David. A behavioural approach to information retrieval system design. *Journal of Documentation,* 1989, 45(3), 171-212.

2. **Egghe, Leo and Ronald Rousseau.** *Introduction to informetrics: quantitative methods in library, documentation and information science,* 1990, p. 252.

3. **Tseng, Gwyneth, Goff Sargent and Jack Meadows.** Round referencing. *Nature,* 1989, 337, 516.

4. **Thompson, R.H. and W.B. Croft.** Browsing in text retrieval systems. *International Journal of Man-Machine Studies,* 1989, 30(6), 639-668.

5. **Shepherd, Michael A. and C.R. Watters and Yao Cai.** Transient hypergraphs for citation networks. *Information processing and management,* 1990, 26(3), 395-412.

CHAPTER 5 : INTERFACE DESIGN ISSUES

5.1 User interface design: General considerations

In the early years of computers, the concept of user interface was not important. During those days engineers and programmers were the users. Grudin in a historical analysis of user interfaces has observed, 'the principal focus of activity in computer development has moved gradually from hardware to software and is now shifting toward the user interface'[1]. Through the 60's and mid-70's computer programmers remained the principal users. At that time, improving the user interface meant improving programmer efficiency. Emergence of interactive terminals and personal computers shifted the emphasis from programmers as users to programmers developing better interfaces for non-programming end-users. This was the time when wordprocessing programmes and spread sheet programmes flooded the market. A new discipline human-computer interaction emerged. It involves a distinct multidisciplinary approach drawing expertise from cognitive psychology, ergonomics, graphical arts, etc. Technological advances that help in the betterment of human-computer interaction include 'the availability of memory and processing power, more sophisticated approaches to knowledge representation in software, and the spread of multitasking operating systems and software products that motivate a greater focus on dialogue and task organization'[2].

Because it is a new domain the terminologies have not yet stabilised in this subject area. Various combinations of words are now used to describe the interaction between humans and computers: 'Human-computer interaction (HCI)', 'Computer-human interaction (CHI)', 'Computer-human interface (CHI)', 'Computer interface', 'User interface', and just 'interface'. A user interface can be generally defined as 'a mechanism that facilitates the flow of information between a computer and a human'[3]. There is a simplistic view of user interface as a front end to a system. But in systems where the data structure itself is to be used for the presentation of proper expressions to satisfy user needs, it is better to view the interface design as the total system design. As Yee has put it[4]:

> It is irrelevant to the user whether the solution is part of the software or part of the data structure. This is not to suggest that one of the solutions may not be preferable in terms of reliability, efficiency, or clarity of the user-computer dialogue, but rather to suggest that design of an effective user-computer interface requires effective design as a whole, including relationships and structures inherent in the data itself.

In OPAC systems, where the links are to be brought to the notice of users, there should be good planning even at the database design stage itself for the development of the interface.

Guidelines for effective human-computer interaction at the interface have been given by many researchers in the area. It is reported that Smith and Mosier have compiled a mammoth list of 944 guidelines for the human-computer dialogue[5]. Barker summarised the guidelines put forth by many researchers into a convenient list of six basic principles[6].

1. Be logical

The logical nature of information processing by the computers is a fact well-known among the users. Thus this logicality should be extended to the interface domain. The tasks to be followed by the users should be arranged logically, naturally and smoothly.

2. Make interfaces and dialogues appropriate.

When designing interactive systems, it is important to ensure that the interface system and its supporting dialogue are appropriate to the requirements.

3. Consistent

Consistency in presenting a certain information which is to appear in all screens is a good design principle.

4. Be humane

It is necessary to give the users a feeling that he/she is part of a well-designed human-computer activity system, as in: giving the error messages, the choice of appropriate terminology for messages, and the nature of tasks to be performed by the users. They should not be overloaded by information or with tasks.

5. Be moderate

This is corollary to the requirement to be humane. It is very important to avoid cluttering of the screen. Overuse of colour or sound is to be avoided. Ergonomically sound, and aesthetic application of audio- and video-effects is to be encouraged. Users should not be confused by offering too many options. The well known 'seven plus or minus two' theory of

G A Miller[7] is to be remembered when the screen is designed. More than nine objects or options will burden the users. Enough vacant spaces should be there on the screen.

6. Make the interface adaptable.

Users, the human component of the system, are all different in their outlooks, temperaments, and perceptions. Physically disabled users also are to be taken care of while designing a system. For example, if colours are used to denote certain aspects on the screen as part of the interface, it will be difficult for the colour-blind people to use the system. It is reported that eight percent of the males of North America and Europe are colourblind[8]. The interface should be adaptable to cater to a wide variety of users.

Reliability and robustness are two other important qualities expected from a user interface[9]. Reliability is a measure of whether the system performs correctly. It should do the intended function very well. Robustness is a measure of the system tolerance of users' mistakes. A wrong move from the user should not result in a fatal error for the system, which may frighten users. As far as possible, there should be a facility for the actions to be made reversible, if warranted. Irreversible functions should be confirmed again with a message before execution.

Visual clarity is another important factor which makes a successful design. Information should be grouped meaningfully to

attract the user's attention. Information displayed on the screen should be clear, well-organised, unambiguous and easy to read. Aesthetic considerations should guide the screen design.
Online user guidance should be provided to help the user. The system should explain the options available for the benefit of the users. Users may forget the aspects of the system after some time if they are not continuously using a system. So help provision is a must in any interface.

A good interface should be easy to learn without much training. The need for memorisation should be minimal. Even a manual should be unnecessary for a good interface. The earliest user interfaces were all command-based. One deterrent factor for users was the need to memorise the commands. When commands are given on the screen as prompts, the screen becomes cluttered. Some of the OPAC interfaces are still like this. Menu-based interfaces were introduced to help users in using the commands without the burden of remembering them. One criticism against the menu-driven system is that only preplanned paths can be followed.

5.2 Direct manipulation interfaces

Direct manipulation with WIMPs (Windows, Icons, Mouse and Pointers) and object-oriented programming changed the whole scene of interface design. Object-oriented programming is defined as 'a method of implementation in which programs are organised as cooperative collections of objects, each of which represents an instant of some class, and whose classes are all members of a hierarchy of classes united via inheritance relationships'[10]. Since direct manipulation involves graphical representations it is also termed graphical user interface (GUI). When Apple's Macintosh computer was launched with graphical user interface (GUI) features it was presented as "the computer for the rest of us"[11] to distinguish it from the contemporary computers which needed a lot of training for use. A graphical user interface replaces the complex command language of a computer's operating system with something that is easy to use and comprehend. The software packages and the designed products using them all now try to become user friendly. It has been welcomed by many because any product that can reduce training time would be invaluable[12]. With direct manipulation, there is no need for commands or menus. The interaction with the computer is by pointing with a mouse or by touch at visual representations of actions (icons) or at words in the text. According to Shneiderman, 'dealing with representations of objects may be more "natural" and closer to innate human capabilities: action and vision skills emerged well before language in human evolution'[13]. The features of direct manipulation interfaces and their beneficial attributes are summarised as follows[14]:

a) Features of direct manipulation interfaces

1. Continuous representation of the objects and actions of interest

2. Physical actions or labelled button presses instead of complex syntax

3. Rapid incremental reversible operations whose impact on the object of interest is immediately visible.

b) Beneficial attributes of direct manipulation interfaces

1. Novices can learn basic functionality quickly, usually through a demonstration by a more experienced user

2. Experts can work rapidly to carry out a wide range of tasks, even defining new functions and features

3. Knowledgeable intermittent users can retain operational concepts

4 Error messages are rarely needed.

5. Users can immediately see if their actions are furthering their goals, and, if not, they can simply change the direction of their activity

6. Users experience less anxiety because the system is comprehensible and because actions are so easily reversible.

7. Users gain confidence and mastery because they are the initiators of action, they feel in control, and the system responses are predictable.

As part of the research on direct manipulation interfaces, studies on pointer devices have been conducted. Mac Aogain and Reilly[15] have recommended the silent parking of the cursor by mouse click as against the mouse click accompanied by audio 'beep' signals from the system. Another study comparing mouse and touch screen as pointing devices reported the advantages of touch screen as it is easy to learn and use[16].

But just like the menu-driven systems, direct manipulation systems also give only preordained choices. Another problem is the interpretation of the icons. The designer's intention for a particular icon may be misunderstood by the users. Although it is told that one picture is worth thousand words it is advisable to give the corresponding verbal representation along with the icon[17].

5.3 Further developments in interface design

A new area of research with tremendous potential for interface design is speech recognition. Now, only tutored voices can be recognised by the computer. Human voice differs from person to person. Moreover speech changes with emotion. Accents pose another problem. Continuous flow of speech with words blurring into each other makes computer recognition of ordinary speech a difficult proposition. Yet work is continuing in this direction[18].

5.4 Interface design procedure: a flowchart

Baker has developed a flowchart for interface design (see Figure 5)[19]. According to this flowchart, first the problem which is implemented using the computer is to be identified and analysed. The requirements are to be delineated. Then at the design interface there are three components to be taken care of. They are hardware, software, and user population.

1. Hardware considerations.

The hardware available or which is selected determines the design procedure. The nature of the peripheral devices, like mouse or touch screen

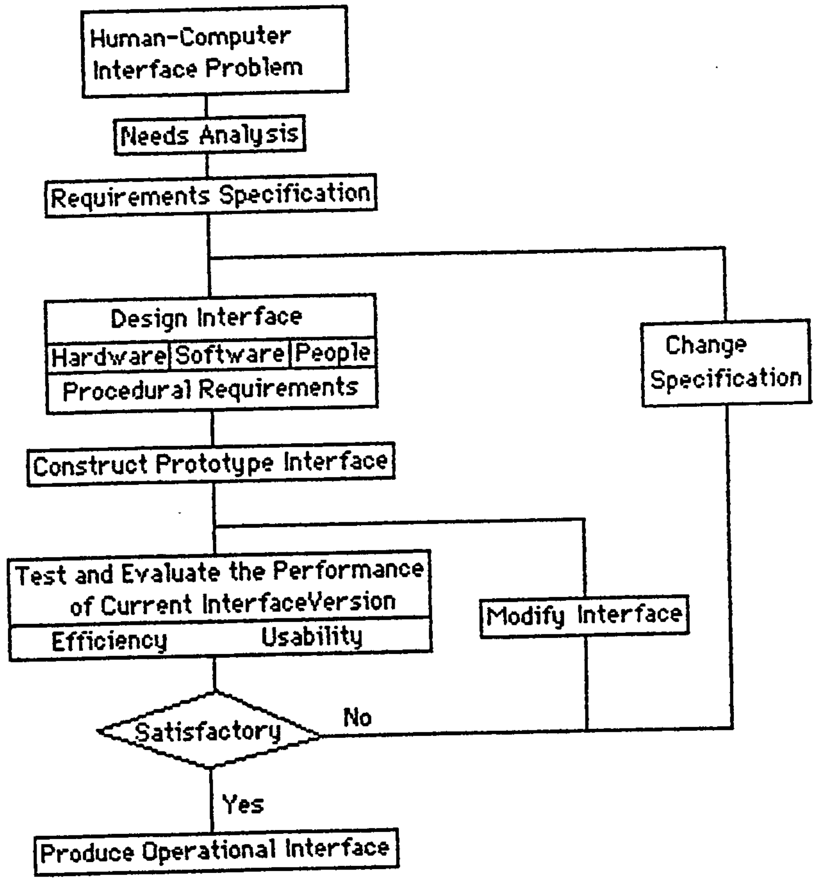

Human-Computer Interface Problem
Needs Analysis
Requirements Specification
Design Interface
Hardware
Software
People
Procedural Requirements
Change Specification
Construct Prototype Interface
Test and Evaluate the Performance of Current InterfaceVersion
Efficiency
Usability
Modify Interface
Satisfactory
No
Yes
Produce Operational Interface

Figure 5 Flowchart for interface design

influences the design. Or it is to be planned, what kind of peripherals are needed for a particular application. Hardware with proprietary operating system also influences the design procedure.

2. Software

The interface needs invariably a software package for its implementation. The selection of the software again depends on the hardware. The software should be able to carry out the design work satisfactorily.

3. Users

Users form an important component of the interface system. It is for the users that the system is designed. So the designer should know the nature of the users. They may be expert or naive, frequent or infrequent users. The success of the design depends on the acceptance of the design by the users. The designer should anticipate the type of training the users need for the use of the product.

The prototype design product should be tested with the users for its efficiency and usability. several methods exist for the evaluation of the interface. These methods are discussed in Chapter 8. Evaluation process is an important aspect of any user interface design.

After the evaluation, the modifications warranted by the testing are implemented and again they are tested. This kind of iterative design process makes a fine design. If the design fails because of the initial specifications which can not be taken care of by the hardware, software or not suitable for the users, the specifications are changed and the design is again done.

5.5 User interface design tools

There are several user interface design tools and specific user interface management systems. Human-computer interface development tools are themselves interactive systems that support production and execution of the human-computer interface. Hix and Schulman[20] evaluated the usability and functionality of three interface development tools, Demo,HyperCard, and Prototyper and recommended HyperCard as the appropriate tool among the three. The selection was dependent on the following criteria:

1. Tool must be capable of producing a large variety of menus and forms.
2. Tool must be easy to use for the types of interfaces it can produce.
3. Tool must mainly use object manipulation for producing interfaces.

HyperCard could satisfy these criteria reasonably well.

REFERENCES

1. **Grudin, Jonathan.** The computer reaches out: the historical continuity of of interface design. *In* : Chew, Jane Carrasco and John Whiteside, eds. *Empowering people: Proceedings of CHI 1990 Conference. Seattle, Washington, April 1-5, 1990,* p. 261.

2. *Ibid.,* p. 264.

3. **Barker, Philip.** *Basic principles of human-computer interface design,* 1989, p. 14.

4. **Yee, Martha M.** System design and cataloging meet the user: user interfaces to online public access catalogs. *Journal of the American Society of Information Science,* 1991, 42(2), p. 79.

5. **Barker, Philip.** ref 3, p. 57.

6. *Ibid.,* p. 57-60.

7. **Shneiderman, Ben.** *Designing the user interface: strategies for effective human-computer interaction.* 1987, p. 275.

8. **Large, Andy.** The user interface to CD-ROM databases. *Journal of Librarianship and Information Science,* 1991, 23(4), 214

9. *Ibid.,*215-216.

10. **Booch, Grady.** *Object oriented design with applications.* Benjamin/Cummings Publishing Company : Red City, California, 1991. p. 36.

11. **Salkind, Neil J.** *The big Mac book,* 1989, p. 38.

12. **King, Alan.** Room with a view: Microsoft Windows and CD-ROM. *Database.,* 14(2), 1991, 100.

13. **Shneiderman,** ref. 7, p. 202.

14. *Ibid.*, pp. 201-202.

15. **Mac Aogain, Eoghan and R. Reilly.** Discourse theory and interface design: the ease of pointing with the mouse. *International Journal of Man-Machine Studies.* 32(5), 1990, 591-602.

16. **Sears, Andrew and Ben Shneiderman.** High precision touch screens: design strategies and comparison with a mouse. *International Journal of Man-Machine Studies.* 1991, 34(4), 593-613.

17. **Kacmar, Charles A. and Jane M. Carey.** Assessing the usability of icons in user interfaces. *Behaviour & Information Technology*, 1991, 10(6), 443-447.

18. **Lange, Holley R.** The voice as computer interface: a look at tomorrow's technologies. *The Electronic Library,* 1991, 9(1), 45.

19. **Barker,** ref 5, p. 20.

20. **Hix, Deborah and Robert S. Schulman.** Human-Computer interface development tools: a methodology for their evaluation. *Communications of the ACM,* 1991, 34(3), 85.

CHAPTER 6 : HYPERCARD FOR INTERFACE DESIGN

6.1 The HyperCard package

6.1.1 The features of HyperCard

HyperCard was launched in 1987 as a proprietary package for the Apple Macintosh computer. Bill Atkinson who was responsible for the software package MacPaint was again the brain behind the team work which brought out this software[1]. HyperCard was given free with every purchase of Macintosh. The present version is 2.0. It runs on the operating system, Macintosh system 6.0.5 or later. It needs 1Mb RAM or 2 Mb RAM when used with the multifinder. HyperCard can take care of information in the form of text, sound, or video. It does not support colour.

From the very beginning HyperCard has been a popular software package. It is an application tool in the sense that one can construct one's own application without much difficulty. Bill Atkinson himself described it as an authoring tool[2]. The two major features of HyperCard are its graphic user interface and hypertext-like links. It has got even relational database management capabilities. Information stored in one stack can be retrieved in other stacks. It goes even further to reveal the full context of the related information in the other stack. Generic links can be established between the stacks. One major advantage of Database Management System (DBMS) packages is that they have got versatile reporting capacity. Now with the version 2.0, HyperCard also has improved its reporting capability. But it is to be borne in mind that HyperCard is optimised for browsing through records in search of desired information.

HyperCard takes on object-oriented programming qualities. Nielsen and others[3] have related the four main features of object-oriented programming with HyperCard. These features are:

1. Objects and encapsulation

An object-oriented programme consists of a number of objects, each consisting of data structures and algorithms to manipulate the object. In HyperCard a large part of the object's specification exists as attributes constructed by direct manipulation and it is not present in textual form. It is easier to construct screen designs by directly placing the objects at the appropriate places.

2. Message sending

Each object understands a certain number of messages and is activated when it receives such a message. System produces such messages as in a mouse click.

3. Polymorphism

The same message may not mean the same thing in different object contexts. Therefore the result of sending a message will depend on the class of the receiving object. In HyperCard, for example, open card message is sent whenever a card is shown or opened. But each card or background will have its own open card handlers specifying the method of dealing with the message.

4. Specialisation and inheritance

Various objects are related in hierarchy. The lower levels inherit the properties of the higher level, but may add new behaviours. For example, the general way to deal with the mouse click is inherited from the higher level. But the lower level handlers can override the higher level messages, if needed. In a higher level, at the stack a 'hide field' message may be there. But at the card level, a 'show field' message may overrule it.

An object is any HyperCard unit which can send and receive messages. Thus objects include buttons, fields, cards, backgrounds, stacks, and HyperCard itself.

6.1.2 The major elements of HyperCard

The major elements are: Stacks, Backgrounds, Cards, Fields, Buttons, the Home Stack, Recent, the Help Stack, the Message Box, and the Scroll Window. The first five of them are the basic building blocks for HyperCard.

1. Stacks

A HyperCard Stack is usually a homogeneous collection of information, which the computer considers as a separate disk file shown as an icon with the name of the stack. It takes the metaphor of a stack of cards. A stack can be opened by double-clicking the icon or by using the option 'Open Stack ...' at the File menu when the HyperCard program is active. With version 2.0, stacks can be opened in separate windows depending on the memory of the Macintosh.

2. Backgrounds

A stack can have more than one background depending on the requirements. All cards belonging to the same background have the same background properties. A background can have shared graphic, shared fields, and shared buttons.

3. Card

Cards inherit the background properties and can have additional card graphics, card fields and card buttons. Card contains textual, graphical information or both.

4. Fields

The textual information which can be accessed is stored in fields. Fields in a card may be common to a background or specific to that card; but the information contained in the fields will be belonging to that card only. Now fields can be barred from checking if needed. Font, size and style of the text in a field can be specified. Also a particular chunk of text can be put in a different font , size and style. A field can contain up to 30,000 characters. Usually a field displays the text which can be accommodated in the designed border of the field. The information is not lost. The field border can be extended to show the rest of the text. Another way is to make it a scrolling field. This is a convenient way to show textual information as appearing in a field meant for abstract in a bibliographic record.

5. Buttons

Buttons are the primary navigational tools. Buttons can be empowered with different powerful functions with proper scripting. Icons can be assigned to buttons. These are the objects usually designed specifically for manipulation of the information. Users can click these buttons for action on the stacks. Creation of buttons is an important part of the interface design.

6. Home Stack

Whenever HyperCard is opened, the first card of the home stack is shown, in which the important stacks can put as buttons. By clicking the button, that particular stack can be opened. Home stack also contains the powerful 'Preferences' card. By the preferences the user level can be set.

Level	**Capabilities**
Browsing	Find information only; no text entry or editing
Typing	Adds text entry and editing on cards
Painting	Adds access to painting tools
Authoring	Adds access to field and button tools
Scripting	Adds access to HyperText scripts

As the level progresses new capabilities are added to the existing capabilities. Thus a user having 'Scripting level' access to HyperCard can only design applications using the programme. An end user of a catalogue designed with HyperCard may be given only 'Browsing' or at the most 'Typing' level access only. Typing level users can enter text in some specified fields which are left unlocked intentionally. The level can be set when the HyperCard is opened and can be changed by going the Home Stack. Other facilities available in the Home stack includes the access to power tools for resource movements as in the case of sound or icon.

7. Recent

It shows the facsimiles of the last 42 unique cards accessed during a transaction. Repeated visits are considered as a single visit only for the purpose of convenience. This is an extension of the 'Back' and 'Forth' options available in the menu 'Go'. The 'Recent' option is also part of the 'Go' menu. One problem with 'Recent' as a navigational aid is that the card representation is so miniatured that similar cards can not be distinguished properly for clicking.

8. Help Stack

Help stack is considered an important part of HyperCard, as contextual help can be requested at any stage.

9. Message Box

Message box can display system messages as well as receive messages from the users. Commands can be negotiated through the message box. Also calculations can be done using it.

10. Scroll Window

The window size can be of any size from 64 x 64 pixels (ie. less than a square inch) to 1280 x 1280 pixels.

Each HyperCard object has its own layer. If it is transparent, the object beneath it can be seen. One should be very careful about the background and the card layers. The properties created for the card layer is specific to that card only. If a property is to be shared by all the cards in a background then it should be applied in the background mode. A second background does not inherit the properties of the first background. If a similar object is to be there on the second background it should be copied and pasted. Every object can have a name, a number which shows its position relational to other similar objects in the same background, and a unique identification number supplied by the system.

6.1.3 Tool palette

The tool palette contains three object tools, namely cursor, button, and field tools. The rest of the tools are meant for painting. Graphics in HyperCard is based on MacPaint. MacPaint can have control over every pixel or dot on the screen. The graphic interface can be used for many purposes in the design process. There are different tools for selection of graphics, painting, drawing lines , making figures like rectangle, curve, etc.

6.2 HyperTalk: Programming language for HyperCard

The programming in HyperCard differs very much from the traditional way. In the usual programming by Pascal, Basic, C, etc. the programming is done from the beginning to the end. In HyperCard, programming is done to control the different objects. The programming language is proprietary for HyperCard and it is called HyperTalk. The programming itself is called 'scripting'. Winkler and Kamins have defined 'script' as a collection of statements associated with a particular object[4]. Within script statements are grouped into handlers. An object executes the statements within one of its handlers when it receives an instruction to do so. That instruction is called a message. HyperTalk is English language-like. So the syntax is user friendly. Even the terminology is very flexible. For example, the name of the object 'Button' can be written 'btn',or 'button'. The flexibility makes the programming very interesting.

