*Information Technology: Research and Development* (1983), *2* (65–72)

# DESIGNING AND IMPLEMENTING A MICROCOMPUTER TRAINING PACKAGE FOR ONLINE BIBLIOGRAPHIC SEARCHING*

C. J. ARMSTRONG AND J. A. LARGE

*College of Librarianship Wales, Aberystwyth*

(*Received 11 January 1983*)

## ABSTRACT

The reasons for developing a microcomputer emulation to train online searchers are discussed along with the major problems which have to be surmounted. The design of such an emulation at the College of Librarianship Wales is described.

## 1. INTRODUCTION

Initial training in the use of online retrieval systems can be provided on the commercial systems which will ultimately be used in the library or information unit, but this has a number of drawbacks.

1. The costs, which are almost directly related to connect time, are high, so that the inexperienced searcher who hesitates and makes typing mistakes is penalized.
2. The pressures of live searching, with the attendant costs, affect all but the most confident trainees.
3. Training sessions must often take place within precise and pre-ordained periods of time. An inability to log on to the system, for whatever reason, disrupts the schedule, and it may be impossible to repeat the session on another occasion.

These factors have persuaded organizations, and particularly library schools, that there is much to be gained from developing teaching aids which will introduce trainee searchers to the keyboard, the command language and search-strategy construction. In-house experience may be gained before live searching is attempted, by the provision of such training aids, and a useful summary of some of these has been compiled by Tedd (1981). Caruso (1981a) has compiled a summary of Computer-Aided Learning (CAL) packages, some of which emulate systems such as DIALOG or SDC, for example, the mainframe-based TRAINER (Caruso, 1981b) and DIATOM (Waldstein, 1981). Until recently microcomputer training has been

* The work described here was completed under a grant from the British Library Research and Development Department.

limited to systems such as FOSSILS (Wood, 1981) or DIALOG with PET (Vickery, 1980) which simulate or demonstrate the large information retrieval systems, and only two truly heuristic microcomputer training aids have been produced—by Aslib (Payne, 1981) and Loughborough (Johnson, 1980). The Aslib system is a working information retrieval system which uses a subset of the European Common Command language against a database of materials held in-house. Record structures and responses are dissimilar to any real system and thus it does not really count as an emulation. It was designed for use in schools and the educational aims differ from ours. The Apple simulator at Loughborough works on a very small database of under 40 records, and has only limited capabilities; it was, in fact, originally written for a minicomputer, and the adaptation to a single 5¼ in. disk microcomputer was never very satisfactory. This paper describes the methodology for creating a true microcomputer emulation and discusses the problems which might be encountered. The emulation is only part of the teaching package, which also includes a CAL module to introduce students to online commands and is described in full in Large and Armstrong (1983).

## 2. THE PHILOSOPHY

Our first decision was that the emulation should be designed to operate on a microcomputer rather than a mini- or a mainframe computer. This was influenced by the fact that the College was acquiring microcomputers for teaching purposes, but we also believed that a micro-emulation offered a number of advantages. Certainly, the use of a microcomputer-based system seemed to offer considerable benefits in terms of portability and thus in terms of the potential use of the final package. When considerable funds have been expended on the development of a teaching aid, it is clearly necessary for these to be offset against many users (Caruso, 1981a). Work done on a mainframe or on a minicomputer offers a number of advantages, such as unlimited database size, fast response times and access from multiple ports. On the other hand, it may be limited to the parent institution. Portability can be reduced by the operating system or the programming language (for example, the emulation DIATOM is written in SAIL on a DEC-10 computer operating under TOPS10) and possibly by the database and program size. Further, many more libraries and library schools are unable to afford their own mini- or mainframe computer, and at best only have a line to an external computer, which can create problems of access and control. On the other hand, microcomputers are relatively cheap—within the grasp of most institutions—and can be dedicated, if necessary, to training. If a microcomputer were used which operated under CP/M, the *de facto* operating system, portability and a wide potential would seem to be assured.

