JUSTIFYING ONE-TIME USE PROGRAMS FOR SPECIAL DATABASE APPLICATIONS

John C. Nash
Faculty of Administration
University of Ottawa
Ottawa, Ontario, K1N 9B5

Mary M. Nash
Nash Information Services Inc.
1975 Bel Air Drive
Ottawa, Ontario, K2C 0X1

ABSTRACT

Conventional techniques for building and using textual databases focus on adapting the data to an established software package or depend on the orderly and well-documented construction of special software. The purpose of this paper is to illustrate how in some instances it may be acceptable practice to prepare disposable programs to handle special textual databases. This approach will be illustrated with the problem of cataloguing a reasonably large collection of graphical materials under severe time constraints, using a limited capacity microcomputer. The implementation decisions and management choices which were made will be discussed. Some of the building blocks which were used for the disposable programs will be noted. An attempt will also be made to compare this approach with more usual ones.

L'utilisation de logiciels jetables pour certaines bases de données particulières

Les techniques conventionnelles de conception et d'exploitation des bases de données consistent soit dans l'adaptation des données à un progiciel existant, soit dans la rédaction méthodique et documentée d'un logiciel particulier. Le présent exposé a pour but de montrer comment il pourrait être avantageux, parfois, de préparer des logiciels jetables pour exploiter certaines bases de données textuelles. Pour illustrer cette approche, on analyse un problème de catalogage, sur un microordinateur à capacité limitée, d'une assez importante collection de documents graphiques et ce, avec une sévère contrainte de temps. L'on discute également les choix administratifs qui furent faits, de même que les modalités d'applications du modèle. L'on présente quelques-uns des modules utilisés pour la préparation des logiciels jetables. Enfin, on compare cette approche avec des méthodes plus traditionnelles.

1. INTRODUCTION

This paper is concerned with a problem in information management where the underlying task was the creation of a catalogue of audio-visual material. The particular problem was special, and the computer techniques we used to solve it were unique in their actual details. We consider the programs to be purpose-built tools which were discarded afterwards. However,the ideas behind the programs have quite general applicability to information management tasks where the raw data is organized into ensembles or databases involving varying sizes of records as well as varying numbers and sizes of fields. In the present case, the fields used to index this collection may vary in number from zero to three, and the number of terms in a single index field may vary from one to five.

The project giving rise to this problem stemmed from the need to organize a collection of photographs, negatives, slides, films and videotapes. After some of the material was discarded, 7893 items were included in some 1520 records in the final catalogue, which was produced in loose-leaf form consisting of an accessions list and three indices -- a subject index, a geographic index and a personality index. The indices record what, where and who the photographic material concerns. The entries in the accessions list describe the materials by date of origin, caption or title (if any), size, medium, format and colour, and the subjects, geographic location, or personality indexed. Some catalogue (i.e. database) records may be simply cross-references e.g. from photographs to their corresponding negatives. The necessity for these records arose from the way in which the collection was housed, with separate storage for the different media; that is, it is physically inconvenient to intermix negatives and slides in storage. Rather than duplicate the descriptive content for some catalogue records, we used cross-references.

We were given approximately 50 working days to complete the entire cataloguing project. Of this time period, 34 days were spent actually coding and processing the material (including labelling and arranging storage). Though this gives an average of approximately 45 records created per day, the most productive day yielded 77 new records (and a very tired cataloguer).

Thus the project to build the catalogue involved the triple difficulties of:

- wide variability of record structure and size
- media which must be stored separately but which are logically related by content
- time pressure for completion of the project.

2. POSSIBLE APPROACHES

In considering how to carry out this project, we had the following alternatives:

a) a purely manual method based on index cards;

b) a microcomputer-based method using commercially available software;

c) a microcomputer-based method using self-programmed software.

Other possibilities, such as the use of a minicomputer or mainframe, were not available to us.

Of the three alternatives, only (c) was considered practical. The purely manual approach, while perfectly straightforward and feasible, did not lend itself to development of the three indices (subject, geographic, personality), since it would require multiple copies of each catalogue card. Furthermore, any typographical errors would be awkward to purge.

Commercial microcomputer software suitable for building catalogues and indices is primarily of the database management variety. At the time the project was undertaken (January to March 1983), we knew of no software explicitly intended for cataloguing. Some programs have recently been advertised, in particular in the Library Association Record (November 1983, vol.85, no.11, page 416); however, we have no personal experience of these offerings. Of the database management packages, Ashton-Tate's dBASE II is one of the better known and serves for illustration.