Scripting is done for the objects in a screen provided for each object. It is always related to handlers which act upon the object. A usual handler is 'mouseUp' which means that mouse is clicked and released on an object. So the handler is 'on mouseUp' and the programming ends with 'end mouseUp'. The capitalisation of the second word is only for the convenience of reading since the two words continuously written; otherwise it can be written in all capitals or all lower case. A handler in a script is shown below:

```
on mouseUp
  go to card "ABC"
end mouseUp
```

The indentation is set by the 'tab'. If the scripting is all right, then the indentations will be at the right place showing a readable sequence. This particular script for a button can be executed by clicking the button. The card named 'ABC' will be shown. If there is no card named 'ABC', then a message is shown: 'There is no such card'. The 'OK' in the message is to be clicked as the acknowledgement for the receipt of the message.

There is a hierarchy for the message path. So the handlers are to be added at the right object in the hierarchy. One above in the hierarchy is right; but if the handler is added at a lower level in the hierarchy, it will not work. For example, if the mouse with the pointer over a button is clicked, the 'mouseUp' message passes to:

1. that button
2. the card it is defined on
3. the background of that card
4. the stack of that background
5. any stack with stacksInUse property
6. the home stack
7. the HyperCard itself

This is a static message path. If the message is dynamic in the sense that it is to go to some other card in some other stack, then the message for a mouse click on a button will pass to:

1. that button
2. the card it is defined on
3. the background of that card
4. the stack of that background
5. the current card (ie., the destination card)
6. the background of that current card
7. the stack of the current card
8. stacks in the stacksInUse property
9. the home stack
10. HyperCard itself.

The message is executed whenever the message is intercepted by a corresponding handler at any of these points.

Boolean logic of true or false is used in the if-then-else nesting. Another interesting function is the 'repeat' command which can be used for a variety of useful loopings.

With the 2.0 version, several new functions and commands are added. Some important commands among them are 'mark cards', 'unmark cards', and 'sort lines' of a container ie., a field or a variable. Commands and functions written in C, ThinkPascal, ThinkC can be incorporated into the system as external commands (XCMDs) and external functions (XFCNs). XCMDs and XFCNs which come with stacks can be copied and used for the creation of non-commercial stacks.

6.3 HyperCard stack design

HyperCard stack design considerations are actually part of the interface design. There are certain aspects peculiar to the HyperCard package which should be taken into account before embarking upon the design. The card metaphor is to be properly understood to make the design functional. Each card can be a record. These records can be arranged into different stacks for convenience. Or these cards can be put into different backgrounds in the same stack. It is immaterial where the cards are available whether they are in the same stack with different backgrounds or in entirely different stacks, HyperCard can establish the links. It is the designer's choice and convenience to create more than one stack for the interface.

6.3.1 Stack structures

The nature, purpose, and display specifications determine the structure of the stack. This structure is actually the way in which the user sees the stack. Actual physical order of the cards in a stack is not the concern of the user. The linking made by the designer makes the structure of the stack. It is possible to structure the stacks in five ways[5]:

1. Linear structure

A linear structure forces the user to pursue the cards in a single logical path as in a straight line. Only forward and backward movements are possible. (Figure 6)

2. Tree structure

A tree structure allows the user to choose several branches to follow the path as in an organisational chart. Navigation in a tree-structured stack lets users to move forward and backward within a branch and return to the most recent branching point. (Figure 7)

3. Network structure

Network structure is not in any hierarchic order. It allows to explore the cards in many different ways. Navigation in network stacks depends on the subject matter. Heavily used cards can form as reference points. Or stack map or menu can help in navigation. Network stack can be designed so that the user can move forward to a new card, return to a card most recently visited, return to a specified card or jump to the stack map or menu. (Figure 8)

4. Single-frame structures

In this stack structure, the user gets the impression that all actions take place in the same card. There is no sense of travelling. But actually there are several cards. It is done by the careful designing of the background and linking buttons.

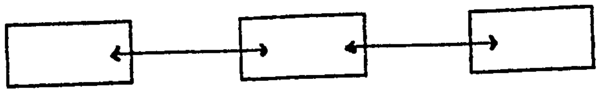


Figure 6 Linear stack structure

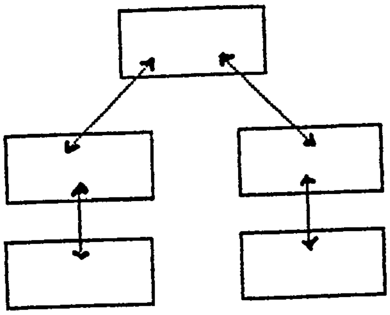


Figure 7 Tree type stack structure

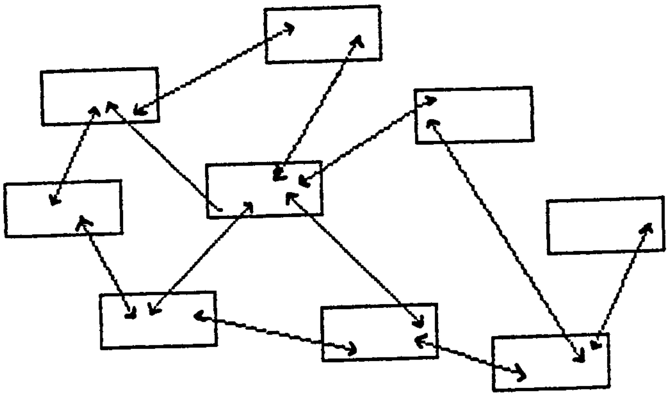


Figure 8   Network type stack structure

5. Combination structures

A combination stack structure uses two or more types of structures mentioned above. There should be considerable contemplation before the selection of the structure of the stack because it is the basis of the performance of the interface.

REFERENCES

1. **Goodman, Danny.** *HyperCard 2.0 handbook,* 1990, p. 853.

2. *Ibid.,* p. 851.

3. **Nielsen, Jakob, I. Frehr** and **H.O. Nymand.** The learnability of HyperCard as an object-oriented programming. *Behaviour and Information Technology,* 1991, 10(2), 111-120.

4. **Winkler, Dan and Scot Kamins.** *HyperTalk 2.0: the book,* 1990, p. 12.

5. **Apple Computer.** *HyperCard stack design guidelines,* 1989, p. 24-33.

CHAPTER 7 : IMPLEMENTATION OF THE NOVEL INTERFACE DESIGN

7.1 The novel interface requirements

The Objectives for the novel interface for OPAC are described in detail in Chapter 4. The objectives can be put succinctly as follows: the novel interface should facilitate browsing with proper navigation through the citation network. It also should give opportunity for a cocitation search. For this prototype design, the subject area of 'Microelectronics' is chosen. In that, a subset area of 'Multichip modules' has been selected for input of documents.

The hardware for this project consists of an Apple Macintosh LC computer with 4Mb RAM, 40 Mb hard disk and a floppy disk drive. The input devices consist of a keyboard and a mouse.

As for the software there has been a choice between FileMaker Pro and HyperCard both proprietary on Macintosh. FileMaker Pro is a database management system with a graphical user interface having colour facility. But keeping in view the requirement of designing a browsing environment just a DBMS package would not be sufficient for the novel interface. So the natural choice has been HyperCard the credentials of which have been proved by a study by Hix and Schuman[1] as highlighted early in Chapter 5.

It is presumed that the users would be researchers and students with some or no computer experience.

7.2 The stack design for novel interface

7.2.1 The database construction

The database or 'stack' in HyperCard terminology had to be created taking in view the links to be provided. The documents for the stack were from the journal, *IEEE Transactions on Hybrids, Components, and Manufacturing Technology.* Articles pertaining to 'Multichip Modules' were selected for inclusion in this database. The references given in the selected articles were entered as separate records or cards. Thus various sources,such as articles from other journals, conference papers, and monographs all got entry into the database. The database contains 219 unique records. At the beginning stage, a small number of fields were created to enter information on Author/Editor, Title, Bibliographic details, and Abstract. A field for Record identification number 'Id' was also created. But as the database is created, the structure had to be decided since the structure is part of the totality of the interface design.

7.2.2 A relational approach for linking the citation network

The references and citations for a document should be browsable. But one-to-one linking will not give such a browsing environment. So alternative linking methods have to be thought of. One approach is to use a relational linking (Figure 9). Document A is connected to documents P, Q and R by having common attributes. D1, D2, D3 can be regarded as unique 'Id' numbers which are entered in a field in the document record A. All the unique 'Id' numbers of the references are to be entered in a field in the card for the document in question. These unique numbers can be used for identifying the references for a document.

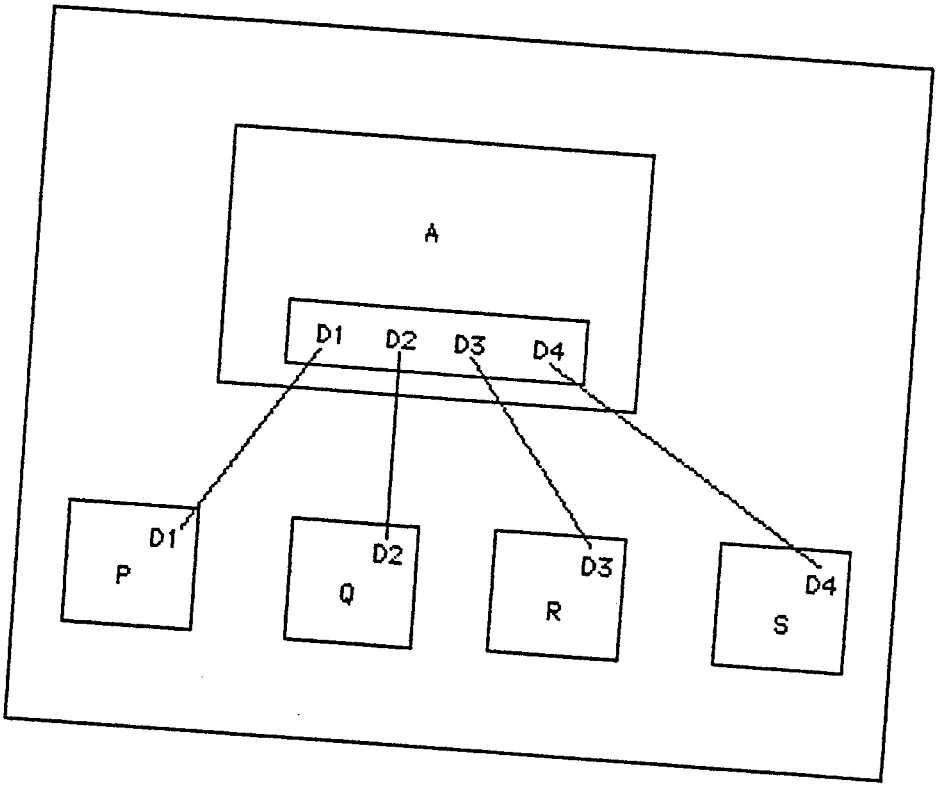


Figure 9 Relational linking of records

Once identified, these can be presented for display. Thus the plan for linking was done with the unique 'id' numbers given to documents. Figure shows the field of references containing the 'id' numbers. This field is always hidden from the view of the users.

Usually it is a common practice in libraries to give unique 'document identification numbers' to documents in the form of accession numbers or barcode numbers.. When a new document is entered, it gets a unique 'Id'. As for the references in a document, some may already be in the system. Duplicate entries can be identified and their 'Id' numbers are noted in the field meant for entering the references. Such references which are not in the system can be entered as separate records with new 'id' numbers and these 'id' numbers are entered like-wise in 'Ref', the field for references. Thus all the references are noted with their 'id' numbers in the 'Ref' field. New documents get added to the system. A search may retrieve materials which are not in the library also and the library can procure such materials on interlibrary loan.

In the early stage a background button with the name 'References' was added. The scripting of this button checks each 'Id' in the references field and marks the corresponding card by finding it in the 'Id' field of the documents. All the marked cards are then sorted. Thus when the button "References" is clicked, all these activities in the scripting are done and the first marked card is presented. Two buttons added to the background, Previous and Next shown by icons of backward and forward arrows do further navigation of going back and forth among these references.

But later it was thought that it is better to give the number of the references at the display itself. A field was created for this. This would have lead to a repetition of the word "references" one, on the button and other in the field where it announced the number of references. So instead of the button, a background field with the number of references is now made to act as a button, because of its locked text and the script added to it.

Citations are also similarly shown as a locked field with the number of citations for the document. If there is no citation the field is hidden. On clicking the button-like citation field, the 'id' is checked in every card's References field and wherever it occurs the card is marked. All the marked cards are checked and a list is made which is presented on a separate card. Any document on this card can be clicked to go to that particular document card. There again one can have browsing facility to see the next citation, if there is one. Again notice is given to guide on which card one is. Each visit to a card is counted and the information is displayed.

7.3 Second stage of the design

7.3.1 Addition of a front end stack

At this stage it was decided that a second stack should be created for the display and manipulation of the instantly created lists of the citations and references. This stack was designed as a front end to the first stack. So the initial screen and the menu screen were added as cards to this stack. This stack was called 'CiteFront' while the main stack was called 'CiteBase'. The Stack, CiteFront is actually a front end to the database to give the cards for the presentation of lists as navigational aids. The main menu is also on this stack. By this arrangement the main stack is not disturbed. The number of cards in the main stack corresponds to the number of records in the database. All these cards are in the same background. The user does not know about the number of stacks involved in the transactions. It is a part of the design strategy to decide the number of stacks to be made for the interface construction. The interface itself is for all purposes a single entity.

7.3.2 The opening screens for the interface

A graphical opening screen is made as an attractive facade to the system (Figure 10). A door is pushed to open the system. Then options appear to take the direction of 'Microelectronics packaging' or 'Microelectronics materials' (Figure 11). 'Microelectronics materials' has no documents and a note appears to that effect if the button is clicked. 'Microelectronics packaging' on the other hand leads to a menu screen.

7.3.3 Menu card facilities (Figure 12)

Authors/Editors

This button leads to a list of authors and editors. The number of documents by each author/ editor is also given. The lines in this list can be clicked to get the chosen person's documents. This uses the clickLine facility of HyperCard on a locked field to reach the documents.

Highly cited documents

This button relates to a list of highly cited documents. The number of citations also are given with each document. A click at any

Figure 10 Opening screen I

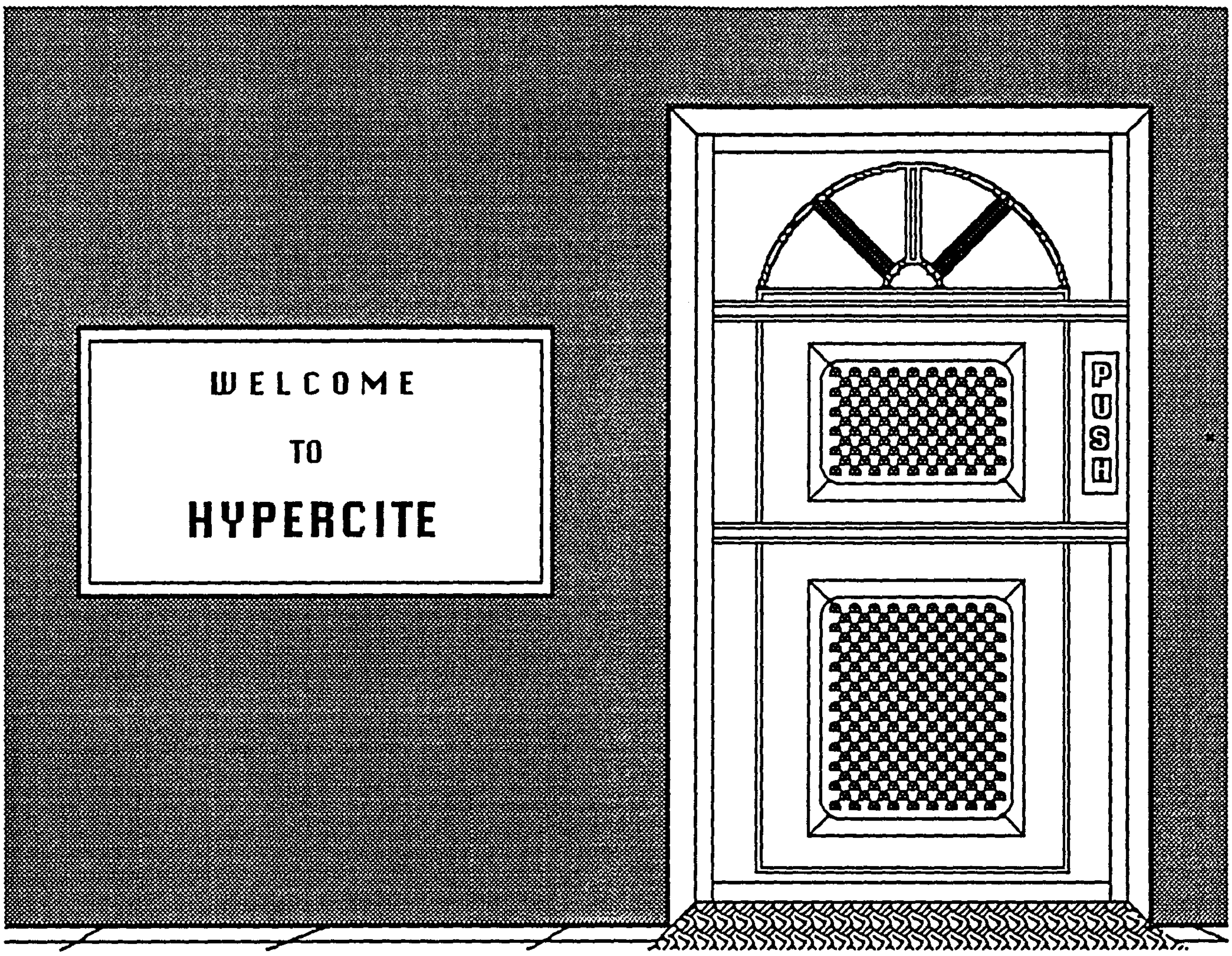


Figure 11 Opening screen II

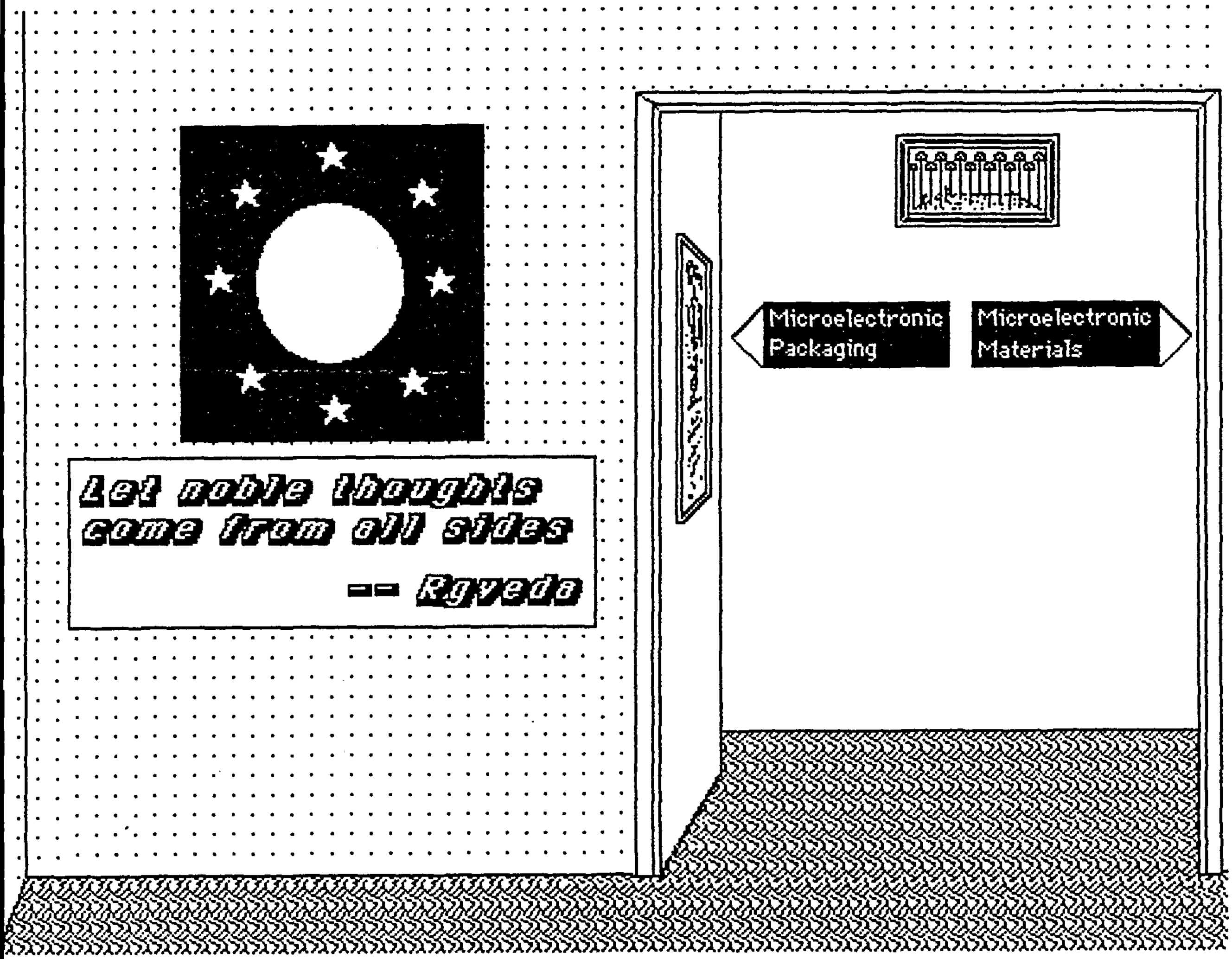
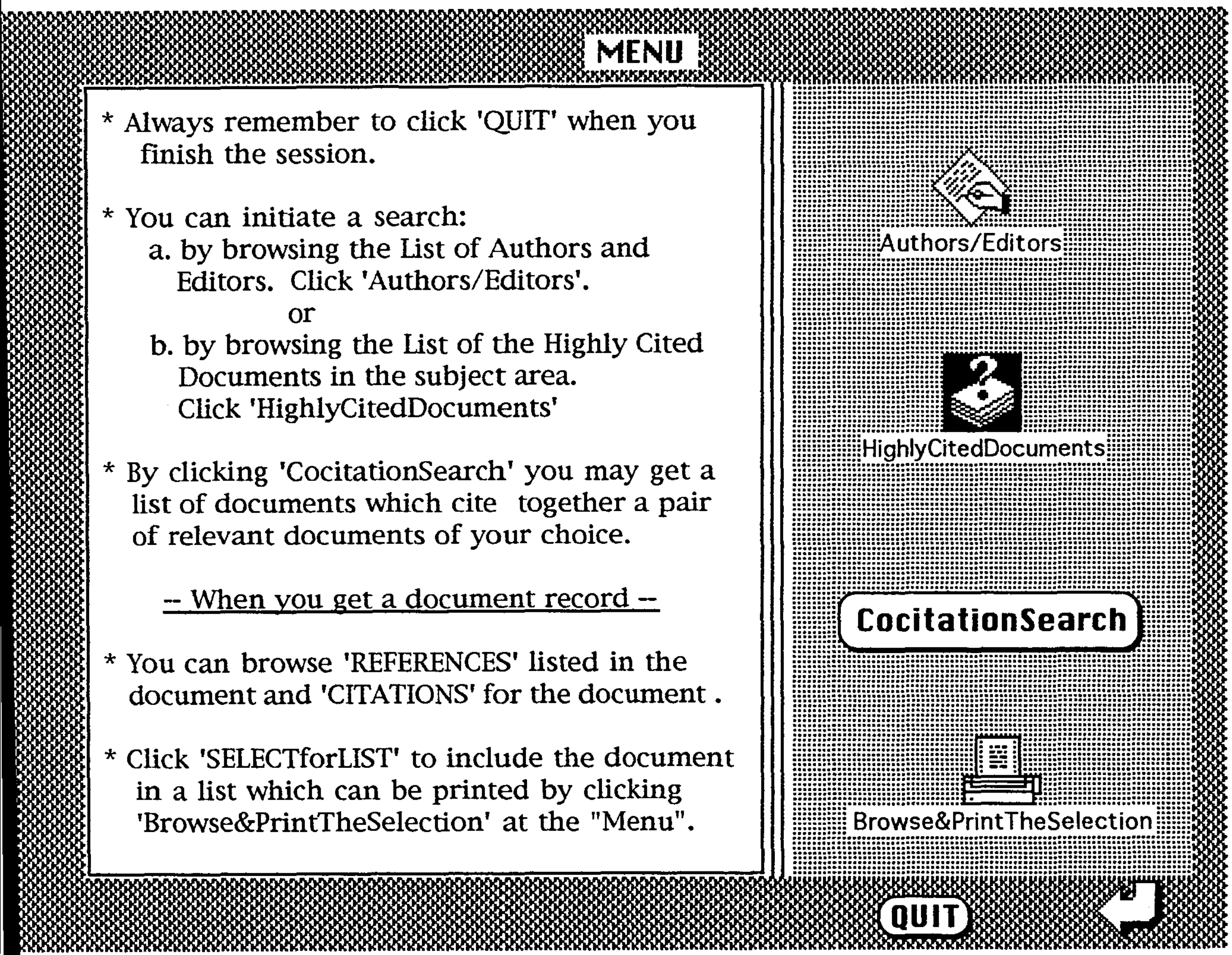

Let noble thoughts come from all sides
-- Rgveda
Microelectronic Packaging
Microelectronic Materials

Figure 12 Menu screen



* Always remember to click 'QUIT' when you finish the session.