However, the use of a microcomputer would clearly impose a number of problems. The maximum memory which could be addressed at the time of planning the project was 64 Kbytes, and the mandatory 8 bit operating system was slow compared with mini- and mainframe computers. Although hard disks were beginning to be added to microcomputers, they were the exception rather than the rule, and to include one in the system would have considerably detracted from its portability. The College of Librarianship Wales at that time owned a Research Machines 380Z microcomputer with 48 Kbytes of main memory and twin 8 in. floppy disk drives, which together gave just under one Mbyte of backing store. As this was a CP/M-based system, work was begun using this machine.

It was decided to base the entire package on DIALOG and to make our microdatabases subsets of ERIC. Both DIALOG and ERIC are universally popular for tuition, as the command language is sophisticated and easy to use, while the database has a wide subject coverage and is not limited to one discipline such as chemistry or agriculture. Perhaps it is for historical reasons that library schools have preferred DIALOG, but Williams (1981) has indicated that, despite more immediate access to databases in Europe, users have reverted to the American products.

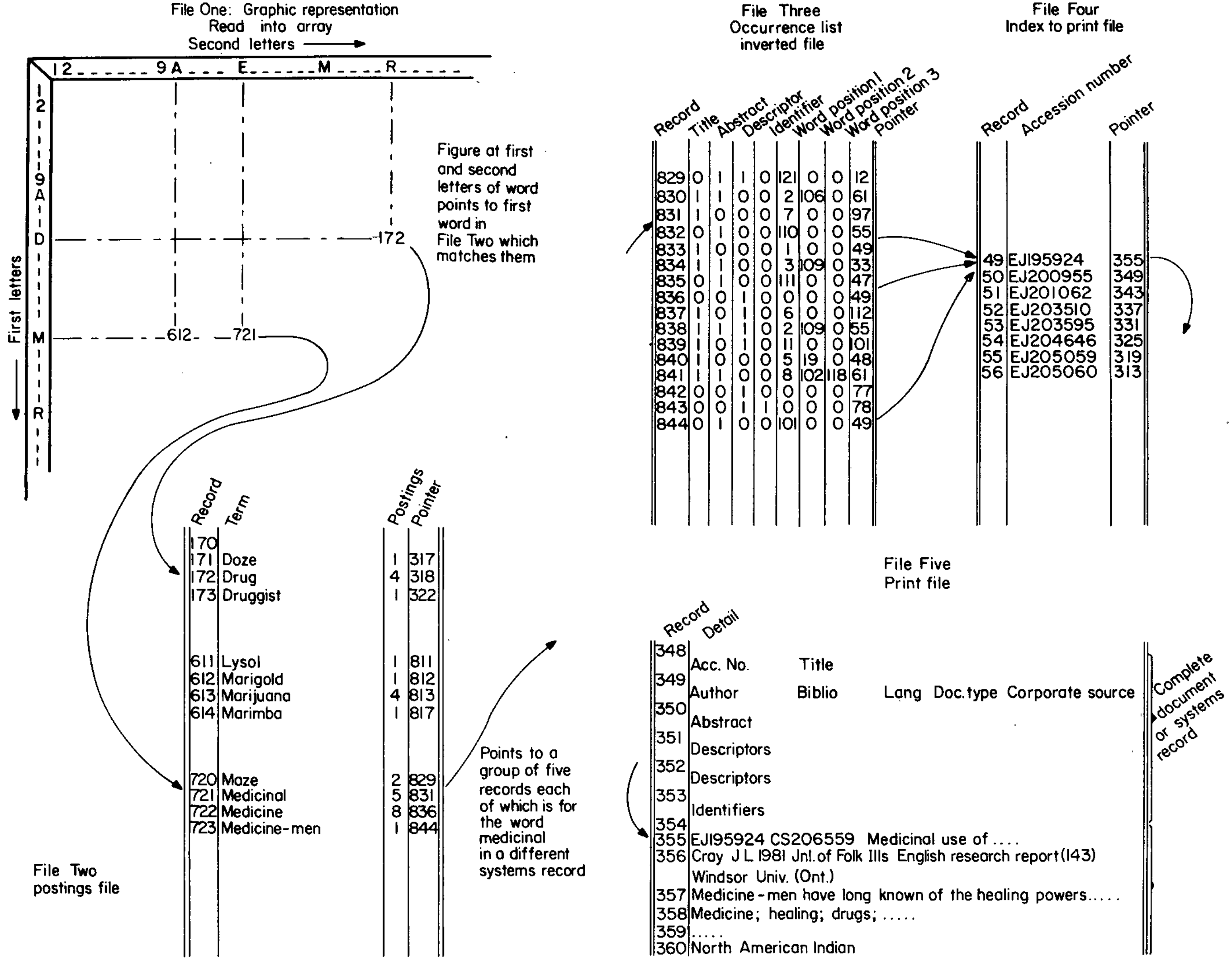
We felt that, if any in-house training of this kind is to be successful, then it is important that the emulation should appear in every way like the real thing: commands would be accepted in any form in which they would be accepted online, and they would elicit a response which is an exact duplicate of that on the real system. Records, when they are typed out, would appear in the correct format and style, and all the major commands and facilities would be present in the emulation. Thus we distinguish between a simulation, or a model, and an emulation. The former copies, while the latter reproduces exactly and may thus be used to introduce students to a system and its operation (or idiosyncrasies). A simulation could only be used to introduce command capabilities, *in vacuo*, or to demonstrate a pre-defined search strategy. Anyone searching on the emulation, on the other hand, would only differentiate it from its exemplar by the smaller set sizes, in some cases the slower response time, and our additional error messages. Additional advantages could be foreseen, in that the interaction with a local computer system would do much to dispel student apprehension when confronted with a terminal linked to a remote computer. Often worries are expressed about damaging the system by pressing the wrong key! Further, it was decided that if the emulation could work through a terminal, as opposed to the microcomputer keyboard, the familiarity with the terminal keyboard would stand the student in good stead when it is ultimately used online.

