A SIMPLE COMPUTER-BASED SOLUTION TO PROJECT BIBLIOGRAPHIES

Aubrey Fricker
Geological Survey of Canada
A.G.C., Bedford Institute, Box 1006
Dartmouth, Nova Scotia B2Y 4A2

Audrey Samson
Biblio-Tech Ltd.
R.R. #2 Head of Chezzetcook
Three Fathom Harbour, Nova Scotia B3B 1K8

ABSTRACT

Though there is much commercial software on the market for permanent library functions, these packages are generally not suited to short-term project support. The requirements are simplicity, speed of creation, speed of access and the flexibility to support access and reporting operations which are specific to the project. This can be achieved using a generalised database management package and minor programming. The system described here took about 3 weeks work from an analyst and a few days from the librarian. It could be set up and adapted to different operations in a day or two. The mechanical aspects of loading and checking almost 800 documents required about 5 weeks of librarian and clerical time. Thus, the system was easily able to keep pace with the project.

RESUME

On trouve, sur le marché, de nombreux logiciels conçus pour les tâches permanentes de bibliothèque; toutefois, ces produits ne se prêtent habituellement pas aux travaux à court terme. Un logiciel conçu pour ce genre de travail doit satisfaire aux exigences suivantes: simplicité, rapidité de création, rapidité d'accès et suffisamment de souplesse pour permettre des opérations d'accès et de compte rendu particulières à chaque projet. Pour réaliser ce genre de logiciel, on peut adapter un progiciel général de gestion d'une base de données, ce qui ne nécessite qu'un léger travail de programmation. Le système décrit dans le présent rapport a nécessité trois semaines de travail de la part d'un analyste et quelques jours de travail de la part d'une bibliothécaire. Il peut se monter et s'adapter à diverses opérations en un jour ou deux. La bibliothécaire et une commis ont mis environ cinq semaines à entrer en mémoire et à vérifier les références de près de 800 documents, ce qui montre bien que le système s'est révélé tout à fait satisfaisant.

PROJECT BIBLIOGRAPHIES

THE PROJECT

A professional librarian had the task of providing information support to a multidisciplinary group of scientists working on a joint project, at the Bedford Institute of Oceanography in Dartmouth, Nova Scotia. She was required to:

1- Perform manual and automated literature searches to identify relevant material,

2- Obtain copies of these documents and others identified by the scientists involved,

3- Keep bibliographic records of the material assembled.

The first two tasks were made manageable by the help of the Institute's permanent library staff and an assistant helping with the large amount of photocopying. The third task was initially managed using a simple card file and the memory of the librarian. These became inadequate as the file increased, as records were revised, and as the cards began to deteriorate.

THE PROBLEM

For these reasons, we considered automating. Specifically, these are the objectives we had to meet:

1- The file had to carry all bibliographic information needed to accurately identify a particular document;

2- In addition to the essential bibliographic information there was information on which scientist had requested it, or in which draft of which section of the scientific report it had been cited. This information needed to be frequently updated when the same material was cited by more than one individual;

3- The card file containing about 500 records, which existed before automation, had to be entered all at once. The system had to keep pace with the project on a more or less daily basis, since it was the only record of all the material assembled (other than the file of material itself). Any aspect of any record could have been changed, and records could have been added (or occasionally removed) at any time;

4- Copies of some papers were provided to a central co-ordinating body in another city. The system had to keep track of which papers had already been sent, which were to be sent, which were not going to be provided and why, and which had not yet been requested.

5- The librarians's notes, usually about location or availability of an item, or material designation (ie. microfiche) had to accompany the bibliographic record.

6- We anticipated being required to retrieve records on a variety of criteria:

author;
year;
subject;
citor or requestor;
status with regard to the reviewing body (ie. sent/not sent),

7- We could have to organise retrieved records in a variety of fashions although alphabetically by author, then date, then title (the standard sequence in bibliographies) was considered to be the most likely. We had to be able to specify how much of a record was to be printed in any listing. Output would probably be as close to a standard bibliographic citation as possible, but it might sometimes have been necessary to produce output which included information on status with regard to the reviewing body, who cited the record, and any notes made by the librarian,

8- The day-to-day operation of the system was the responsibility of the librarian and her assistant. The librarian, while familiar with on-line searching and other automated databases, was certainly not a programmer. The system had to have commands simple and straightforward enough for her to quickly grasp, while she was still a new user.

OPTIONS

Although it initially appeared that a number of different options for implementation were available to the project, most of the options were quickly discarded. The system was implemented using SYSTEM 2000, the general database management system running on the Institute's mainframe Cyber. Other options included Keyword in Context programs, the purchase of library software packages, and the in-house programming of library-type packages.