* You can initiate a search:

a. by browsing the List of Authors and Editors. Click 'Authors/Editors'.

or

b. by browsing the List of the Highly Cited Documents in the subject area. Click 'HighlyCitedDocuments'

* By clicking 'CocitationSearch' you may get a list of documents which cite together a pair of relevant documents of your choice.

-- When you get a document record --

* You can browse 'REFERENCES' listed in the document and 'CITATIONS' for the document .

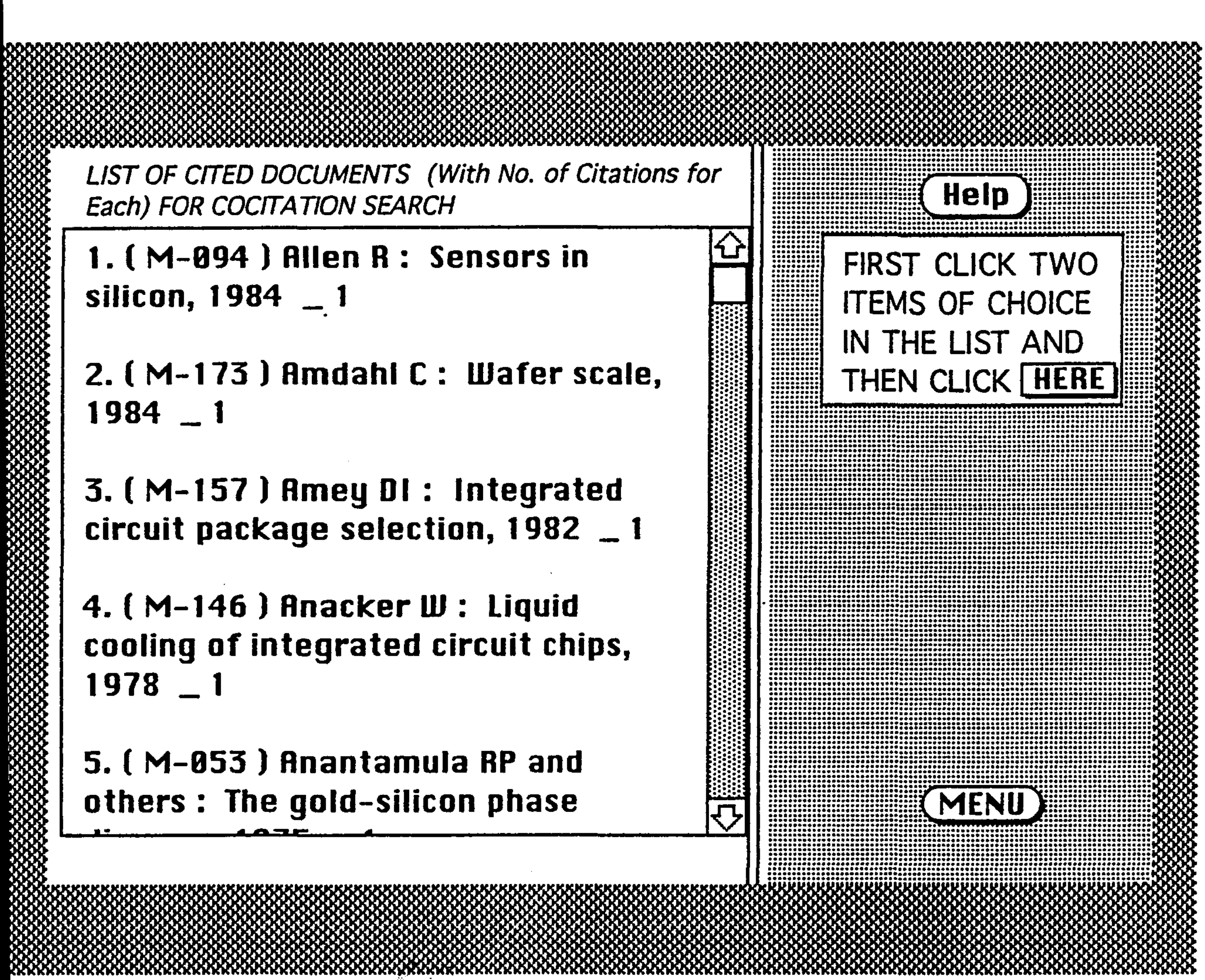
* Click 'SELECTforLIST' to include the document in a list which can be printed by clicking 'Browse&PrintTheSelection' at the "Menu".

chosen document will lead to the citations of that document.

Cocitation search

This is a very important byproduct of the citation network. Since the list of cited documents can be prepared and presented, it is quite natural that there should be facility for searching as to whether two such chosen documents are cited together in any of the documents. For this, a list of cited documents with the respective numbers of citations for each is created (Figure 13). On each update of the system this can be automatically updated. The field giving the document list is locked and is amenable for search by clicking the documents. There is a notice that two documents can be clicked. The clicked documents are remembered by the system and when instructed, a 'Here' button in the notice field is clicked, a further notice appears confirming the selection giving the entry name of the first author of the first document and the entry name of the first author of the second document. One can confirm it by clicking 'Ok' or abandon the search by clicking 'Cancel'. If it is 'Ok' then the search proceeds to show the results. If there are no hits, a notice comes to that effect. If there are hits for the cocitation search comprising the two specified authors, a list of the resulting documents appear. This list can be used for further navigation to see the individual documents in detail. This again consists of marking the documents where the 'Id's of the two documents in question are checked in the references field of every card. If the two 'Id's occur together the card is marked. Again all the marked cards are arranged in descending order of the year of publication and the list is prepared and presented with the recent document coming first in the list.

Figure 13 Cocitation search screen



LIST OF CITED DOCUMENTS (With No. of Citations for Each) FOR COCITATION SEARCH

1. (M-094) Allen R : Sensors in silicon, 1984 _ 1

2. (M-173) Amdahl C : Wafer scale, 1984 _ 1

3. (M-157) Amey DI : Integrated circuit package selection, 1982 _ 1

4. (M-146) Anacker W : Liquid cooling of integrated circuit chips, 1978 _ 1

5. (M-053) Anantamula RP and others : The gold-silicon phase

FIRST CLICK TWO ITEMS OF CHOICE IN THE LIST AND THEN CLICK HERE

Cocitation search can be done for more than two documents also. But it is not encouraged, since in that case the number of hits will be very minimal because of the size of the database. As the number of documents in the cocitation search increases the relevance of the documents retrieved actually increases.

Browsing and printing of the selected list

The browsing of the catalogue through its various linkings actually results in the selection of documents as it proceeds. These 'picked' documents are put in a list to see all documents selected in that session (Figure 14). One can remove a document from this list by clicking the document and going to the document and 'deselecting' it. Or the cards selected can be browsed again. If the list is all right then it can be printed. Although options for print formats can be given, only one print format option was provided at this stage.

7.4 **A normal card screen** (Figure 15)

In a normal card screen the following elements can be seen:

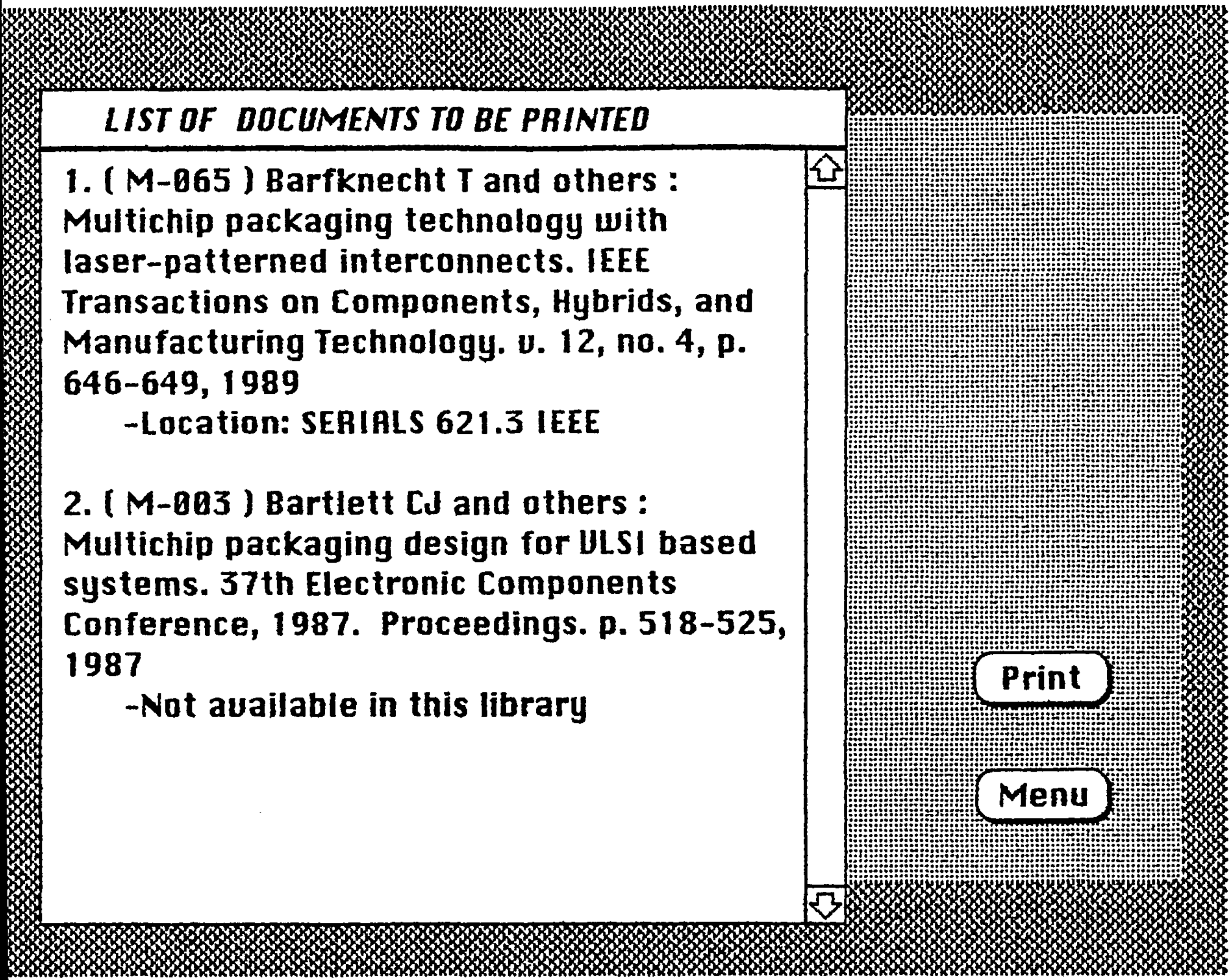
Author/editor field

This field contains the names of the authors and editors associated with the document. It is given as a scrolling field. If a number of authors extending the border of the field are there, scrolling becomes active. All the authors can be seen by scrolling the field.

Title field

The title field is again a scrolling field, since there can be long

Figure 14 Browsing and printing screen

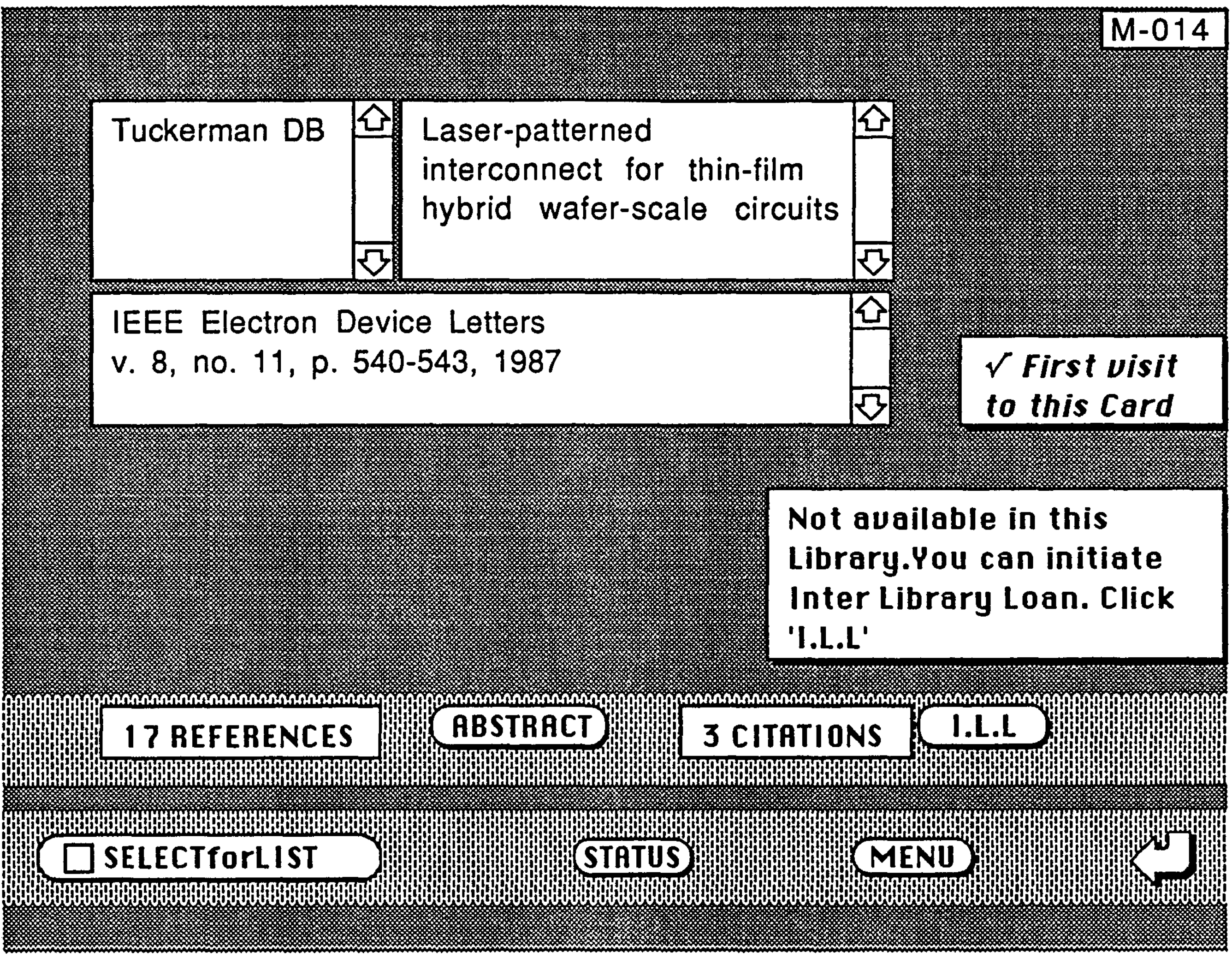


LIST OF DOCUMENTS TO BE PRINTED

1. (M-065) Barfknecht T and others : Multichip packaging technology with laser-patterned interconnects. IEEE Transactions on Components, Hybrids, and Manufacturing Technology. v. 12, no. 4, p. 646-649, 1989
   -Location: SERIALS 621.3 IEEE

2. (M-003) Bartlett CJ and others : Multichip packaging design for VLSI based systems. 37th Electronic Components Conference, 1987. Proceedings. p. 518-525, 1987
   -Not available in this library

Figure 15 A normal card screen



Tuckerman DB

Laser-patterned interconnect for thin-film hybrid wafer-scale circuits

IEEE Electron Device Letters
v. 8, no. 11, p. 540-543, 1987

√ *First visit to this Card*

Not available in this Library.You can initiate Inter Library Loan. Click 'I.L.L' titles which may extend beyond the present border of the field.

Bibliographic details field

Here all the other bibliographic details of the document are given. There is no difference given for journal articles or monographs. A field for giving the document type, like journal article, monograph, or conference paper can be maintained. For the present project such minute details were not considered. For the user, from the display it must become clear as to whether the document is monograph or journal article. Field labels also were considered unnecessary. The screen design for the document explains the nature of fields.

Number of references and Number of citations

The field for the number of references in the document is given as an object for manipulation. It acts as a button. Its position and looks suggest the nature of a button. The user's attention is drawn to it by the bold lettering. The field for the number of citations shown on the right hand side also acts as a button. It is similar to the 'Number of references' object. If the user clicks one, then he/she knows the use of the other button also. These two clickable objects are kept in a separate layer to show the difference.

Abstract

At the centre of the layer is the button 'Abstract'. A click on this button shows the abstract field. Usually it is kept hidden from the user's view. It is given only when it is demanded. The abstract appears as a scrolling field. A mouse click will hide the abstract again. This saves the screen from being cluttered. If there is no abstract entered with the document, a notice to that effect appears.

The Next and Previous buttons

The Next and Previous buttons are basic navigational tools. These appear on the screen only if there are two or more cards to be shown as a result of an action. These are given as forward and backward arrows. At the outset, if it is the first card in a bunch of cards to be shown only the Next button appears. At the last card Next button disappears and only the Previous button remains to show that the end point has reached. If more than two cards are there, both the button remains for the middle cards so that one can go forward and backward in a chain.

Back button

In the next layer, all the objects shown are buttons. 'Back' button is again a powerful navigational tool, to traverse back to the original transaction area. For example if a citation search is being done, one can back at any time to the list from which the citation record is selected. The 'Back' button facility is available at every stage in all screens. It is given as an icon implying 'going back' as different from the straight 'Previous" button.

Menu button

Menu button is also a navigational aid. At any stage in browsing one can go back to the menu to start a new search. This will help the user to come out from lengthy browsing. Menu button helps in quitting the session also.

Status button

This button has got the useful function of showing the presence and availability of the document. If the document is not present a notice appears to indicate it. And a button for interlibrary loan initiation appears. If the document is on loan a notice to that effect and a button for reserving the item appears.

Button for the selection of documents

'SELECTforLIST' is a button for selecting the document as perceiving it useful in the course of the browsing. When selected it indicates the selection by changing its name to 'SELECTEDforLIST'.

Where-you-are indication field

This field appears in most of the cases when there are two ore more cards are to be shown as a set. For example if the citations for a document is shown, a note field appears saying 'This is number 1 of the 4 citations for Martin, B, 1989'. Notices like this are shown to indicate the current position. This again is a navigational aid.

Number of visits field

This field is a useful reminding device. From the first visit to a card itself, the visits are logged and shown.

These are the objects usually shown on the screen. Whenever an object is not needed or not usable, it is hidden to avoid any confusion. If there are no citations for a document, a citations object is just not necessary. If it is there, users will expect that there are citations and will try to click it. Thus fields and buttons are made to vanish when they are not functional.

7.5 Scripting for the novel interface design

Most of the objects needed HyperTalk scripting in this interface. Straight HyperCard linking available with the buttons was never used because the interface was based on a network approach. The nodes of this network are prepared as and when they are needed by the system by the scripting involved. The structure of the database also was designed to effect this network on a relational model.

The stack, 'CiteBase' has extensive scripting attached to it at the stack level. Mostly it involves the 'openCard' handler. So many fields and buttons are to be hidden when the card is opened. At the same time on the basis of some condition, like 'if...then...else' some fields and buttons should appear on the screen. Appendix 1 gives the script for the main stack 'CiteBase'.

Simple scripts operate with buttons, 'Authors/Editors' and 'HighlyCitedDocuments'. But the script with the list presented for manipulation is worth noticing (Appendix 2).

Programming for the cocitation search involved the checking of two variables and also involved some confirmations from the user. Appendix 3 gives the script which negotiates a cocitation search. Browse and print also involved some scripting (Appendix 4).

The scripting for the Status button uses the hidden fields of call number information and loan status. It is the only time when the user has to use the keyboard to identify himself by giving a User Id, when interlibrary loan or reservation is to be executed.

Appendix 5 shows the scripting for 'Status' button. If one enters a User Id a second time for reserving the same book or in requesting interlibrary loan for the same book, it is identified and gives a message that this has already been input.

REFERENCE

1. **Hix, Deborah and Robert S. Schulman.** Human-computer interface development tools: a methodology for their evaluation. *Communications of the ACM,* 1991, 34(3), 85.

CHAPTER 8 : EVALUATION OF THE INTERFACE DESIGN

8.1 Evaluation of interfaces: aims and techniques

8.1.1 Aims of evaluation of interface design

The aim of an interface design is always to improve the usability as far as possible. So the usability of the design is to be checked and evaluated with the help of prospective users. According to Ravden and Johnson[1], an evaluation is likely to reveal :

- the major problem areas
- the particular area requiring improvement
- particularly good aspects of the interface
- how successful the interface is in meeting the tasks

Also it may reveal the misunderstandings, misinterpretations and confusions about the system

First the users as evaluators are to be identified and requested to participate. Some background information on them is necessary. There is no need of a formal questionnaire for this. An informal interview will suffice for gathering such information like their previous computer experience, educational qualifications, etc. The different backgrounds, expertise and experience of the evaluators can give valuable inferences at the stage of the analysis of the evaluation. It is important that this information is gathered before the evaluation.