In order to maximize the potential use of any teaching package, a wide spectrum of users must be catered for, and thus the package should not be dependent on external tuition or help, but should be as user-friendly and self-contained as possible.

## 3. HARDWARE AND SOFTWARE IMPLICATIONS

From the outset, it was clear that disk space would be at a premium, and calculations and experiments soon made it clear that with a microdatabase of between 100 and 120 records, the disk space left after the programs were stored would nearly all be used up. Since files under CP/M cannot be divided between 2 disks, or even 2 disk sides, the limit of any one file was about 250 Kbytes. Although 100 system records, each containing 6 disk records of 255 characters, only use about 170 Kbytes, the associated inverted files use 2 complete disk sides between them. Our original specification required the full facilities found on the DIALOG system to be copied, and a file and record structure was designed to allow for all these facilities, including full text searching—the most demanding in terms of file space (Fig. 1). This was then modified as calculations demonstrated the files' potential to use up disk space. At the outset, 100 to 150 records had seemed a reasonable target—sufficient to demand the formulation of coherent search strategies, but not so many as to overfill the disks or produce unreasonable response times. In the event, the optimum figure appears to be 110.

In order to achieve this number, it was necessary to reduce record content in the print file, and this in turn meant a slight reduction in capabilities. The 6 record


File One: Graphic representation
Read into array
Second letters
12 9 A E M R
First letters
1 2 9 A D M R
172
612
721
Figure at first and second letters of word points to first word in File Two which matches them
File Two
postings file
Points to a group of five records each of which is for the word medicinal in a different systems record
File Three
Occurrence list
inverted file
File Four
Index to print file
File Five
Print file
Complete document or systems record

File Two postings file

| Record | Term | Postings | Pointer |
|---|---|---|---|
| 170 | | | |
| 171 | Doze | 1 | 317 |
| 172 | Drug | 4 | 318 |
| 173 | Druggist | 1 | 322 |
| 611 | Lysol | 1 | 811 |
| 612 | Marigold | 1 | 812 |
| 613 | Marijuana | 4 | 813 |
| 614 | Marimba | 1 | 817 |
| 720 | Maze | 2 | 829 |
| 721 | Medicinal | 5 | 831 |
| 722 | Medicine | 8 | 836 |
| 723 | Medicine-men | 1 | 844 |