The first consideration for use of any package is its capacity to handle the amount of data involved. Many database packages, and dBASE II in particular, use fixed field sizes and fixed number of fields per record, hence fixed record sizes. While the fields may be left empty, the empty space is still allocated on disks and in memory. The biggest fields and the records with the most fields therefore determine the overall memory and storage requirements. In our case, the biggest record has approximately 2000 characters. This exceeds the maximum for dBASE II, although techniques involving associated databases could be used to overcome this difficulty. However, the total space requirement with a package like dBASE II is still much higher than with programs using variable length records. The database packages also need custom programming to produce suitably formatted printouts of the catalogue and indices. Therefore, even if the primary software is available "off-the-shelf", it requires:

1. adaptation of the data to accomplish the requirements of the database software;

2. programming of the commands to prepare printed catalogues and indices.

Additionally, in our own case, there was the need of acquiring, installing and learning a database package.

3. THE INDEXING AND CLASSIFICATION SCHEME

After examining the indexing systems and classification schemes of various other audio-visual collections in Ottawa, we decided to treat the material according to a modified AACR II scheme.

The unique element in each record is the accession number. This is an alphanumeric code consisting of a letter, indicating the type of material

N = negative, P = photograph, S = slide, F = film, V = videotape,

followed by the year that the item was created, followed by a number showing the order in which the catalogue entry was created for that year. For example, S82-48 is the 48'th item that we catalogued in the "slide" category for the year of origin 1982.

Each record also contained one or more subject headings. These were modified LC subject headings, initially modified by a research institute in Canada and further altered by us to reflect current practice in the subject field. There were (eventually) a total of 51 subject headings to choose from. The subject headings were cross-referenced with "see" and "see also" references. Another index reflected the possible 82 geographic locations where the material may have originated. The personality index had 105 different personalities listed. The complete description of any item (or group of items) was only given in the accessions list.

4. SOFTWARE USED

The main software functions involved in this project were:

- entry of the catalogue accession records;

- printing of the catalogue accessions list in a reasonable order;

- assembly of the index entries in alphabetic order;

- printing of the index entries.

Because the classification and structure of the collection was highly specialized, we decided to use programs which could be discarded after the project was completed. Our task was to produce a catalogue, NOT generate cataloguing software. Therefore, we could take the unusual steps, from a programming point of view, of

- ignoring program efficiency in favour of correctness of results and savings of human time to complete the project;

- omission of checks on data and user-friendly features which required extensive program code;

- no documentation whatever outside of program comments used in debugging.

This strategy was quite successful in this particular project. However, upon learning that we intended to use a computer to carry out the work, our client anticipated that he would be able to take over the database and programs, thus having an automated system made-to-measure to his application. Given the ensemble of undocumented programs we had used to perform the principal task and to patch errors or overcome unforeseen difficulties, we were in no position to supply software to users.

Our strategy does not imply that we ignored all user-friendly features. In the data-entry program, a number of checks and aids to detection and correction of errors were especially included, since accuracy at this stage in the work was critical to the quality of the product. Furthermore we employed several programs to check or print the contents of files. These checks were supplemented by careful use of a simple data-entry sheet for each record and a set of index cards, one (or more) for each year and type of material, in order to tally the accession numbers so that we could avoid duplication or non-use of accession numbers. Also the data-entry sheets could be used to record progress, an important consideration since there was a deadline to meet.

In correcting entries and in initially preparing the index terms we made use of word processing software of our own creation along with an editor from Anderson Technoproducts Inc. of Ottawa.

5. STRUCTURE OF THE SOFTWARE USED

The starting point for the software generation was the establishment of the initial set of indices. These had records of the form

(code) (delimiter) (index term)

or

- see (index term)

or

- see also (index term)

The delimiter chosen was "~" (ASCII character number 126). Codes were a number prefaced by S, G, or P for subject, geographic or personality indices respectively. The indices were in separate files on a diskette. The numbers had no real meaning, since the indices were stored alphabetically by index term. An actual sort can be avoided, since we can enter each index record at the proper position using a text editor. Moreover, the indices were not built all at one time. An initial block of terms for each index was created based on a preliminary inspection

of the materials and discussions with the client. However, as the cataloguing proceeded, new geographic locations and new personalities as well as new subjects were encountered. The only operational nuisance we encountered in entering the index terms was the need to reformat the internal storage of the files due to a peculiarity of our computer system. However, the reformatting permitted very much more rapid reading of the indices by our programs, which were coded in BASIC on a North Star Horizon. Note that a simple sorting program cannot be used to sort the index files alphabetically because of the "see" and "see also" entries.