8.1.2 Methods of evaluation

There can be four ways to evaluate a user interface: formally by some analysis technique, automatically by a computerised procedure, empirically by experiments with users, and heuristically by simply looking at the interface and passing judgment according to one's own opinion[2]. Formal analysis models are still under experimentation and they are not much applied for extensive evaluation. Automatic evaluation methods such as the transaction log analysis are also used in evaluation. But such methods are considered very primitive since through those methods the actual user reactions can not be ascertained. Empirical evaluation is the most used method and can give rise to a thorough testing of the system. Task analysis is the most commonly employed method of empirical testing. Heuristic method is simple and is useful at the early stages of the design. Apart from all these evaluation methods, in an ideal situation it is the relevance of the retrieved records that matters most. This factor is a most subjective aspect. O'Brien[3] is of the opinion that relevance also should form part of the evaluation process. But till now there are no such analytical methods capable enough to elicit a useful understanding of end user's information seeking behaviour needs, to know the success rate at an OPAC.

8.2 Task analysis method for the evaluation of the novel interface

8.2.1 Task analysis method: definition and techniques

Task analysis is defined as a formal methodology, derived from system analysis, which describes and analyses the performance demands made on the human elements of a system[4]. The tasks which represent the functions of the interface are to be identified.

These tasks must be realistic and the majority of the users should be able to carry out the tasks successfully. These tasks are listed and given to the evaluators for performance.

In 'thinking aloud' method the evaluators are asked to give a running commentary of the procedures they are adopting to carry out the tasks. If the task is problematic, that also is is reported. This commentary is recorded by the designer who conducts the evaluation or a taperecorder is used. Later this commentary is analysed to find out the usability problems.

In normal task analysis method the user is asked to perform the tasks and then after a stipulated time the evaluator is interviewed to find out the difficulties, if any encountered during the performance of the tasks.

Virzi[5] has found out that a designer who chooses to run an evaluation with a very few subjects will be able to identify most of the major usability problems and some portion of the less important problems. He suggests that 80 percent of the problems can be identified after only five subjects.

8.2.2 Task analysis method: procedural considerations

At the outset, the evaluators are to be given a brief overview about the system and its objectives. And also they should be made aware of the aim of the evaluation. The evaluators should be convinced about how their opinions are valuable for the design. Then the evaluation procedure itself is to be explained fully to the evaluators. They should not get the feeling that by evaluation they are being tested. They should feel free to report all the difficulties experienced at the system. If some tasks can not be done it should be considered as the problem of the interface. In short, the evaluators should feel at ease at the evaluation process.

The tasks must be clear. There should not be any ambiguity in the terminology used to describe the tasks. Technical terms should be properly explained before the commencement of the task performance.

After the task performance the designer should again question the evaluator to find out the difficulties on each task performance. General comments on the usability, and other aspects should be elicited.

When the system is amended as a result of the evaluation, the amendments done to satisfy some evaluators should not clash with the requirements of some others. Only general problems should be looked into. Some specific problems may be peculiar to only certain users.

8.3 Evaluation of the novel interface

8.3.1 Task analysis procedure for the evaluation of novel interface

It has been decided that a task analysis method would be suitable for the evaluation of the novel interface. A list of tasks was designed with seven tasks relating to the performance of the interface and a corresponding interview schedule was prepared (Appendix 6). It was tested with five post-graduate students from the Department of Library and Information Studies and with two research students from the Department of Electronics and Electrical Engineering.

Each evaluator was given 10 minutes for the execution of the tasks. Further 15 minutes were taken for the interviewing after the tasks were completed.

8.3.2 Analysis of the evaluation process

Task completion

Of the total seven respondents, four could complete the tasks well in time. The other three evaluators were very slow. These three subjects were not much experienced in the manipulation of mouse and Macintosh screen specialities like scrolling.

Task performance in stipulated time

Tasks	No. of evaluators completing the task
1. Give the volume number of the journal of the most highly cited document	7
2. Which is the most recent article citing the most highly cited document? (Give the name of the first author only)	7
3. How many articles are there by Hagge, J.K.?	7
4. How many items cocite the two documents: Spielberger and others (1984) and Johnson R.W. and others (1986)?	4
5. How many items cocite Hagge, J.K. (1988) and Taylor, G.F. and others (1985)?	4
6. Is the document by Hopper and others (1992) available in the library?	7
7. Select the 3rd reference in Hopper and others (1992) and print	7

Only one evaluator could not do the second task. That evaluator reported some confusion in understanding the term 'citing'. Although 'cocitation search' was done by all the four respondents who completed the tasks it was found that they just followed the instructions on the screen only. The concept of cocitation search was not clear. It had to be explained to them again after the task completion.

8.3.3 Suggestions and comments

The general appearance of the interface was commended favourably by all. One major suggestion was about the cocitation search screen. At the time of the evaluation there was no provision for distinguishing the selected documents when chosen in the documents list presented for cocitation. Some sort of marking for the selected document was suggested. Another suggestion has been for the provision of abstracts also when the list is printed. Those who had knowledge of the subject field could appreciate the relevance of cocitation search. This was the only difference between the evaluators with subject expertise and other evaluators. Although browsing was done, the significance of the 'Back' button was not easily understood. Everybody could understand the 'Next' and 'Previous' buttons. It was suggested that the names of these buttons also should be given for proper understanding.

8.3.4 The analysis of the evaluation: inferences and observations

Experience in manipulating mouse is a prerequisite for using this interface. The three evaluators who could not do the cocitation search were not experienced in the manipulation of mouse and scrolling fields. It was observed that those who were novices in using Macintosh improved their performance second time when the process of scrolling was explained. The concept of cocitation search is new to all the evaluators. But its value was acknowledged. Browsing environment was appreciated by the two subject specialists.

The meanings of the terminologies 'references' and 'citations' should be consistently understood separately, so that there should not be any confusion in this kind of search. "References' should stand for the items cited by a document and citation should mean the documents citing a particular document.

REFERENCES

1. **Ravden, Susannah and Graham Johnson.** *Evaluating usability of human-computer interfaces: A practical method,* 1989, p. 96.

2. **Nielsen, Jakob and Rolf Molich.** Heuristic evaluation of user interfaces. *In: Empowering people: Proceedings of CHI 1990, Seattle, Washington, April 1-5, 1990,* New York: Association for Computing Machinery, 1990, p. 249.

3. **O'Brien, Ann.** Relevance as an aid to evaluation in OPACs. *Journal of Information Science,* 1990, 16(4), 265-271.

4. **Ravden,** ref. 1, p. 78-79.

5. **Virzi, Robert A.** Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors,* 1992, 34(4), 467.

CHAPTER 9 :CONCLUSIONS AND RECOMMENDATIONS

9.1 Conclusions

Subject access by Query method using the matching of words and phrases at the OPAC interface is always a difficult task because of various reasons. Prominent among them being the difficulty in describing information needs in the vocabulary of the system. It is aggravated by the explicit use of the Boolean logic. OPAC users are not trained as the expert search intermediaries of online Abstracting and Indexing (A&I) databases. Now work is going on several areas to make the user and the system meet at the interface for meaningful dialogue. Information retrieval research solutions and artificial intelligence and expert system based tactics are being employed to circumvent the difficulties at query mode of search. But a major trend nowadays has emerged to explore the potentials of browsing for subject access. Interdisciplinary research approach and the inference that browsing fosters creativity have increased the interest in browsing as a search strategy. Another development in information technology also enhanced the interest in browsing. That is the emergence of hypertext systems. Browsing systems have become a possibility.

Direct manipulation systems with Graphical user interface have changed the scene of user interface design. An application designer tool like HyperCard can now make user friendly systems. Interface design is part of the solution for any system development. All the links and structures which help in making the system usable should be made in the planning stage itself. The interface should be designed as a total system. Front end alone can not save the situation, if the internal structures of the database are not coherent. The interface should be functional and aesthetically pleasing. Since the focus is now on the users, any system should be flexible and adaptable. The rigidity of the commands and the consequent error messages which have vexed the users are no longer tolerated. Graphical user interfaces are now helping a wide variety of users.

The versions of HyperCard 2.0 and its programming language HyperTalk can take care of a relational approach. Thus transient links can be easily made between nodes which can be browsed. The links remain only for the session. Thus the browsing and relational approaches in HyperCard can be meaningfully used for the creation of a citation network with proper scripting with HyperTalk. Of course, the database or stack design should take care of this linking by creating a separate field for the inputting of all the unique identifiers for all the references in a document. At the time of data entry all these reference numbers are to be entered and should make sure that the cited documents are also available in the system. Even if a document cited thus is not present in the library, it should be entered as a separate document entry only to say later that the document is not available in the library. This document can be procured on interlibrary loan. With the down loading facilities from other databases this need not be a difficult proposition. The workload of the library personnel will increase and this is a major deterrent for the creation of such a network. Another major problem is disk space since articles and other documents which are not in the library also are to be included. It is presumed that as the computer technology progresses the problem of storage will be solved.

9.2 Recommendations

1. Browsing and picking useful material is a usual form of search strategy and such systems are to be made available to the users of OPACs. The final list which one makes after this search strategy should be again amenable for browsing and checking.

2. Cocitation search is to be popularised as it gives relevant materials. This can be done only if systems with such a facility is provided in OPACs. When it is easy to click two documents and ask for the documents which cite these together, users will start using this search strategy. Researches done in the area of cocitation searching should not rest as just academic exercises.

3. A system with a citation network approach is recommended for small libraries and information systems catering to very small areas of research.

4. A system like the present one, which facilitates browsing and access through 'highly cited documents' and 'authors/editors' in the subject area can form a subsystem of the OPAC for a big library. The linking procedure for such a subsystem is to be further explored. Through a relational approach it is shown that it is possible to make transient citation network links through HyperCard. In a big library a list of highly cited documents will offer documents from widely different subject areas and it will not be helpful. But if a particular document is found relevant, a citation network from its narrow subject area can be created

transiently and the search through 'highly cited documents' and 'authors/editors' in the subject specialisation can be offered. This kind of citation linking for browsing is a matter of further research in the area of interface design for OPAC in a library catering to many disciplines.

5. The worth of HyperCard as an interface design tool particularly where browsing is involved is established as inferred from the evaluators' appreciation of the screens and the browsing mechanism.

6. Icons including those for the directional buttons should have its names also included as part of the object representation on the screen. The icon alone will not divulge the meaning of the object. This is a confirmation of the finding of Kacmar and Carey[1].

7. Touch screen is a possible suggestion for public access systems like OPACs, as mouse as a pointing device is found to give some strain to some inexperienced users. Any aspect of the system which constrains the use of the system should be amended. Touch-screen technology as reported by Sears and Shneiderman[2] is becoming more and more sophisticated.

REFERENCES

1. **Kacmar, Charles A. and Jane M. Carey.** Assessing the usability of icons in user interfaces. *Behaviour & Information Technology*, 1991, 10(6), 443-447.

2. **Sears, Andrew and Ben Shneiderman.** High precision touch screens: design strategies and comparison with a mouse. *International Journal of Man-Machine Studies.* 1991, 34(4), 593-613.

BIBLIOGRAPHY

APPLE COMPUTER. *HyperCard stack design guidelines.* Reading Massachusetts: Addison-Wesley, 1989.

APTED, S.M. General purposive browsing. *Library Association Record,* 1971, 73(12), 228-230.

BARKER, Philip. *Basic principles of human-computer interface design.* London: Hutchinson, 1989.

BATES, Marcia J. The design of browsing and berrypicking techniques for the online search interface. *Online Review,* 1989, 13(5), 407-424.

BATH UNIVERSITY. *Information requirements of researchers in the social sciences. Vol.1 Text.* Bath: Bath University, 1971.

BAWDEN, David. Information systems and the stimulation of creativity. *Journal of Information Science,* 1986, 12(5), 203-216.

BOOCH, Grady. *Object oriented design with applications.* Red Wood City, California: Benjamin/Cummings Publishing Company, 1991.

BROOKS, Terrence A. Private acts and public objects : An investigation of citer motivations. *Journal of American Society for Information Science,* 1985, 36(4), 223-229.

BUCKLAND, Michael K. Bibliography, library records, and redefinition of library catalogue. *Library Resources and Technical Services,* 1988, 32(4), 299-311.

CHEW, Jane Carrasco and John WHITESIDE. *Empowering people.: Proceedings of CHI 1990, Seattle, Washington, April 1-5, 1990.* New York: Association for Computing Machinery, 1990.

CONKLIN, Jeff. Hypertext: an introduction and survey. *Computer,* 1987, 20(9), 17-41.

COOPER, William S. Getting beyond Boole. *Information Processing and Management,* 1988, 24(3), 243-248.

CULKIN, Patricia B. Rethinking OPACs: The design of assertive information systems. *Information Technology and Libraries,* 1989 8(2), 172-177.

DOSZKOCS, T.E. Natural language processing in an online catalogue. *Information Technology and Libraries*, 1983, 2(4), 364-380.

EFTHIMIADIS, EFTHIMIS N. Online public access catalogues: characteristics of the literature. *Journal of Information Science*, 1990, 16(2), 107-112.

EFTHIMIADIS, EFTHIMIS N. and Colin NEILSON. *A classified bibliography on online public access catalogues.* 2nd ed. London: British Library, 1989.

EGGHE, Leo and Ronald ROUSSEAU. *Introduction to informetrics: quantitative methods in library, documentation and information science.* Amsterdam: Elsevier Science Publisher, 1990.

ELLIS, David. A behavioural approach to information retrieval system design. *Journal of Documentation*, 1989, 45(3), 171-212.

FAYEN, Emily Gallup. *The online catalog : Improving public access to library materials.* White Plains, NY: Knowledge Industry Publications, 1983.

GARFIELD, Eugene. *Citation indexing - its theory and application in science, technology, and humanities.* New York: John Wiley, 1979.

GARFIELD, Eugene. Citation behaviour an aid or a hindrance to information retrieval. *Current Contents*, 1989, 18, 4.

GOODMAN, Danny. *HyperCard 2.0 handbook,* Toronto: Bantam Books,1990.

GREGOR, Dorothy and Carol MANDEL. Cataloguing must change. *Library Journal*, 1991, 116(6), 42-47.

HANCOCK-BEAULIEU, Micheline. Userfriendliness in online catalogues. *Program*, 1992, 26(1), 33-34.

HILDRETH, Charles R. Beyond Boolean: designing the next generation of online catalogues. *Library Trends*, 1987, 35(4), 647-667.

HILDRETH, Charles R. *Intelligent interfaces and retrieval methods for subject searching in bibliographic retrieval systems.* Washington, D.C.: Cataloging Distribution Service, Library of Congress, 1989.

HILDRETH, Charles R. ed. *The online catalogue: developments and directions.* London: The Library Association, 1989.

HIX, Deborah and Robert S. SCHULMAN. Human-computer interface development tools: a methodology for their evaluation. *Communications of the ACM,* 1991, 34(3), 74-87.

KACMAR, Charles A. and Jane M. CAREY. Assessing the usability of icons in user interfaces. *Behaviour and Information Technology,* 1991, 10(6), 443-447.

KESSLER, M.M. Bibliographic coupling between scientific papers. *American Documentation,* 1963, 14(1), 10-25.

KING, Alan. Room with a view: Microsoft Windows and CD-ROM. *Database,* 1991, 14(2), 99-101.

KOCHEN, Manfred, ed. *The growth of knowledge: Readings on organization and retrieval of information.* New York: John Wiley, 1967

KWOK, K.L. On the use of bibliographically related titles for the enhancement of document representations. *Information Processing and Management,* 1988, 24(2), 123-131.

LANGE, Holley R. The voice as computer interface: a look at tomorrow's technologies. *The Electronic Library,* 1991, 9(1), 7-11.

LARGE, Andy. The user interface to CD-ROM databases. *Journal of Librarianship and Information Science,* 1991, 23(4), 203-217.

LARSON, Ray R. Between Scylla and Charybdis : subject searching in the online catalog. *Advances in Librarianship,* 1990, 15, 175-236.

LIPETZ, B. and P.J. PAULSON. A study of the impact of introducing an online subject catalog at the New York State Library. *Library Trends,* 1987, 35(4), 597-617.

MAC AOGAIN, Eoghan and Ronan REILLY. Discourse theory and interface design: the ease of pointing with the mouse. *International Journal of Man-Machine Studies,* 1990, 32(5), 591-602.

MARKEY, Karen. Subject searching experiences and needs of online catalog users: implications for library classification. *Library Resources and Technical Services,* 1985, 29(1), 34-51.

MATTHEWS, Joseph R., G.S. LAWRENCE and D.K. FERGUSON. *Using online catalogs: a nation wide survey.* New York: Neal-Schuman 1983.

McKNIGHT, Cliff and Andrew DILLON. *Hypertext in context.* Cambridge: Cambridge University Press, 1991.

MITEV, Nathalie N., Gillian M. VENNER and Stephen WALKER. *Designing an online public access catalogue.* London: British Library, 1985.

NELSON, Theodor H. Managing immense storage. *Byte,* 1988, 13(1), 225-238.

NIELSEN, Jakob, I. FREHR and H.O. NYMAND. The learnability of HyperCard as an object-oriented programming. *Behaviour and Information Technology,* 1991, 10(2), 111-120.

O'BRIEN, Ann. Relevance as an aid to evaluation in OPACs. *Journal of Information Science,* 1990, 16(4), 265-271.

ODDY, R.N. et al. *Information Retrieval Research.* London: Butterworths, 1981.

QUINT, Barbara. Journal article coverage in online library catalogues: the next stage for online databases? *Online,* 1987, 11(1), 87-90.

RAVDEN, Susannah and Graham JOHNSON. *Evaluating usability of human-computer interfaces: A practical method.* Chichester: Ellis Horwook, 1989.

RITCH, Alan. Ten years of monitoring MELVYL: a librarian's view. *Information Technology and Libraries,* 1992, 11(2), 172-179.

SALKIND, Neil J. *The big Mac book,* Carmel, USA: Que Corporation, 1989.

SEARS, Andrew and Ben SHNEIDERMAN. High precision touch screens: design strategies and comparison with a mouse. *International Journal of Man-Machine Studies,* 1991, 34(4), 593-613.

SHEPHARD, Michael A., C.R. WATTERS and Yao CAI. Transient hypergraphs for citation networks. *Information Processing and Management,* 1990, 26(3), 395-412.

SHNEIDERMAN, Ben. *Designing the user interface: strategies for effective human-computer interaction.* Reading, Mass.: Addison-Wesley, 1987.

SMALL, Henry G. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, 1973, 24(4), 265-269.

STERN, B.T. ed. *Information and Innovation.* Amsterdam: North-Holland, 1982.

THOMPSON, R.H. and W.B. CROFT. Support for browsing in an intelligent text retrieval system. *International Journal of Man-Machine Studies*, 1989, 30(6), 639-668.

TSENG, Gwyneth, Goff SARGENT and Jack MEADOWS. Round referencing. *Nature*, 1989, 337, 516.

VAN STYVENDAELE, B.J.H. University scientists as seekers of information: sources of reference to periodical literature. *Journal of Librarianship*, 1977, 9(4), 270-277.

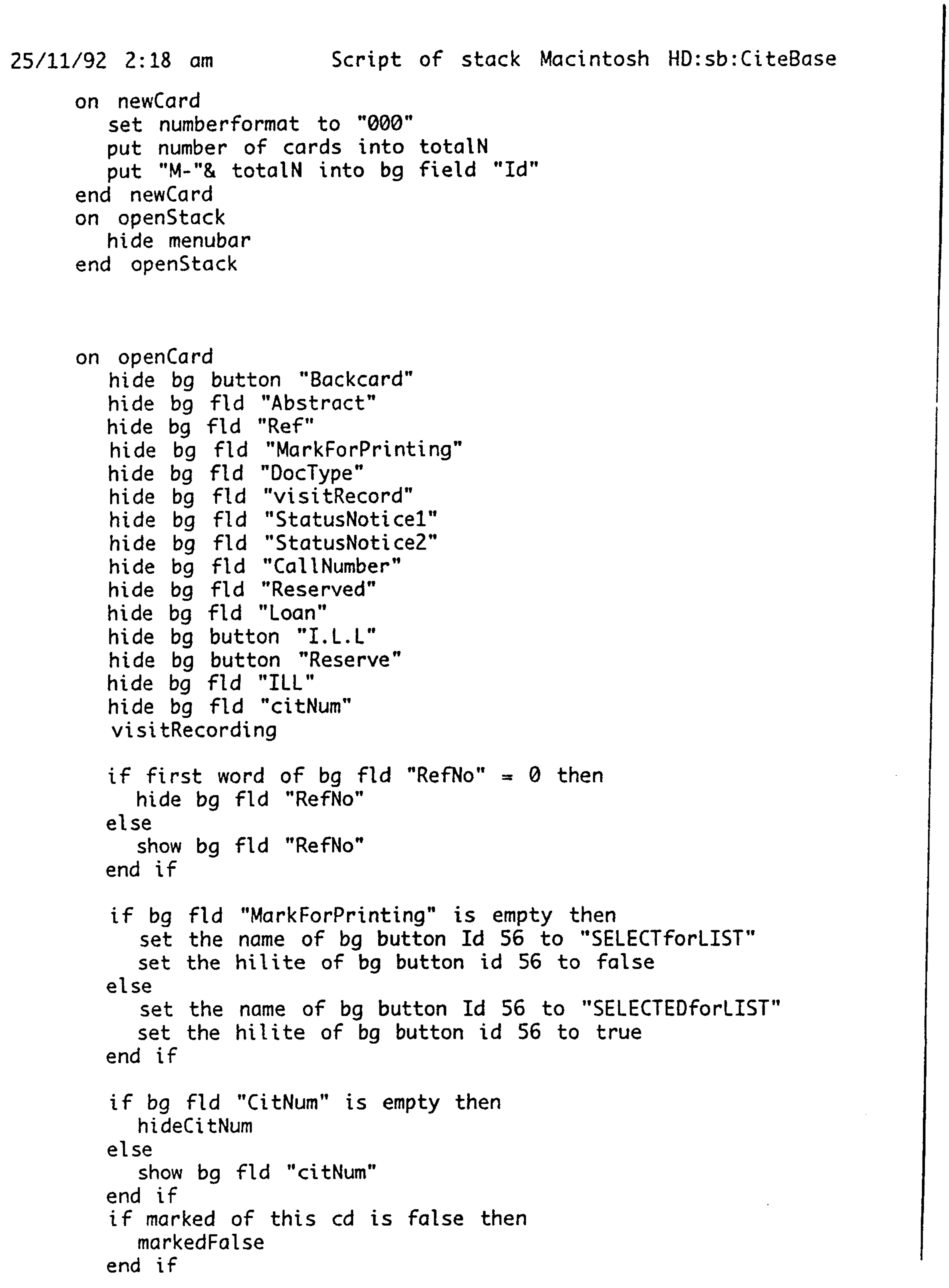
VIRZI, Robert A. Refining the test phase of usability evaluation: How many subjects is enough? *Human Factors*, 1992, 34(4), 457-468.

WILLIAMS, Martha E. and Thomas H. HOGAN, comps. *National Online Meeting - 1981, Proceedings.* Medford, NJ: Learned Information, 1981.