File Three Occurrence list inverted file

| Record | Title | Abstract | Descriptor | Identifier | Word position 1 | Word position 2 | Word position 3 | Pointer |
|---|---|---|---|---|---|---|---|---|
| 829 | 0 | 1 | 1 | 0 | 121 | 0 | 0 | 12 |
| 830 | 1 | 1 | 0 | 0 | 2 | 106 | 0 | 61 |
| 831 | 1 | 0 | 0 | 0 | 7 | 0 | 0 | 97 |
| 832 | 0 | 1 | 0 | 0 | 110 | 0 | 0 | 55 |
| 833 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 49 |
| 834 | 1 | 1 | 0 | 0 | 3 | 109 | 0 | 33 |
| 835 | 0 | 1 | 0 | 0 | 111 | 0 | 0 | 47 |
| 836 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 49 |
| 837 | 1 | 0 | 1 | 0 | 6 | 0 | 0 | 112 |
| 838 | 1 | 1 | 1 | 0 | 2 | 109 | 0 | 55 |
| 839 | 1 | 0 | 1 | 0 | 11 | 0 | 0 | 101 |
| 840 | 1 | 0 | 0 | 0 | 5 | 19 | 0 | 48 |
| 841 | 1 | 1 | 0 | 0 | 8 | 102 | 118 | 61 |
| 842 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 77 |
| 843 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 78 |
| 844 | 0 | 1 | 0 | 0 | 101 | 0 | 0 | 49 |

File Four Index to print file

| Record | Accession number | Pointer |
|---|---|---|
| 49 | EJ195924 | 355 |
| 50 | EJ200955 | 349 |
| 51 | EJ201062 | 343 |
| 52 | EJ203510 | 337 |
| 53 | EJ203595 | 331 |
| 54 | EJ204646 | 325 |
| 55 | EJ205059 | 319 |
| 56 | EJ205060 | 313 |

File Five Print file

| Record | Detail |
|---|---|
| 348 | Acc. No. Title |
| 349 | Author Biblio Lang Doc.type Corporate source |
| 350 | Abstract |
| 351 | Descriptors |
| 352 | Descriptors |
| 353 | Identifiers |
| 354 | |
| 355 | EJ195924 CS206559 Medicinal use of .... |
| 356 | Cray J L 1981 Jnl.of Folk Ills English research report(143)<br>Windsor Univ. (Ont.) |
| 357 | Medicine-men have long known of the healing powers..... |
| 358 | Medicine; healing; drugs; ..... |
| 359 | ..... |
| 360 | North American Indian |

FIG. 1 File structure

systems record (Fig. 2) now contains sufficient fields and data to maintain the verisimilitude and capabilities of the emulation at an acceptable level. Original records which contain more than one author have all authors beyond the first removed. In addition, all abstracts are reduced to the first punctuation prior to 255 characters. In the case of records relating to journal articles, this is normally sufficient to include the original full abstract; abstracts from report documents, however, are usually reduced. This author and abstract editing is carried out automatically by the validation program. All descriptors and identifiers are retained, and additional indexes are maintained for publication year, author, corporate source and language. Other common additional indexes—document type, journal name and update—are not available, but it is felt that the four which have been included are sufficient to demonstrate the system's capabilities. In addition to allowing DIALOG-like operations, the files had to permit reasonable response times. As the postings must reflect the documents in the microdatabase, and are therefore much smaller than on DIALOG, it is unreasonable to have response times which are vastly in excess of those on DIALOG. For this reason, the complex multi-part file structure shown in Figure 1 is necessary on the emulation. At every stage, file and record structures constantly had to be weighed against capabilities, until a balance could be achieved which left sufficient records in the microdatabase. For instance, the inclusion of an extra disk record to double the length of the stored abstract would be perfectly possible in terms of the print file; however, it would generate an additional 20 to 30 records in the occurrence list for each of the 100 or more system records. As the occurrence file uses a complete disk side, this in turn could only be achieved by reducing its capability to store only two word positions in the title or abstract, as opposed to three.

| | | | | | |
|---|---|---|---|---|---|
| Accession number 18 | Title 235 characters | | | | |
| Author 43 | Year 4 | Bibliographic detail 76 | Lang. 20 | Doc. type 30 | Corporate source 80 characters |
| Abstract 253 characters | | | | | |
| Descriptors 253 characters | | | | | |
| Descriptors 253 characters | | | | | |
| Identifiers 253 characters | | | | | |

FIG. 2. The 6 record systems record

ERIC on DIALOG provides an online thesaurus lookup, but memory limitations prevented its inclusion on the emulation. This was unfortunate, but early calculations showed that the requisite file structure was impossible.