The attraction of SYSTEM 2000 was chiefly the reduced software development time and the reliability of a well-supported package. The problems which arose in this project included dealing with variable length text fields, formulating iterative search strategies, and dealing with the complexity of a DBMS query language.

Author, title and citor fields were variable length text fields for which it was necessary to be able to search on subfields. In order to accomplish this, software was written which extracted keywords from

the title and author names from the author and citor fields during the loading of the database. This software interfaced with SYSTEM 2000 to create the appropriate indexes to the documents.

The retrieval system implemented for the project was intended for use by trained personnel rather than by uninitiated users. Therefore, the system could be based on a command language rather than on a "user-friendly" menu-type query language.

THE DATABASE

There were eight items recognised in the information on the original card system: author(s); the standard role abbreviations, ie. "ed.", "tr."; date; title; publishing information; status with regard to the reviewing body; citor; librarian's notes. In all such systems an entry key is both theoretically and pragmatically necessary. A unique integer number is good enough for the job, and is easiest to program. This unique integer key and the status of the document could be automatically initialised, the status initially set as "not yet requested".

We had to distinguish between simple and compound items. Here, we use these two terms to describe the difference between "atomic" items whose possible values are each a string of characters but are always used as a whole, and "molecular" items which are stored as a unit but for certain functions have to be parsed for substrings which they contain. Three of the eight items were simple, essentially one-word.

The authorship of a document when stored in the database had to preserve the order of names as they appeared on the original document, and in accordance with accepted bibliographic conventions. Bibliographic output could be sorted by the whole sequence of authors names. However, selection by individual author was required. Similarly, the title had to be stored exactly as it appeared on the document, but selection was based on significant words (keywords) in the title. These keywords were used for searching in any combination.

The requirement for the compound items was thus to recognise some words for certain operations. What is more, some of these sub-items could have had an intermediate compound item, such as double surnames for authors, and similar compounds within the title. There was thus a minor "grammar" or syntax involved in each of the compound items.

Decisions such as that to treat all the publishing information as a single item might seem radical to many librarians, but are an essential part of the analysis for project support. It is the ability to do this that results in a flexible but powerful tool. It makes the operational

work considerably more straightforward. Strict rules for operation must be avoided in favour of simple tools which allow human judgement to control and fill any gaps in the system. The analyst was several times asked to tune the operation and generally did this in about 1/2 hour.

PERFORMANCE

Preparation, addition, checking and updating were the tasks to be performed on a daily basis. Rough estimates indicated that about 100 documents could be prepared for entry in one day's work, and this was not far from our experience. Even allowing for human variability, at this level the system could easily keep up.

Reporting requirements in this project were most uncertain because several people and agencies could request output for different purposes, many of these unforseen at the outset of the project. Hence more flexibility was needed in selecting, ordering, and formatting records than is available in most packaged software. The time-frame could be typically one day's notice. We decided that the librarian, or some other trained database manager, would always act as intermediary between information requesters and the database. In fact, the project researchers did not even need to know of the existence of the database, as long as they could be provided with the material and citations they require. The database was a tool that enabled the librarian to do this.

The cost of the computer support was not a significant performance consideration. If it were translated into dollars it would be in the hundreds, which was certainly less than the salary cost. One document took about one central-processor second to load, on a CDC Cyber 171-6. The total processing was about an order of magnitude greater. The total size of the file (SYSTEM 2000 file 6) was just over 100,000 characters. At this size, interactive response in SYSTEM 2000 is usually reasonable in most mainframe installations.

PROGRAMMING

The software available was SYSTEM 2000. Since the work had to be done on a CDC machine, control language procedures for initiation at a terminal constituted 25% of the written code. 60% of the code was ANSII 77 Fortran, and 15% was SYSTEM 2000 instructions. Therefore, the 411 lines of code (not including comments) should be fairly easy to adapt to another site. We will refer to the SYSTEM 2000 interactive access language as Natural Language. It would have an equivalent in most generalised database packages.

The first step taken in such a project should be to draft the user manual. Since our project was small, this eliminated the need for

analysis and design documents. The revision of the manual constituted all the project control necessary. The basic operations, and the procedures to perform them became settled through review of the manual.

Creating the file structure should be the second step. Our information was very simple. It could have been stored in a "flat" file, such as a card index or a single record format in a computer. Furthermore, the report requirements were easy to program. The part that was special in this application was in the retrieval and sorting needs. They required reorganisation of the information in the fields concerned. Since the original form of that information had also to be preserved, the same information had to be stored in two places. Such redundancy cannot be avoided if maximum flexibility of output reporting is to be achieved.