WINKLER, Dan and Scot KAMINS. *HyperTalk 2.0: the book.* New York: Bantam Books, 1990.

YEE, Martha M. System design and cataloging meet the user: user interfaces to online public access catalogs. *Journal of the American Society for Information Science*, 1991, 42(2), 78-98.

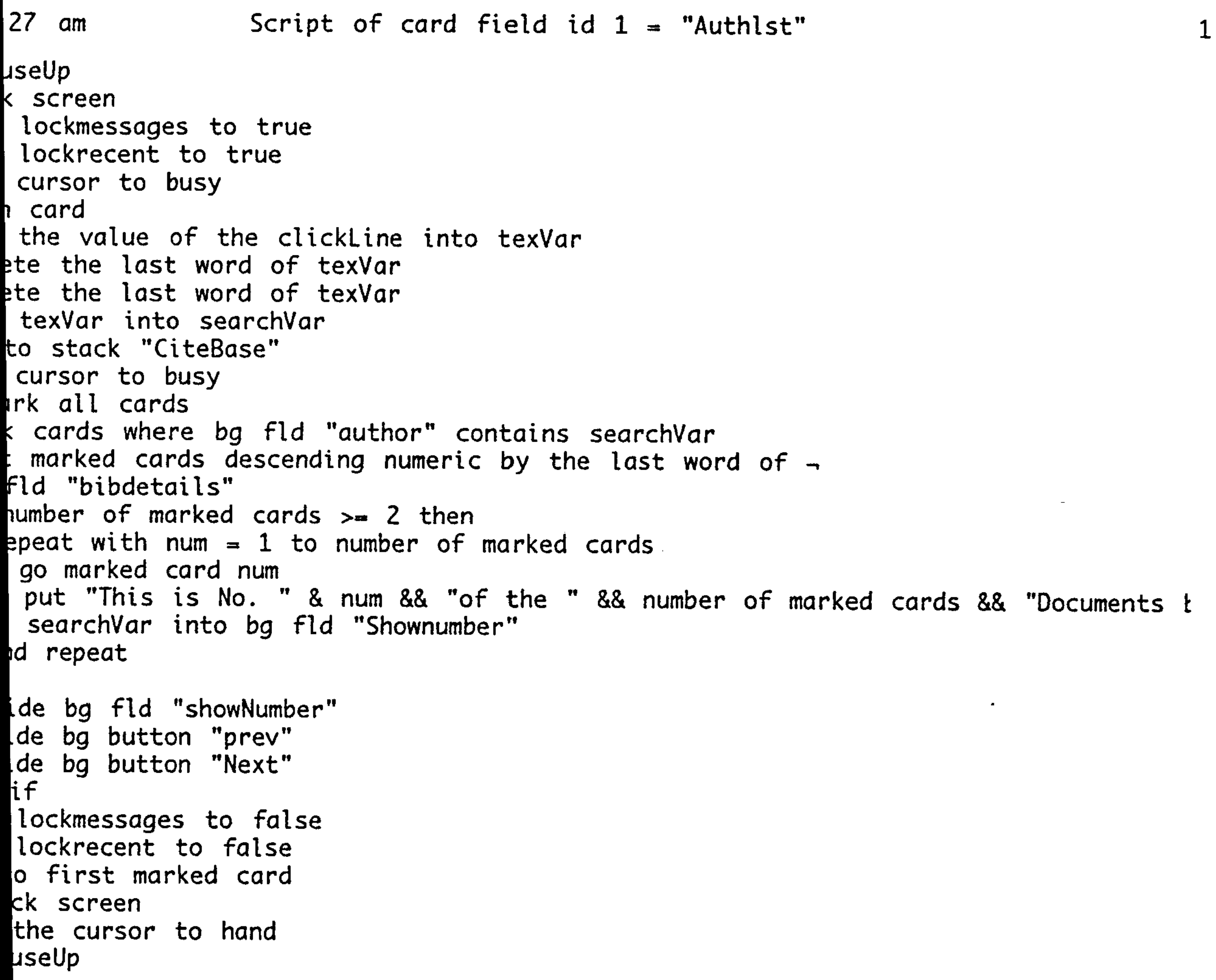
APPENDIX 1 STACK SCRIPT FOR CITEBASE,THE MAIN STACK



```
25/11/92  2:18  am          Script  of  stack  Macintosh  HD:sb:CiteBase
    on  newCard
      set  numberformat  to  "000"
      put  number  of  cards  into  totalN
      put  "M-"&  totalN  into  bg  field  "Id"
    end  newCard
    on  openStack
      hide  menubar
    end  openStack



    on  openCard
      hide  bg  button  "Backcard"
      hide  bg  fld  "Abstract"
      hide  bg  fld  "Ref"
      hide  bg  fld  "MarkForPrinting"
      hide  bg  fld  "DocType"
      hide  bg  fld  "visitRecord"
      hide  bg  fld  "StatusNotice1"
      hide  bg  fld  "StatusNotice2"
      hide  bg  fld  "CallNumber"
      hide  bg  fld  "Reserved"
      hide  bg  fld  "Loan"
      hide  bg  button  "I.L.L"
      hide  bg  button  "Reserve"
      hide  bg  fld  "ILL"
      hide  bg  fld  "citNum"
      visitRecording

      if  first  word  of  bg  fld  "RefNo"  =  0  then
        hide  bg  fld  "RefNo"
      else
        show  bg  fld  "RefNo"
      end  if

      if  bg  fld  "MarkForPrinting"  is  empty  then
        set  the  name  of  bg  button  Id  56  to  "SELECTforLIST"
        set  the  hilite  of  bg  button  id  56  to  false
      else
        set  the  name  of  bg  button  Id  56  to  "SELECTEDforLIST"
        set  the  hilite  of  bg  button  id  56  to  true
      end  if

      if  bg  fld  "CitNum"  is  empty  then
        hideCitNum
      else
        show  bg  fld  "citNum"
      end  if
      if  marked  of  this  cd  is  false  then
        markedFalse
      end  if
    if number of marked cards = 1 then
      markedCardOne
    end if

    if number of marked cards >= 2 then
      if the marked of this card is true then
        markedCardMoreThanOne
      end if
    end if
  end openCard

  on markedFalse
    hide bg fld "showNumber"
    put empty into bg fld "showNumber"
    hide bg button "prev"
    hide bg button "Next"
  end markedFalse
```

```
 markedCardOne
hide bg fld "showNumber"
hide bg button "prev"
hide bg button "Next"
d  markedCardOne

 markedCardMoreThanOne
show bg fld "showNumber"
put the fourth word of bg fld "ShowNumber" into markedNum
if markedNum = 1 then
  hide bg button "Prev"
else
  show bg button "prev"
end if
if markedNum = number of marked cards then
  hide bg button "Next"
else
  show bg button "Next"
end if
d  markedCardMoreThanOne

 hideRefNo
hide bg fld "RefNo"
d  hideRefNo

 hideCitNum
hide bg fld "CitNum"
d  hideCitNum


 visitRecording
if bg fld "visitRecord" is empty then
  put "Visit 1" & return into bg fld "visitRecord"
  put "√ First visit to this Card" into bg fld "VisitRecordNotice"
else
  get number of lines in bg fld "visitRecord"
  put it into visitNo
  put "Visit" && (visitNo + 1) & return after bg fld ¬
  "visitRecord"
  put "√ Visited this Card" && (visitNo + 1) && "times" into ¬
  bg fld "VisitRecordNotice"
end if
d  visitRecording
```

APPENDIX 2 SCRIPT FOR THE LISTING OF DOCUMENTS



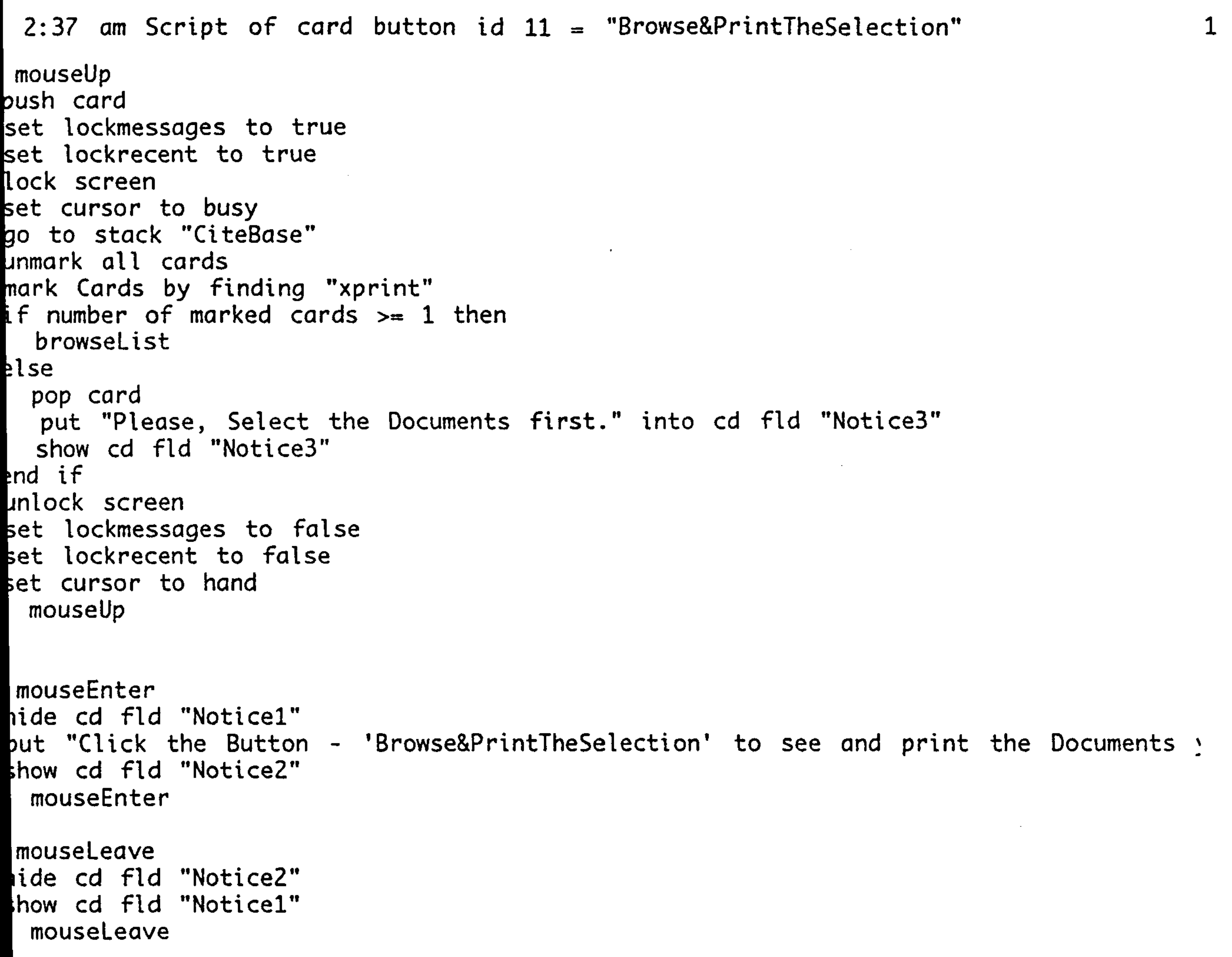
```
27 am           Script of card field id 1 = "Authlst"                        1

iseUp
k screen
 lockmessages to true
 lockrecent to true
 cursor to busy
n card
 the value of the clickLine into texVar
ete the last word of texVar
ete the last word of texVar
 texVar into searchVar
to stack "CiteBase"
 cursor to busy
rk all cards
k cards where bg fld "author" contains searchVar
t marked cards descending numeric by the last word of ¬
fld "bibdetails"
number of marked cards >= 2 then
epeat with num = 1 to number of marked cards
 go marked card num
 put "This is No. " & num && "of the " && number of marked cards && "Documents b
 searchVar into bg fld "Shownumber"
nd repeat

ide bg fld "showNumber"
ide bg button "prev"
ide bg button "Next"
if
 lockmessages to false
 lockrecent to false
o first marked card
ck screen
the cursor to hand
iseUp
```

APPENDIX 3 SCRIPT FOR COCITATION SEARCH

```
 3:10 am      Script of card button id 1 = "CocitationSearch"                    1
mouseUp
o to cd "CocitSearch"
 mouseUp


mouseEnter
ide cd fld "Notice1"
ut "Click the Button - 'CocitationSearch' to initiate a search for Documents whicl
how cd fld "Notice2"
 mouseEnter

ouseLeave
ide cd fld "Notice2"
ow cd fld "Notice1"
 mouseLeave



2 3:13 am      Script of card field id 1 = "CoCitClickList"                        1

 mouseUp
 set the cursor to busy
 global searchClick, nameVar
 get the value of the clickLine
 put the third word of it & " " after searchClick
 put the fifth word of it & " & " after nameVar
 set the cursor to hand
d mouseUp
```

```
 mouseUp
lock screen
set the cursor to busy
push card
set the lockMessages to true
set the lockRecent to true
global searchClick, nameVar
delete the last word of nameVar
if nameVar is not empty then
  answer nameVar with "OK" or "CANCEL"
  if it is "OK" then
    set the cursor to busy
    put searchClick into searchVar
    put empty into searchClick
    go to stack "CiteBase"
    unmark all cards
    mark cards where bg fld "Ref" contains first word of searchVar
    unmark cards where the second word of searchVar is not ¬
    in bg fld "Ref"
    put number of marked cards into markNo
    if markNo = 0 then
      go back
      put "No results. Try with another combination" into cd fld ¬
      "Notice"
      show cd fld "Notice"
      put empty into nameVar
    else
      put "There are " & markNo && "documents cociting " & nameVar ¬
      into CitNumVar
      repeat with count = 1 to markNo
        go to marked card count
        put count after citeVar
        put ". ( " & bg field "Id" & " ) " after citeVar
        if number of lines in bg fld "Author" > 1 then
          put line 1 of bg fld "Author" && "and others" after citeVar
        else
          put line 1 of bg field "Author" after citeVar
        end if
        put " : " & bg fld "Title" & ", " & the last word of ¬
        bg fld "Bibdetails" & return & return after citeVar
      end repeat
      unmark all cards
      go stack "CiteFront"
      go to cd "CocitationList"
      put CiteVar into cd fld "CocitationIndex"
      put citnumVar into cd fld "cocitnumber"
      put empty into nameVar
      put empty into citnumVar
    end if
  else
    put empty into searchClick
    put empty into nameVar
  end if
lse
  pop card
nd if
nlock screen
et the lockMessages to false
et the lockRecent to false
et the cursor to hand
 mouseUp
```

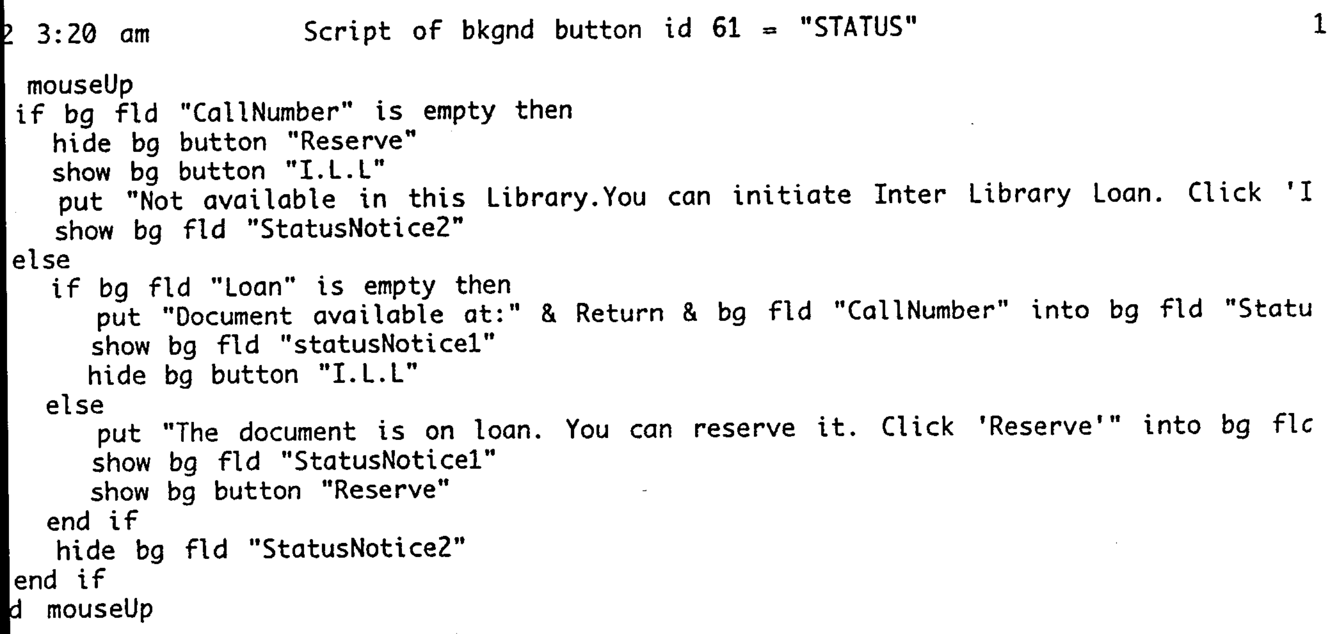
APPENDIX 4 SCRIPT FOR BROWSE AND PRINT



```
 2:37  am Script of card button id 11 = "Browse&PrintTheSelection"                    1

 mouseUp
ush card
set lockmessages to true
set lockrecent to true
lock screen
set cursor to busy
o to stack "CiteBase"
nmark all cards
ark Cards by finding "xprint"
f number of marked cards >= 1 then
  browseList
lse
  pop card
   put "Please, Select the Documents first." into cd fld "Notice3"
  show cd fld "Notice3"
nd if
nlock screen
set lockmessages to false
set lockrecent to false
et cursor to hand
 mouseUp


 mouseEnter
ide cd fld "Notice1"
ut "Click the Button - 'Browse&PrintTheSelection' to see and print the Documents
how cd fld "Notice2"
 mouseEnter

 mouseLeave
ide cd fld "Notice2"
how cd fld "Notice1"
 mouseLeave
```

```
browseList
ut empty into listVar
ort marked cards by the first line of bg fld "Author"
epeat with count = 1 to number of marked cards
  go to marked card count
  put count & ". ( " & bg fld "Id" & " ) " after listVar
  if number of lines in bg fld "Author" = 2 then
     put line 1 of bg fld "Author" & " and " & line 2 of ¬
     bg fld "Author" after listVar
  end if
  if number of lines in bg fld "Author" >= 3 then
     put line 1 of bg fld "Author" && "and others" after listVar
  end if
  if number of lines in bg fld "Author" = 1 then
     put line 1 of bg field "Author" after listVar
  end if
  put " : " & bg fld ¬
  "Title" & ". " & line 1 of bg fld "Bibdetails" & ". " & ¬
  line 2 of bg fld "Bibdetails" & return ¬
  after listVar
  if bg fld "CallNumber" is not empty then
     put "       -Location: " & line 1 of bg fld "Callnumber" & " " & line 2 of bg f
     "CallNumber" & " " & line 3 of bg fld "Callnumber" after listVar
  else
     put "       -Not available in this library" after listVar
  end if
  if bg fld "Loan" is not empty then
     put " (Now on loan)" after listVar
  end if
  put return & return after listVar
end repeat
go to cd "browseCard" of stack "CiteFront"
put listVar into cd fld "browse" of cd "browseCard"
  browseList
```

APPENDIX 5 SCRIPT FOR STATUS INFORMATION



```
2  3:20  am           Script of bkgnd button id 61 = "STATUS"                      1

 mouseUp
 if bg fld "CallNumber" is empty then
   hide bg button "Reserve"
   show bg button "I.L.L"
   put "Not available in this Library.You can initiate Inter Library Loan. Click 'I
   show bg fld "StatusNotice2"
else
   if bg fld "Loan" is empty then
      put "Document available at:" & Return & bg fld "CallNumber" into bg fld "Statu
      show bg fld "statusNotice1"
      hide bg button "I.L.L"
  else
      put "The document is on loan. You can reserve it. Click 'Reserve'" into bg flc
      show bg fld "StatusNotice1"
      show bg button "Reserve"
  end if
   hide bg fld "StatusNotice2"
end if
d  mouseUp
```

APPENDIX 6 TASK ANALYSIS SCHEDULES

HYPERCITE : LIBRARY CATALOGUE WITH A CITATION NETWORK APPROACH

This Online Public Access Catalogue (OPAC) gives opportunity for browsing through the references and citations of a document.

Please attempt the following questions while exploring the OPAC.

1. Give the volume no. of the journal of the 'most highly cited document'.

2. Which is the most recent article citing 'the most highly cited document'? (Give the name of the first author only)

3. How many articles are there by Hagge JK ?

4. How many items cocite the two documents: Jensen RJ and others (1984) and Spielberger and others (1984)?

5. How many items cocite Chang TP and others(1981) and Hagge JK (1988)?

6. Is the document by Hopper and others (1992) available in the library

7. Select the 3rd reference in Hopper and others (1992) for printing

---Thank you very much for your cooperation.

Evaluation of Questions

1. The problems if any, in the negotiation of the questions:

Q. 1.

Q. 2.

Q. 3.

Q. 4.

Q. 5.

Q.6.