## 4. MICRODATABASE GENERATION

The microdatabase is generated from a subset of the ERIC database. Rather than select a small number of records at random from ERIC, it was decided to select a topic around which the microdatabase could be constructed. Our first database is on 'Television and violence'. The records are taped in full format (format 5) on a Hewlett Packard HP2645A with twin tape drives, and then read on to the Research Machines 380Z. A program identifies the individual fields in each record, and edits the author and abstract fields if necessary, as described above. If the edited abstract is considered unsatisfactory, it can be discarded and a new abstract entered manually. Once each record has been checked at the VDU, the database creation program automatically generates the inverted files (this takes around 16 hours). The database creation procedure has been automated as much as possible, to reduce the need for manual input and eliminate errors. As many microdatabases as required can be generated in this way, as long as a new pair of disks is used for each database.

It was our original intention to generate microdatabases from a series of DIALOG databases. ERIC was selected first because it is a very popular database for training purposes in both British and American library schools, and offers most database facilities on DIALOG. Unfortunately, we had to abandon plans to extend our microdatabases to the other databases, because the record structures differ so radically. The emulation program which recognizes fields within the record was designed to deal with ERIC records, and would need to be rewritten for other record structures. Furthermore, not all DIALOG commands work in the same way on all databases; the method of locating documents according to language, for example, is not uniform across all DIALOG databases.

## 5. EMULATION DIAGNOSTICS

As the package specification required a stand-alone teaching aid, which could be used without the presence of a tutor, the systems analysis included an umbrella guidance module which monitored the search progress, the use of commands and the ongoing strategy, and volunteered advice as necessary. As the emulation is completely heuristic and commands may be used in any sequence, any judgement made on a search strategy requires considerable artificial intelligence to be built into the system with consequential high overheads in terms of memory. Thus the implementation of this module has proved largely impracticable, because of memory limitations (RAM and disk), as well as slow file access. A limited series of help messages has been included, however, most of which deal with semantic or logical errors in command use. These either enhance the DIALOG error messages, or, more usually, warn the user of strategic faults. The failure to use BEGIN at the commencement of a search or SELECTing a term which has appeared in an EXPAND array, for example, will both produce a warning, although in the first case the command is aborted and in the second case it continues and produces a new set.

## 6. MEMORY PROBLEMS

Problems of mounting such a major system on a microcomputer centre on the effective use of the available memory. The usual approach to this is to chain a number of small program modules together so that, at any one time, the majority of the memory is free for data processing. For reasons of portability, the best CP/M BASIC language then available (CBASIC) was used; unfortunately, however, this closes all files every time a fresh module is invoked and, as a full second is required to re-open each file, this approach is inappropriate for an emulation in which response time is important. It was decided to produce a single unchained program in the first instance, and then to use this prototype to experiment with the data processing in limited memory. By the time that about 80 per cent of the commands were functioning, the program had already expanded to use all the available 48K. As it was possible to expand the RML380Z to 56K—a figure more in keeping with other microcomputers on the market—this was adopted as the first remedy. However, it was obvious that this would not be a sufficient increase to accommodate the remaining commands and to allow even the lowest level of diagnostic help to function, so other savings had to be made. Some sections of the code—the COMBINE logic, the word-proximity comparison and the initial command recognition—particularly lent themselves to machine code, and these were rewritten. Initially we had tried to avoid the use of machine code as it makes between-machine portability more difficult, but it became inevitable. Finally, two sections of the program which dealt with two distinct commands were separated and chained, as was the early code which dimensions and assigns the variables. It was with some reluctance that these steps were taken, as it meant that the two commands, EXPAND and TYPE, would operate relatively more slowly, and although the machine code did not prohibit the package's removal to other machines, it did make it more difficult. These two commands were chosen as they have limited file use in themselves and thus the delay appears after the response and before the next prompt, while the main program files are being re-opened. As both commands produce responses which require some examination on the part of the searcher before continuing, this was more acceptable. The final system (Fig. 3) represents the best solution possible, given the constraints of hardware and programming language at the time.