The items on which selection and ordering were based were authorship of the documents, the content as expressed by the title, and the identity of the researchers who cited the document and required its record in the system. Existing software systems also use such items, but tend to be inflexible in the functional design. The items were extracted during the loading process, to set up three indexes. The database manager had the opportunity to affect retrieval by editing these indexes. For example, American spelling might have been preferred in the indexes without altering the spelling of these words in the title field.

The third step should be drawing up a format for routine loading. Since our information was simple, and many staff were happy with the computer line-editor, a format resembling the layout of index cards was convenient. The major concession to programming was the addition of a line-classifying character at the beginning of each line. The only other consideration related to continuation from line to line.

The fourth and final step should be coding the computer procedures. Four operations were identified. Two of the four recognised operations were direct access to the database. One was updating, the other was retrieval. The other retrieval procedure initiated off-line printing of bibliographic citations for the entire file, in as close a format as possible to a standard bibliography. The reporting requirement remained a simple Master Listing operation. It could obviously be easily expanded.

The major coding effort went into the loading operation. The algorithms for creating the index records were a compromise. Some simple parsing principles, and simple "stop" lists went a long way. The real complications arose in the useage of special characters for punctuation. It was not always clear when an item should be a candidate for the index. By erring on the side of putting too much in, the librarian could refine the indexing quite quickly.

We reviewed the system from time-to-time to reduce the amount of code if possible. A level of quantitative reliability was not specified, but our experience was that other than time spent on early tests and trials, probably only a few hours work were spent correcting errors arising from the system. Significantly more time was lost from the usual computer "downs" and human errors in saving of information.

OPERATION

Careful proofreading of input files should ensure that incorrect records are not loaded. If mistakes were discovered after a record was loaded, they could be corrected using the Natural Language edit commands. Editing records after they were loaded could be cumbersome and risky. A backup of the database was always done before any editing was undertaken.

Assuming records were entered correctly, elements in them still needed to be updated as the project progressed. For example, if an item were sent to the reviewing body, its status needed to be changed from "not yet requested" to "sent". If a new section of the scientific report cited an item already in the database, a code representing that section had to be appended to the record.

To optimize retrieval, author and keyword indexes were scanned for discrepancies, i.e. an author's name appearing sometimes with one initial, sometimes with two, or U.S.A. appearing sometimes with periods, sometimes without. As a rule, we decided not to try to correct keyword discrepancies due to usage discrepancies, i.e. color/colour; data base/database. Generally, in a keyword index, the researcher must anticipate these discrepancies and allow for them in the retrieval process.

INDEXES

All fields in this database could be used for search, provided that the exact value (spelling) were known for the purely sequential search. It was also possible to build ways to sort on them. Creating a database index to them (indexed sequential organisation), however, opens up opportunities that go beyond improved efficiency.

The foremost virtue is the education of anyone accessing the information for the first time. After examining the definitions of the file, the next thing to do is to list the indexes. These can be used in some degree as Thesauri. Secondly, spellings can be standardised by editing the entries and checking the index list. Thirdly, by listing an ordered index it is often possible to detect missing information.

Thus the index could be refined by the librarian to say what she wanted about the entries in the file. Although because of time constraints we chose not to include non-title keywords or to standardise the spellings in our database, the control that the Data Manager has over the indexes easily allows these changes to be made.

The ability to refine the contents of an index was also important in this situation where the loading was designed only to give a partial analysis of the compound items. A number of "odd" situations got entered, such as a parenthesis before a keyword. In the project context it is much more effective to edit these in a few minutes than to quadruple the code written for the loading and to similarly increase the time and cost.

APPLICABILITY

Research projects which require management of fairly large bibliographic files (too large to be easily handled by a card file and the researcher(s) memory) might wish to make use of our approach. The three necessary ingredients in quickly developing an on-line database for management of bibliographic information are:

1- An experienced professional librarian who can articulate the system requirements, and then run the system.

2- An experienced analyst who can elicit and respond to those requirements, and follow up with instruction and timely support in novel situations.

3- A generalised database management package with good data definition and selection capabilities.

Our resources included these three ingredients, and in fairly short order we had a system up and running which was able to do everything required by the project for which it was developed. This included:

1- Quick entry of both a fairly large back file and of new records as required.

2- Easy editing of the record items most frequently needing updating.

3- Flexible retrieval on a variety of record elements, including author, year, title keyword, and two non-bibliographic elements required by this particular project.

4- Flexible formatting of bibliographic output, including selection of fields to be printed, and order in which citations are to be sorted.