Q. 7.

2. Please comment on the following:

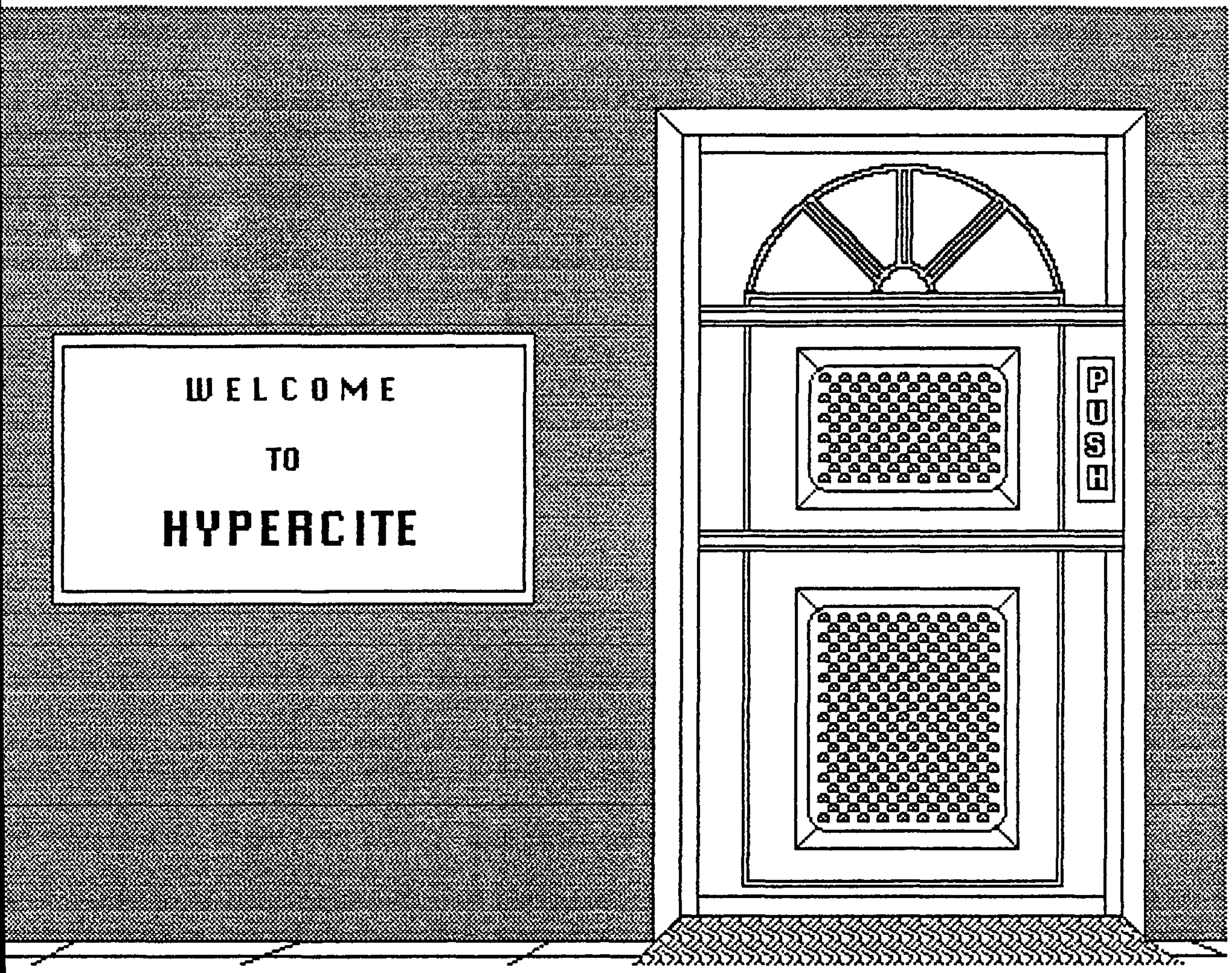
The Menu screen:

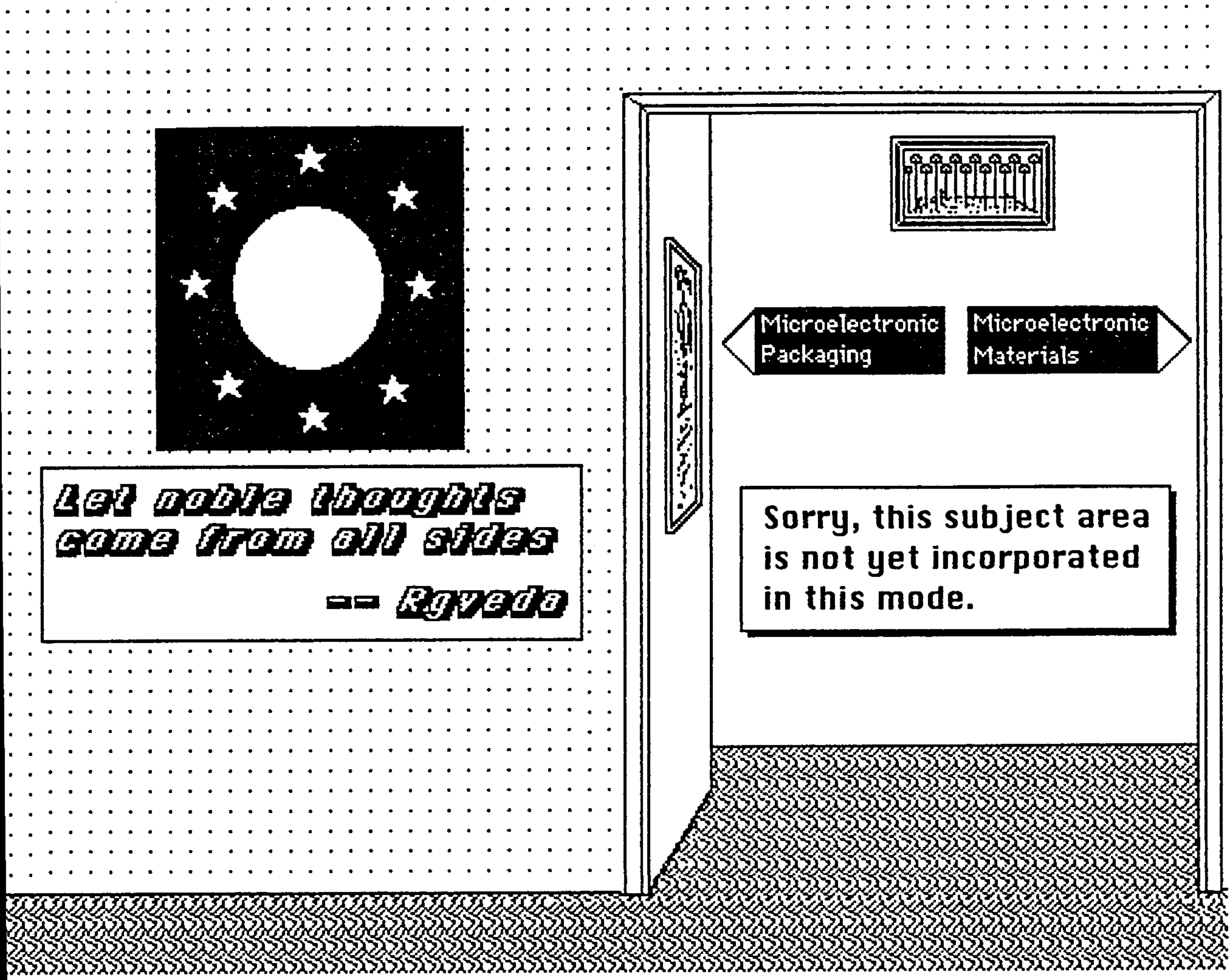
The terminology:

The document record screen layout:

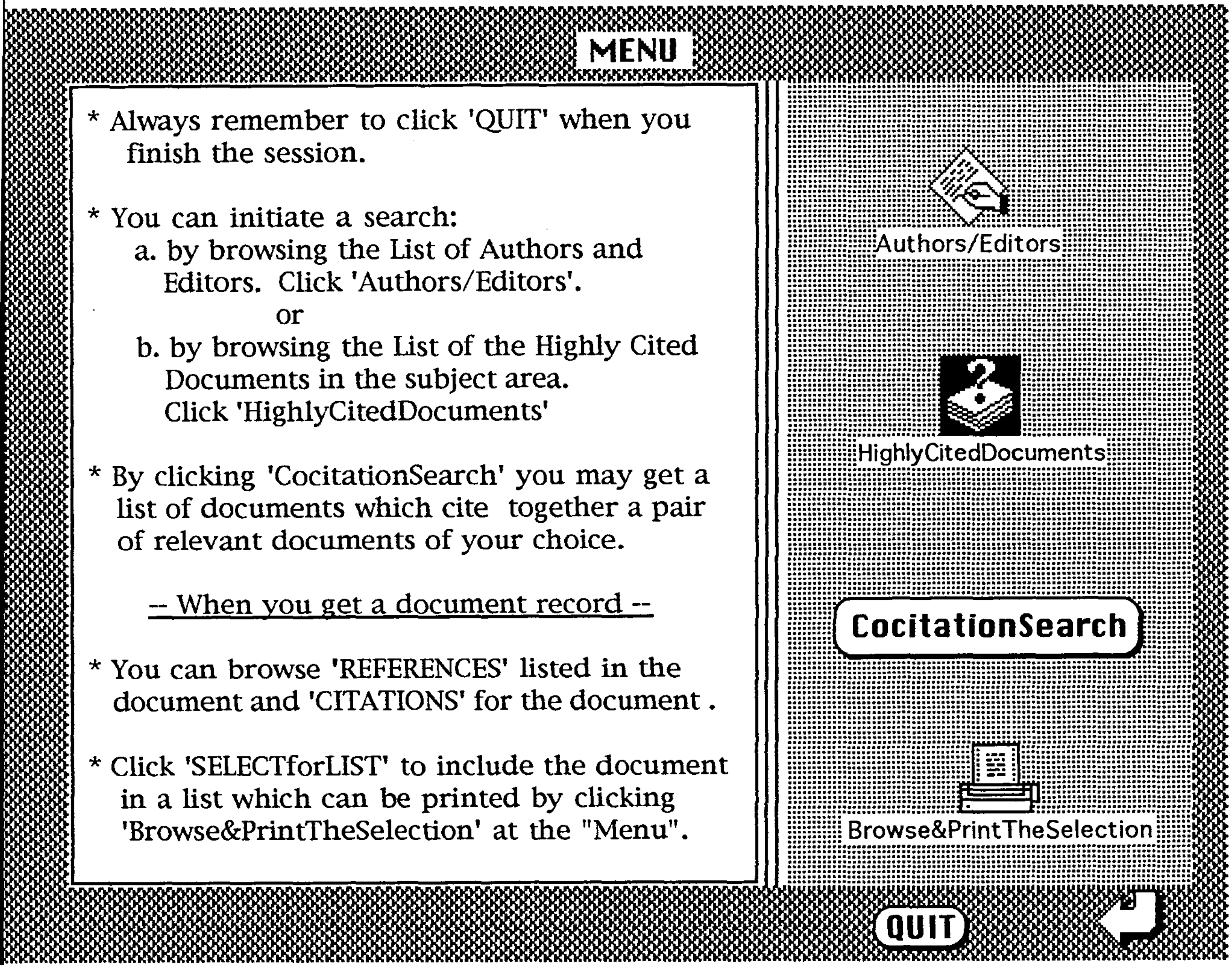
Usability of buttons:

3. General comments:


WELCOME
TO
HYPERCITE
PUSH



Sorry, this subject area is not yet incorporated in this mode.



MENU

* Always remember to click 'QUIT' when you finish the session.

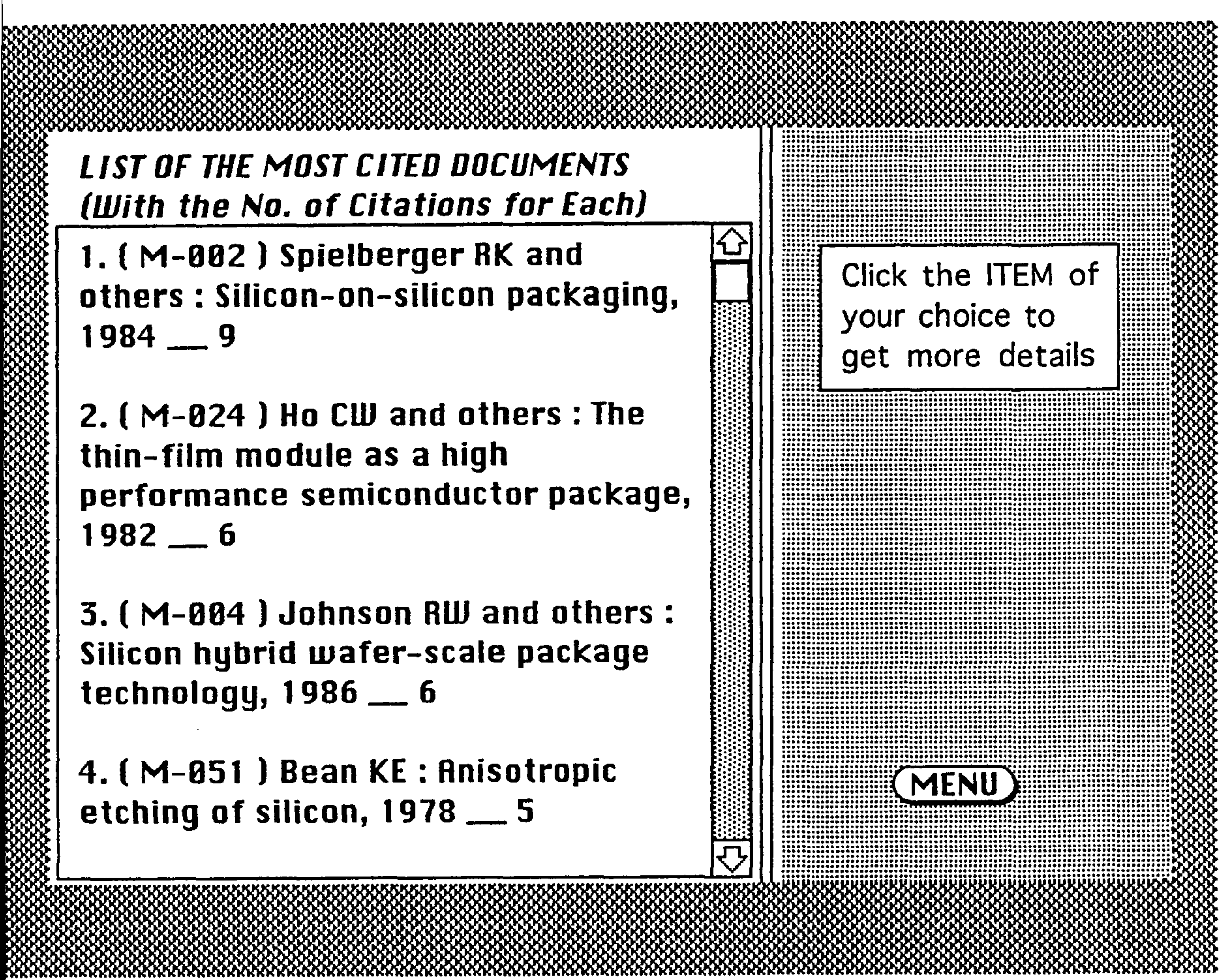
* You can initiate a search:
  a. by browsing the List of Authors and Editors. Click 'Authors/Editors'.
  or
  b. by browsing the List of the Highly Cited Documents in the subject area. Click 'HighlyCitedDocuments'

* By clicking 'CocitationSearch' you may get a list of documents which cite together a pair of relevant documents of your choice.

-- When you get a document record --

* You can browse 'REFERENCES' listed in the document and 'CITATIONS' for the document .

* Click 'SELECTforLIST' to include the document in a list which can be printed by clicking 'Browse&PrintTheSelection' at the "Menu".



LIST OF THE MOST CITED DOCUMENTS
(With the No. of Citations for Each)

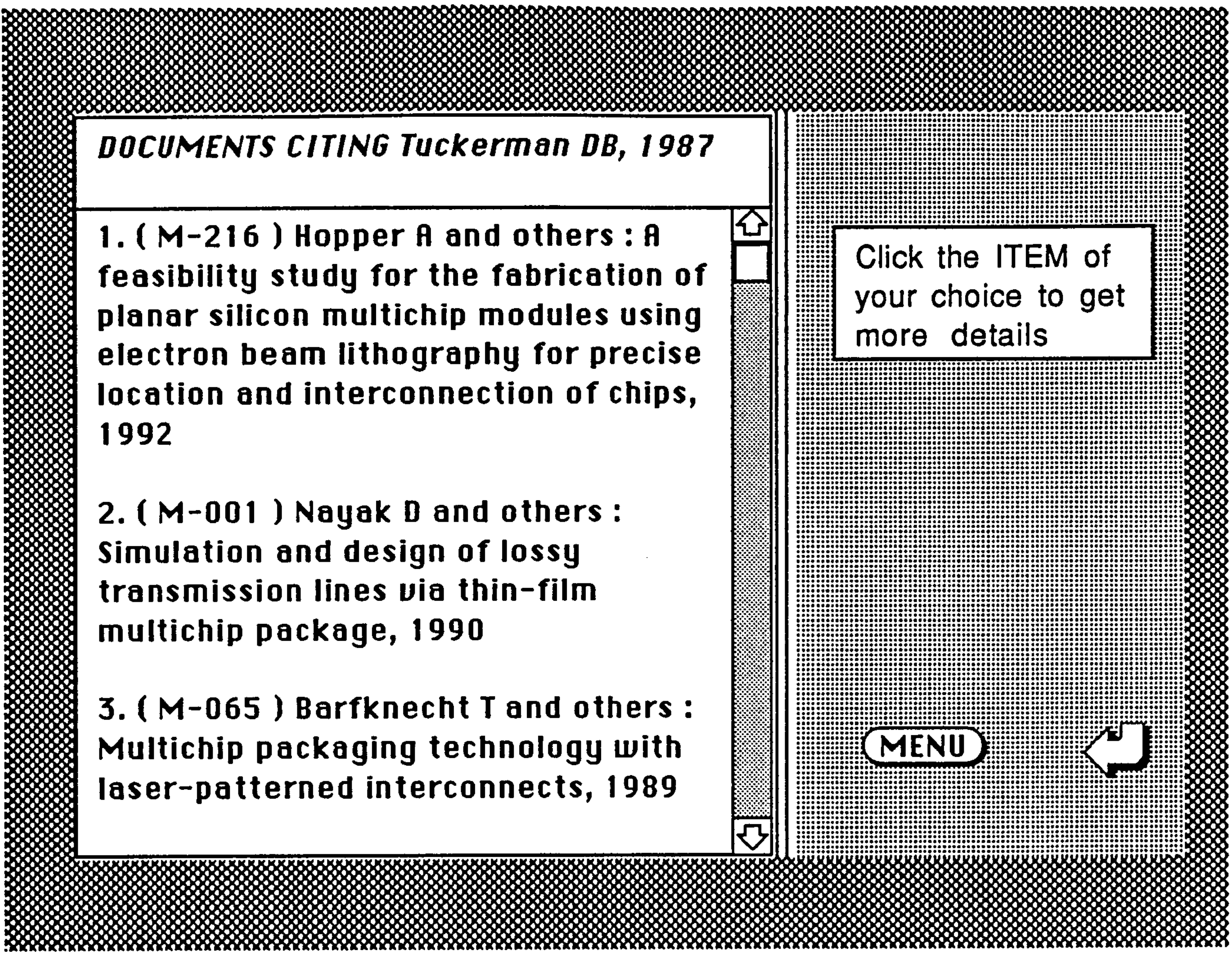
1. (M-002) Spielberger RK and others : Silicon-on-silicon packaging, 1984 __ 9

2. (M-024) Ho CW and others : The thin-film module as a high performance semiconductor package, 1982 __ 6

3. (M-004) Johnson RW and others : Silicon hybrid wafer-scale package technology, 1986 __ 6

4. (M-051) Bean KE : Anisotropic etching of silicon, 1978 __ 5

Click the ITEM of your choice to get more details



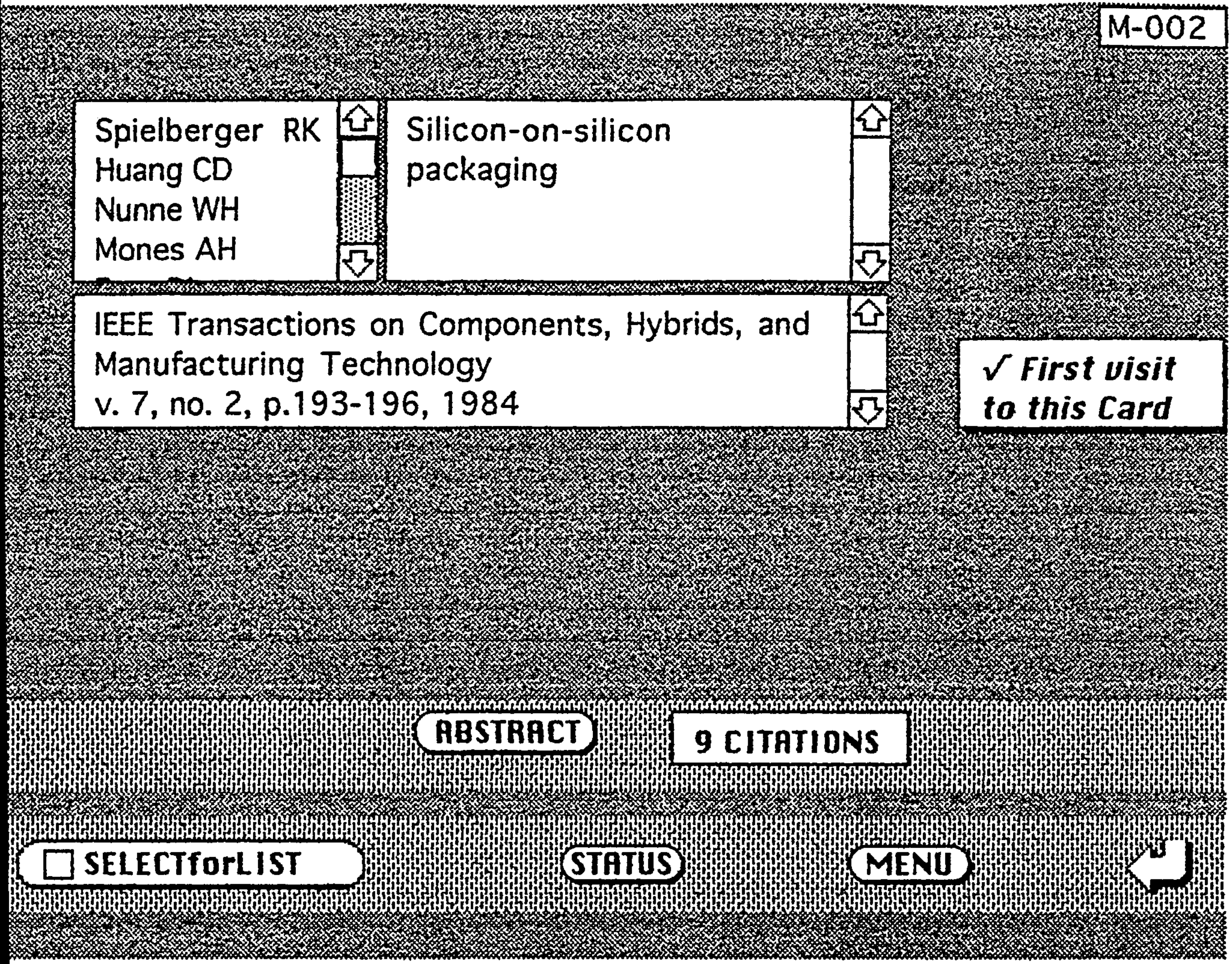
DOCUMENTS CITING Tuckerman DB, 1987

1. (M-216) Hopper A and others : A feasibility study for the fabrication of planar silicon multichip modules using electron beam lithography for precise location and interconnection of chips, 1992

2. (M-001) Nayak D and others : Simulation and design of lossy transmission lines via thin-film multichip package, 1990

3. (M-065) Barfknecht T and others : Multichip packaging technology with laser-patterned interconnects, 1989