## 7. THE FUTURE

When the emulation module was originally conceived, hard disks were still relatively uncommon, and it was considered that to use hard disks instead of floppy disks

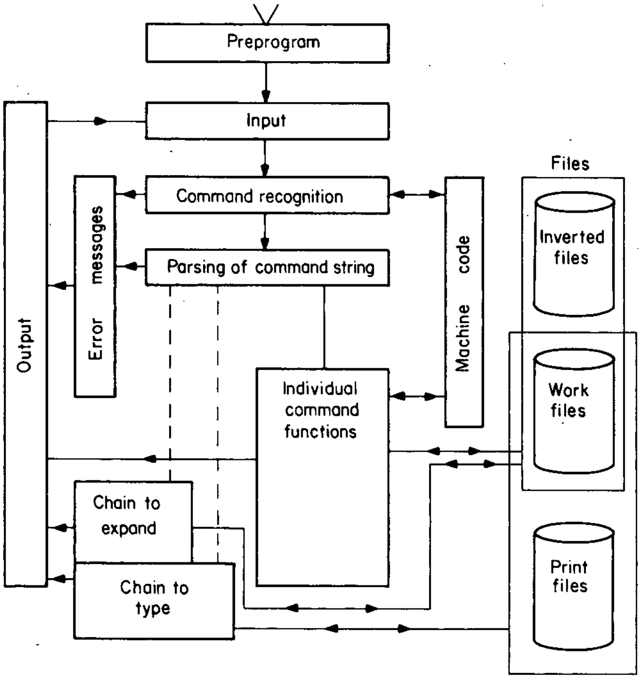


FIG. 3. Overview of emulation system

would therefore have reduced the portability of the programs. In the event, hard disk systems have now become much more widespread, and their much greater storage capacity would enable each microdatabase to be increased to around 600 records. Furthermore, response time would be improved with hard disks. Another significant development has been the increasing size of random access memory available on microcomputers. As RAM size posed considerable difficulties, the use of a microcomputer with 256K RAM, instead of 56K as on our RML 380Z, would enable us to implement the outstanding DIALOG commands and add more sophisticated diagnostics as well as the thesaurus. It is our hope that we shall be able to continue our work along these lines.

It is still too early to draw firm conclusions about the value of the emulation as a training aid. It is only now beginning to be used by students, although initial response is certainly favourable. Online searching can be practised for as long as necessary on the emulation at no cost, enabling the student to grasp the essentials of searching strategies. It is not claimed that the emulation replaces genuine online searching, but rather that it enables expensive online time to be maximized by restricting its use to those training aspects which cannot be mastered on the emulation. In particular, practice at handling very large sets cannot be acquired on the emulation.

## REFERENCES

Caruso, E. (1981a) Computer aids to learning online retrieval. *The Annual Review of Information Science and Technology*, Vol. 16. (M. E. Williams, ed.) pp. 317–335. New York: Knowledge Industry Publications.

Caruso, E. (1981b) TRAINER. *Online 5*, 36–38.

Johnson, D. K. (1980) *Report on the conversion of an online information retrieval system simulator for use on an ITT 2020 (Apple) microcomputer.* London: British Library Research and Development Department (BLR&D Report 5580).

Large, J. A. and Armstrong, C. J. (1983) The microcomputer as a training aid for online searching. *Online Review 7*, 51–59.

Payne, A. (1981) Online information retrieval in schools. *CAL News 16*, 11.

Tedd, L. A. (1981) Teaching aids developed and used for education and training for online searching. *Online Review 5*, 205–216.

Vickery, A. (1980) The CIS software series. Available from: the author, Central Information Service, Room 504, Senate House, University of London, Malet Street, London WC1E 7HU, UK.

Waldstein, R. (1981) DIATOM—A DIALOG simulator. *Online 5*, 68–72.

Williams, P. W. (1981) *NCC Survey of online systems.* Manchester: The National Computing Centre.

Wood, F. E. (1981) Online teaching aids from the Department of Information Science, University of Sheffield, England. *Online Review 5*, 487–494.