Click the ITEM of your choice to get more details

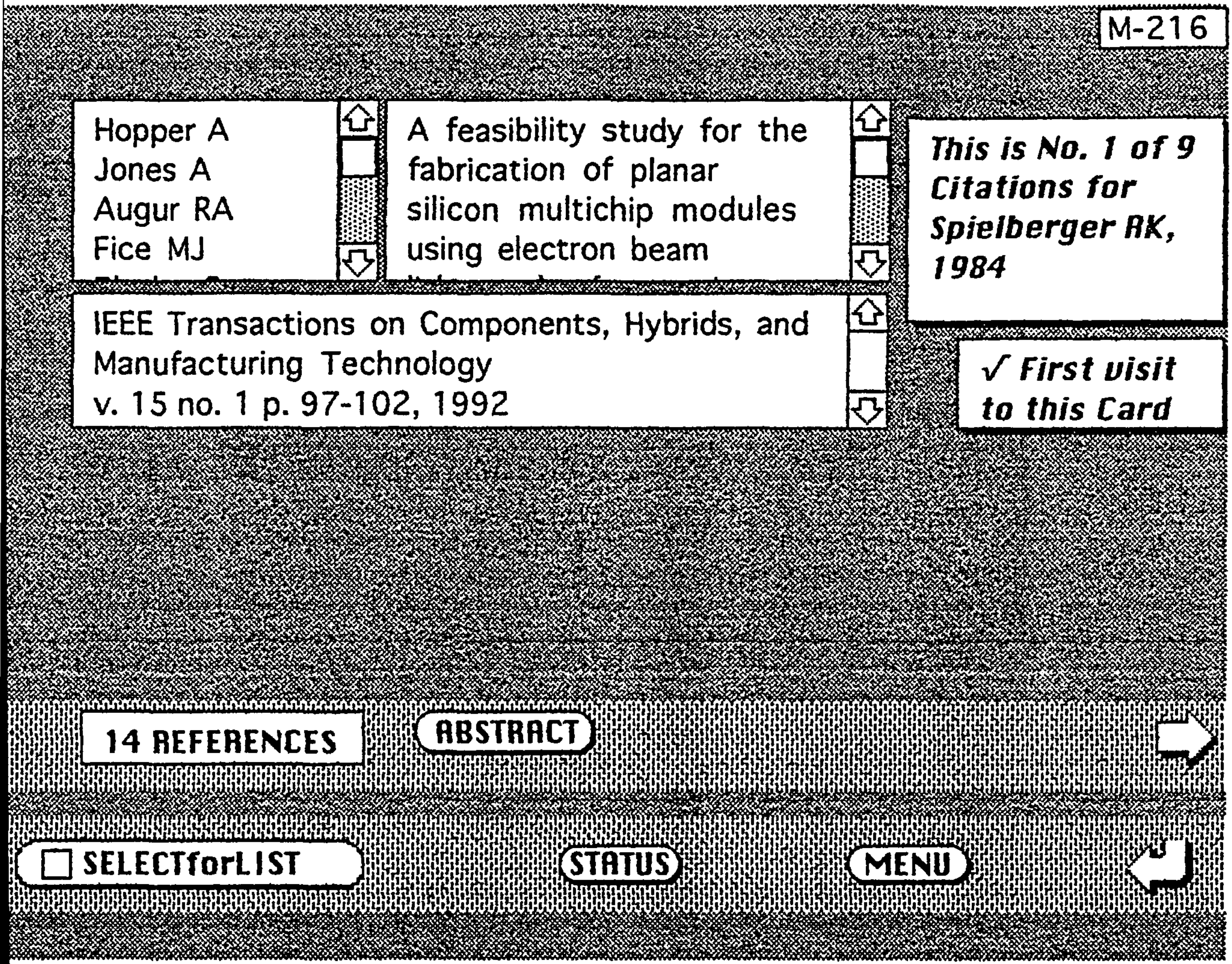


Spielberger RK
Huang CD
Nunne WH
Mones AH

Silicon-on-silicon packaging

IEEE Transactions on Components, Hybrids, and Manufacturing Technology
v. 7, no. 2, p.193-196, 1984

✓ First visit to this Card



M-216

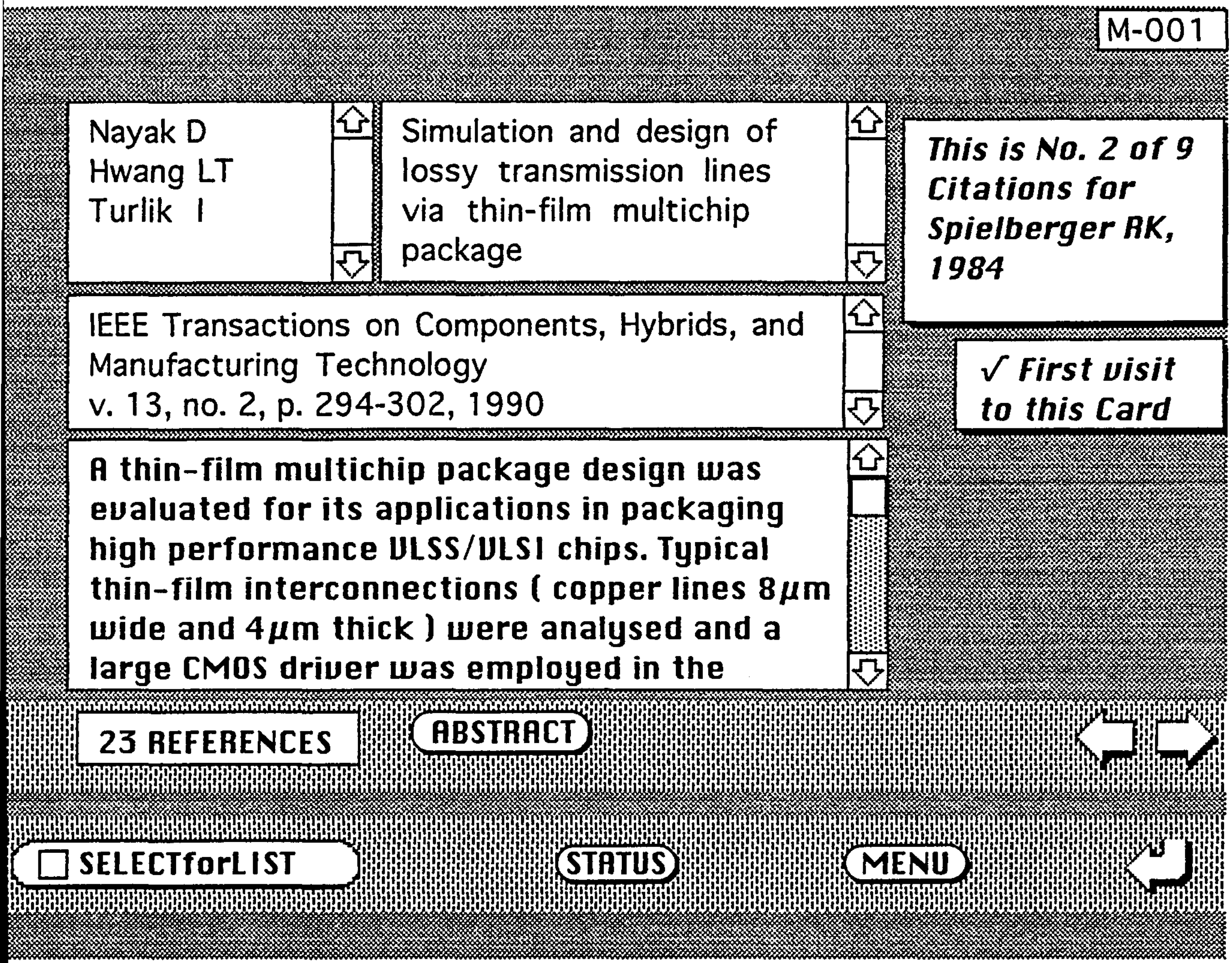
Hopper A
Jones A
Augur RA
Fice MJ

A feasibility study for the fabrication of planar silicon multichip modules using electron beam

IEEE Transactions on Components, Hybrids, and Manufacturing Technology
v. 15 no. 1 p. 97-102, 1992

This is No. 1 of 9 Citations for Spielberger RK, 1984

√ First visit to this Card



Nayak D
Hwang LT
Turlik I

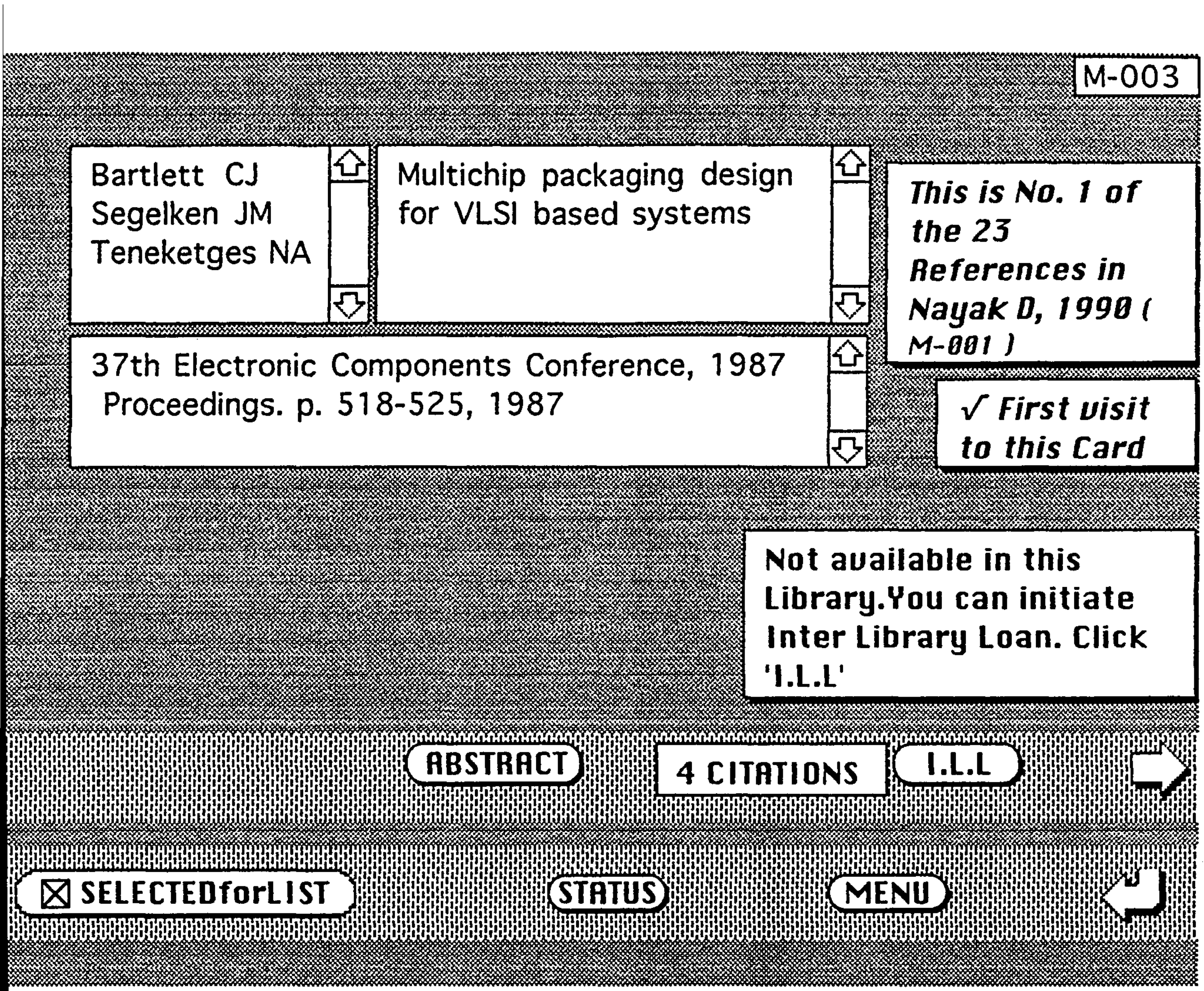
Simulation and design of lossy transmission lines via thin-film multichip package

This is No. 2 of 9 Citations for Spielberger RK, 1984

IEEE Transactions on Components, Hybrids, and Manufacturing Technology
v. 13, no. 2, p. 294-302, 1990

√ First visit to this Card

A thin-film multichip package design was evaluated for its applications in packaging high performance VLSS/VLSI chips. Typical thin-film interconnections (copper lines 8μm wide and 4μm thick) were analysed and a large CMOS driver was employed in the



M-003

Bartlett CJ
Segelken JM
Teneketges NA

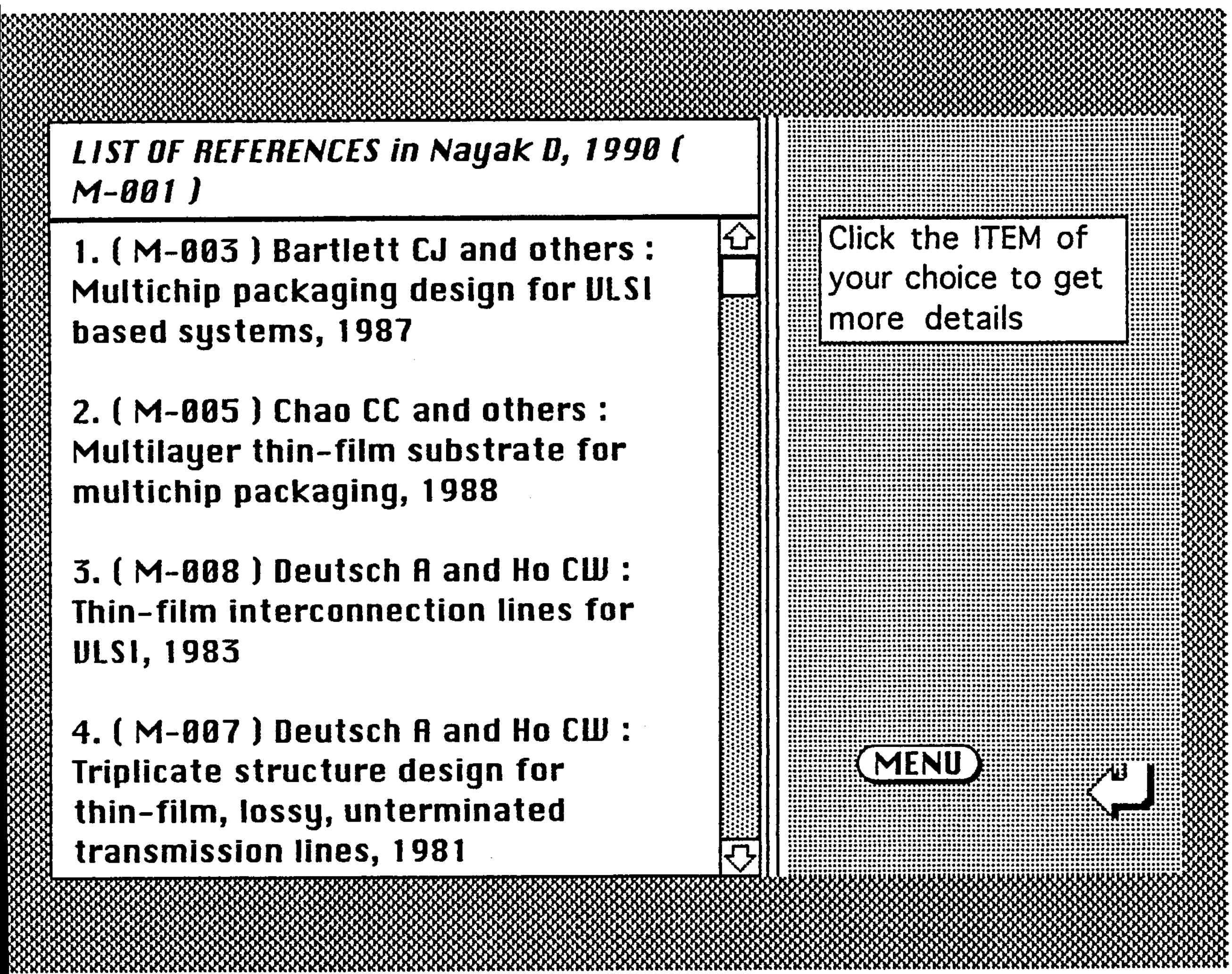
Multichip packaging design for VLSI based systems

37th Electronic Components Conference, 1987
Proceedings. p. 518-525, 1987

This is No. 1 of the 23 References in Nayak D, 1990 (M-001)

√ First visit to this Card

Not available in this Library.You can initiate Inter Library Loan. Click 'I.L.L'



LIST OF REFERENCES in Nayak D, 1990 (M-001)

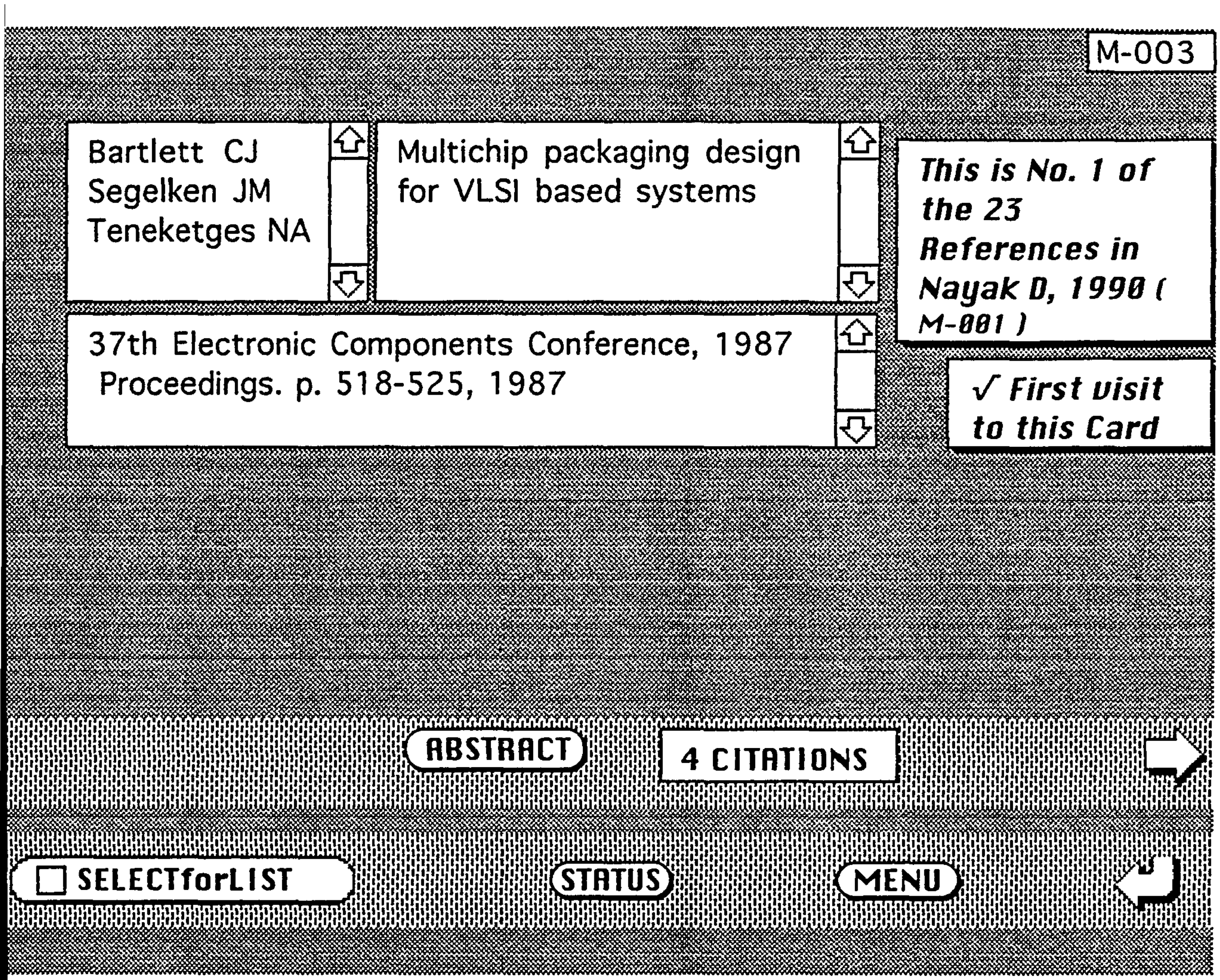
1. (M-003) Bartlett CJ and others : Multichip packaging design for ULSI based systems, 1987

2. (M-005) Chao CC and others : Multilayer thin-film substrate for multichip packaging, 1988

3. (M-008) Deutsch A and Ho CW : Thin-film interconnection lines for ULSI, 1983

4. (M-007) Deutsch A and Ho CW : Triplicate structure design for thin-film, lossy, unterminated transmission lines, 1981

Click the ITEM of your choice to get more details



M-003

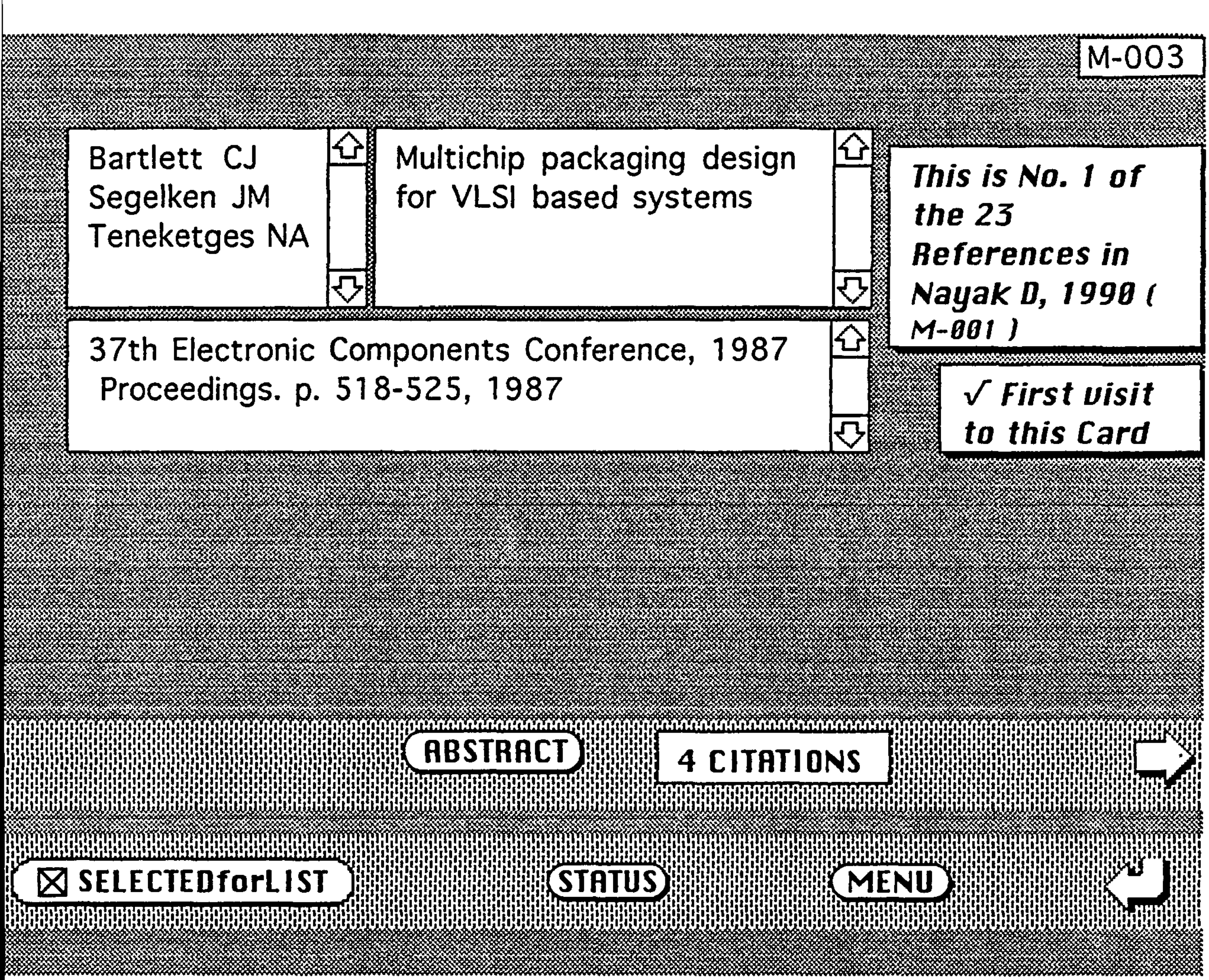
Bartlett CJ
Segelken JM
Teneketges NA

Multichip packaging design for VLSI based systems

37th Electronic Components Conference, 1987
Proceedings. p. 518-525, 1987

This is No. 1 of the 23 References in Nayak D, 1990 (M-001)

✓ First visit to this Card



M-003

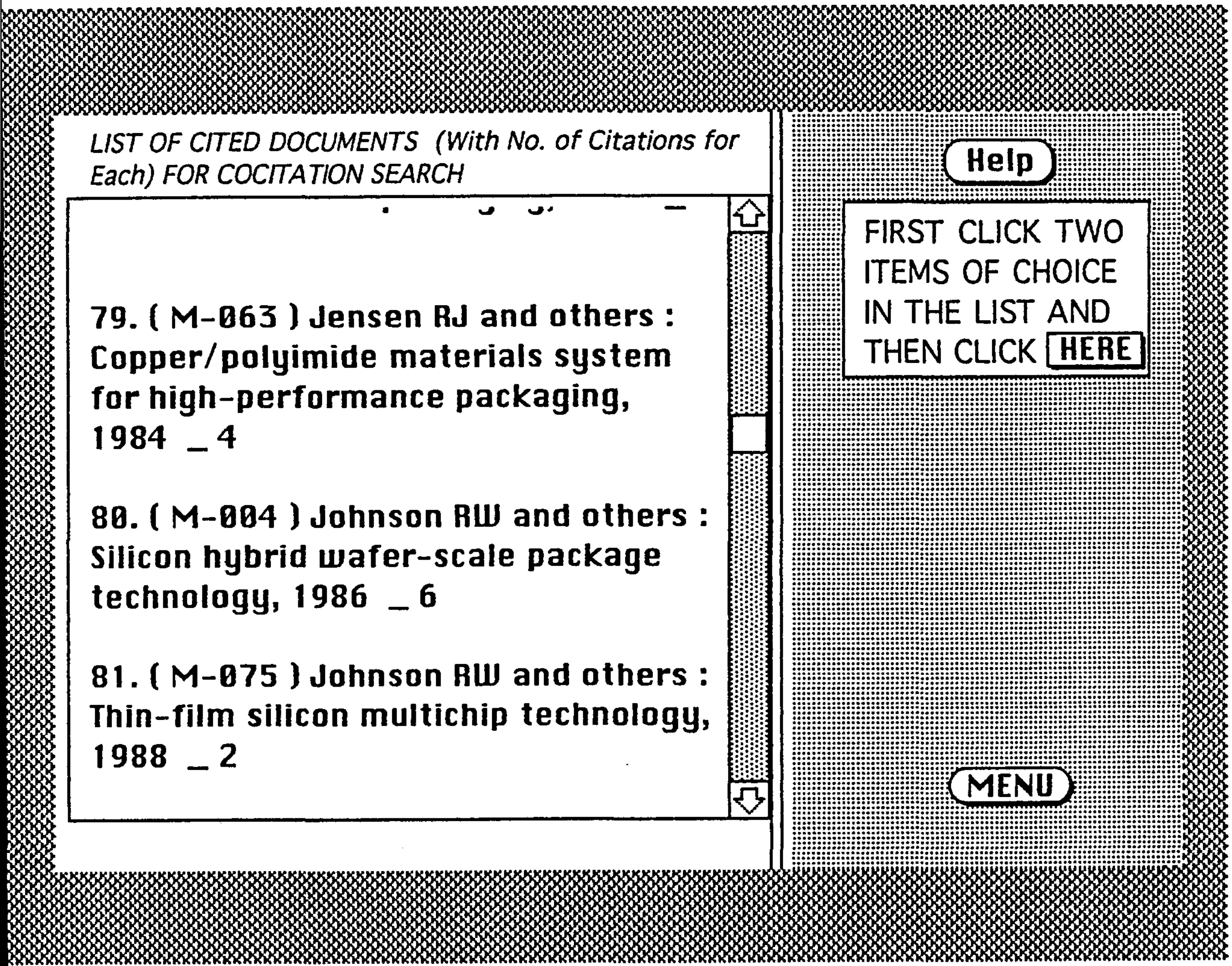
Bartlett CJ
Segelken JM
Teneketges NA

Multichip packaging design for VLSI based systems

37th Electronic Components Conference, 1987 Proceedings. p. 518-525, 1987

This is No. 1 of the 23 References in Nayak D, 1990 (M-001)

√ First visit to this Card



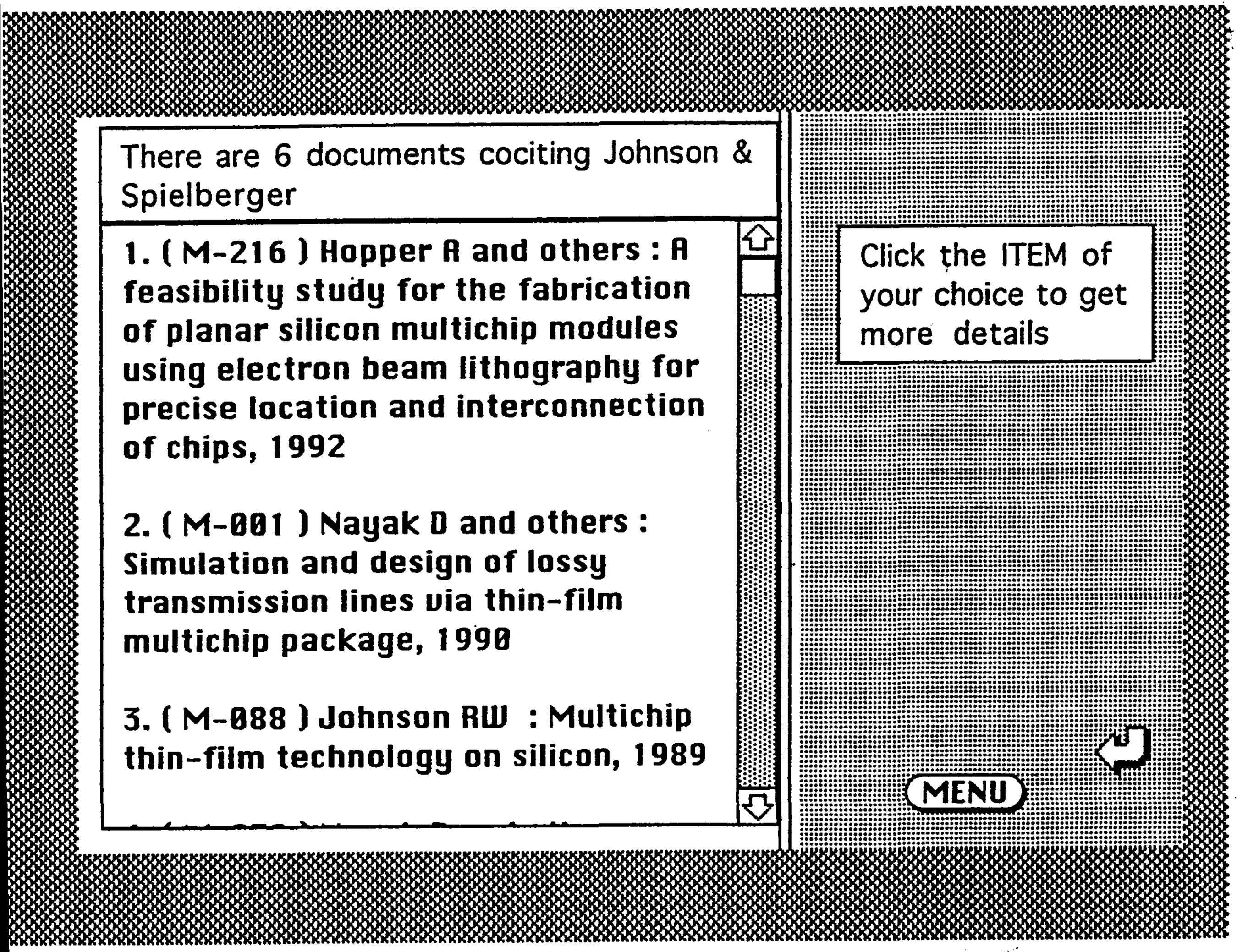
LIST OF CITED DOCUMENTS (With No. of Citations for Each) FOR COCITATION SEARCH

79. (M-063) Jensen RJ and others : Copper/polyimide materials system for high-performance packaging, 1984 _4

80. (M-004) Johnson RW and others : Silicon hybrid wafer-scale package technology, 1986 _6

81. (M-075) Johnson RW and others : Thin-film silicon multichip technology, 1988 _2

FIRST CLICK TWO ITEMS OF CHOICE IN THE LIST AND THEN CLICK HERE



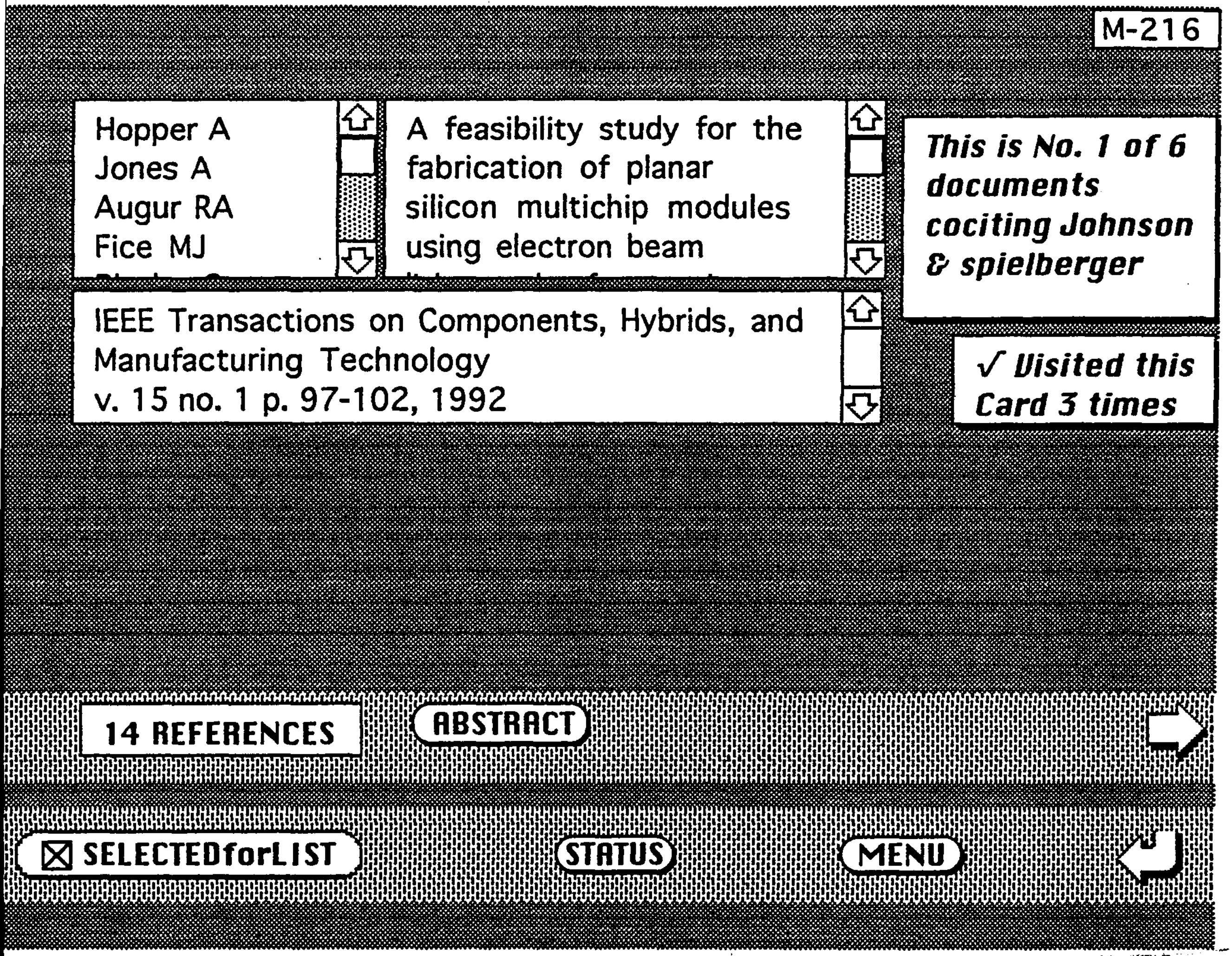
There are 6 documents cociting Johnson & Spielberger

1. (M-216) Hopper A and others : A feasibility study for the fabrication of planar silicon multichip modules using electron beam lithography for precise location and interconnection of chips, 1992

2. (M-001) Nayak D and others : Simulation and design of lossy transmission lines via thin-film multichip package, 1990

3. (M-088) Johnson RW : Multichip thin-film technology on silicon, 1989

Click the ITEM of your choice to get more details



M-216

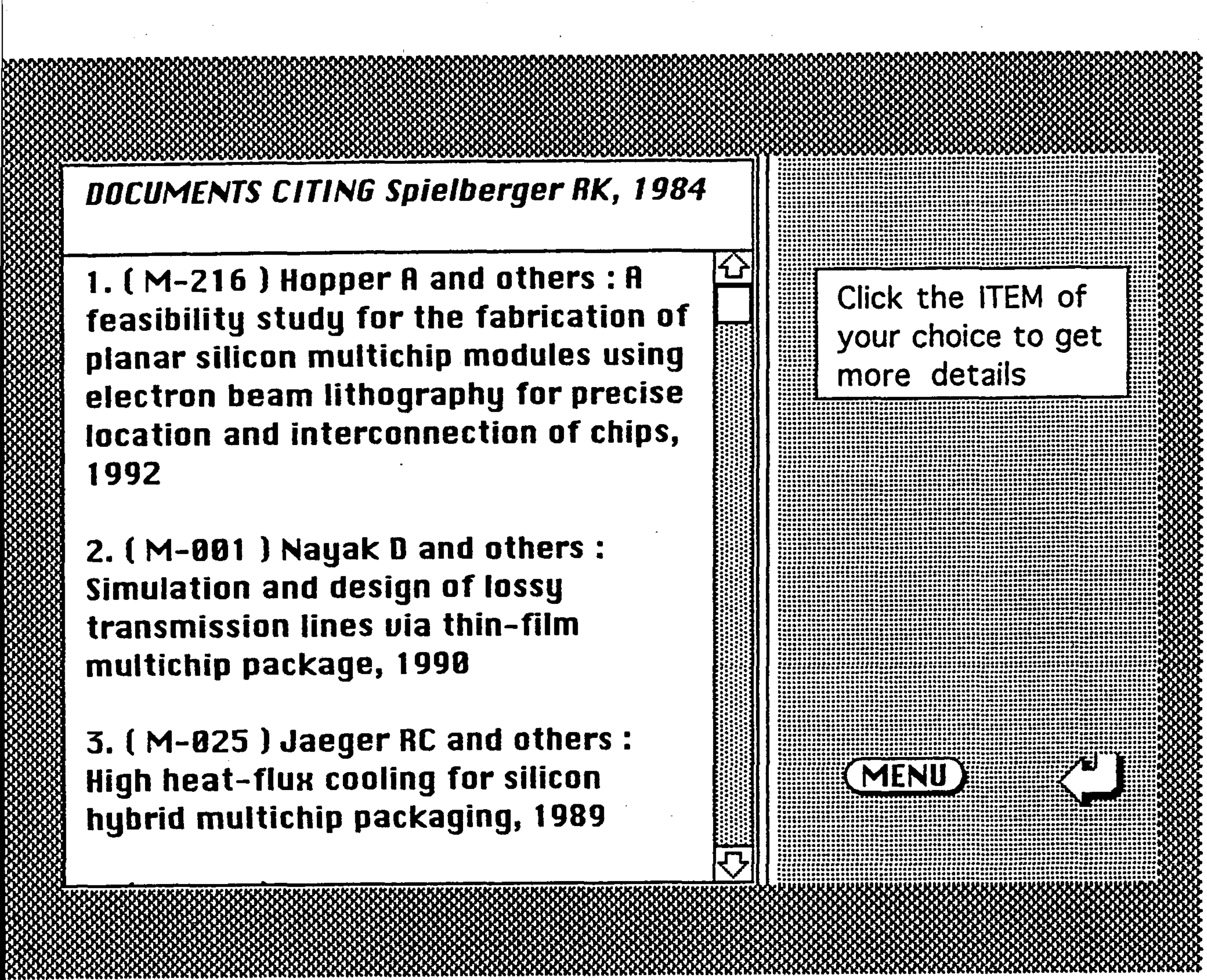
Hopper A
Jones A
Augur RA
Fice MJ

A feasibility study for the fabrication of planar silicon multichip modules using electron beam

IEEE Transactions on Components, Hybrids, and Manufacturing Technology
v. 15 no. 1 p. 97-102, 1992

This is No. 1 of 6 documents cociting Johnson & spielberger

✓ Visited this Card 3 times



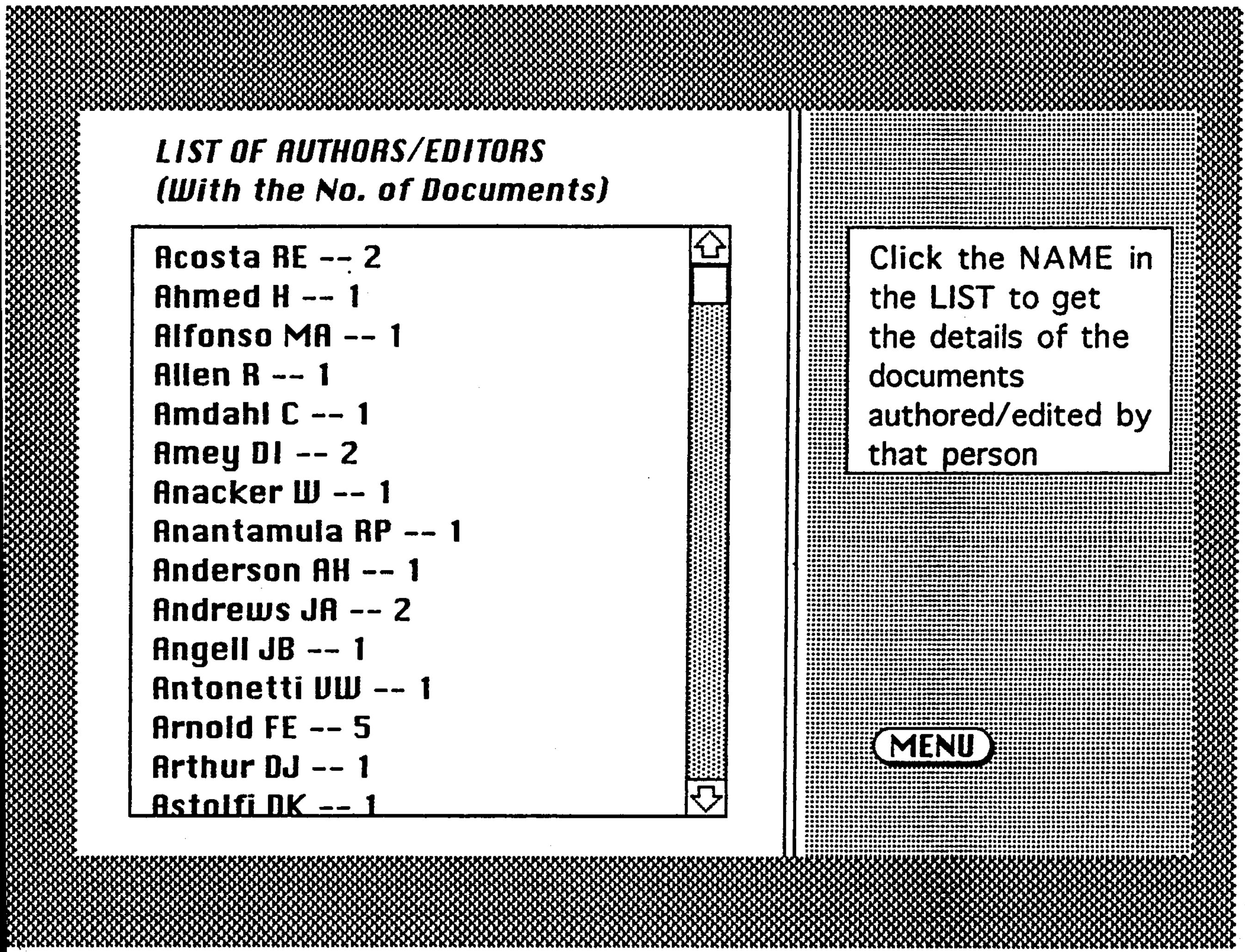
DOCUMENTS CITING Spielberger RK, 1984

1. (M-216) Hopper A and others : A feasibility study for the fabrication of planar silicon multichip modules using electron beam lithography for precise location and interconnection of chips, 1992

2. (M-001) Nayak D and others : Simulation and design of lossy transmission lines via thin-film multichip package, 1990

3. (M-025) Jaeger RC and others : High heat-flux cooling for silicon hybrid multichip packaging, 1989

Click the ITEM of your choice to get more details



LIST OF AUTHORS/EDITORS
(With the No. of Documents)

Acosta RE -- 2
Ahmed H -- 1
Alfonso MA -- 1
Allen R -- 1
Amdahl C -- 1
Amey DI -- 2
Anacker W -- 1
Anantamula RP -- 1
Anderson AH -- 1
Andrews JA -- 2
Angell JB -- 1
Antonetti VW -- 1
Arnold FE -- 5
Arthur DJ -- 1
Astolfi DK -- 1

Click the NAME in the LIST to get the details of the documents authored/edited by that person

IST OF DOCUMENTS SELECTED

. (M-003) Bartlett CJ and others : Multichip packaging design for VLSI based systems. 37th
ectronic Components Conference, 1987. Proceedings. p. 518-525, 1987
-Not available in this library

(M-216) Hopper A and others : A feasibility study for the fabrication of planar silicon multichip
odules using electron beam lithography for precise location and interconnection of chips. IEEE
ansactions on Components, Hybrids, and Manufacturing Technology. v. 15 no. 1 p. 97-102, 1992
-Location: SERIALS 621.3 IEEE (Now on loan)

(M-001) Nayak D and others : Simulation and design of lossy transmission lines via thin-film
ltichip package. IEEE Transactions on Components, Hybrids, and Manufacturing Technology. v. 13,
. 2, p. 294-302, 1990
-Location: SERIALS 621.3 IEEE