Having numbered index terms greatly speeded up the coding and data entry, but required that completed entries be checked very carefully. By providing a printoff for completed entries, and using a numerically ordered listing of the indices, we could correct catalogue accession entries relatively easily. Here a sorter can produce such numerically ordered lists if the "see" and "see also" lines are dropped from the alphabetically ordered file. Two points we noted in dealing with the index files were:

1. Index terms longer than one line (approximately 64 characters) caused programming inconvenience and cataloguer stress, since they were not easily displayed and distracted the eye. We carefully adjusted index terms so that they were all less than 64 characters long.

2. The daily printoff of accession entries for the cataloguer to check was prepared in a relatively compact format. For ease of use, the layout should not be so compressed as to be difficult to read, nor so spread out that the reader must deal with (and eventually discard) large amounts of paper. We adjusted the layout of the check printout several times before settling on one which seemed to work reasonably well.

The collection of material was not processed in date order, so on any one day the cataloguer would deal with a number of types and years of material. The accession numbers served to segregate the catalogue entries by type and year. This breakdown was preserved for storing the catalogue entries. Our computer (5 years old at the time of the project) had only 2 disk drives with individual capacity for only 75K characters of storage. To keep each file of accession numbers small enough so that catastrophes of human or machine origin would have minimal effect, we arbitrarily imposed an artificial limit of approximately 2500 characters per file. This was quite close to the amount of data for a single type of material and year of origin which might be catalogued in one day. The positive benefit of files which were small, easily edited and "safe" against disaster was countered by the large number of such files, with backup copies, spread over 8 diskettes. We overcame this file management problem with a series of small BASIC programs which used local extensions to the language so that the disk directory could be read. Thus we could print inventories of the accession numbers used, with reports of missing numbers or duplicates. We could sort the files, which used names like S76 for slides in 1976, or S76B for the second in a series of such files, and

even automatically consolidate all such files for a given type of material and year. A simple in-memory sorter was also developed to organize the final accession files so that each was in the proper order.

The accessions list, in final printed form, had 86 pages for negatives, 53 for photographs, 34 for slides, 3 for films and 1 for videotapes. To ensure that the printing could be smoothly picked up after a paper jam or other minor disaster, the accession printing program was based on the following principles:

A template for each record was constructed in memory. Index numbers were augmented with their corresponding terms (the index files were small enough to keep in memory for this operation).If sufficient space was left on the current page, the record was printed; otherwise, a new page was started, then the record was printed. This means no record crosses a page boundary, and considerably simplifies the task of restarting from a given page number or accession number. Pages were numbered within a given material type (or index),that is, we did not use global pagination in the catalogue, but each page is titled e.g. Negatives - 16.

The indices were built in a two stage process. First, a quite large but simple file containing records of the form

(index number) (delimiter) (accession number)

e.g.

G32~S82-21

implies that the accession record for the 21'st slide catalogued for 1982 refers to the 32'nd geographic index term. Through building this file by looking at films, negatives, photographs, slides and videotapes in that order, and within these types of material keeping to a date order, the file of index numbers is in the correct accession order and does not need resorting.

Display of the indices requires a program which can collect all the accession numbers which are appropriate to a given index term and print them in a presentable form. The program to do this was arranged to build up a part of a single index in memory by searching the simple file(s) described in the previous paragraph. The created index could then be output to a file for accumulation and editing into the final index.

6. JUSTIFICATION OF THE STRATEGY

Throughout the discussion above we have given our reasons for choosing particular approaches to the problem. We feel quite strongly that the type of project carried out is sufficiently common that others will arise for consideration by our company or others. Nevertheless, we feel

certain that the particular details of each will be such that an attempt to prepare a generalized software package for cataloguing projects of this type is a waste of effort. Moreover, keeping the software for any appreciable length of time requires that it be fully documented. For the size of programs involved (the largest is about 3 pages long) the documentation takes as much effort as the programming and debugging. What we have preserved is a set of notes (from which this paper is drawn) pointing to the ideas which helped or hindered our progress. We also note that the most time-hungry part of the catalogue process was the generation of the partial index files i.e. the sections of the final indices which were later combined using word processing software. This is hardly unexpected, since sorting and searching are well-known to be tasks which are demanding of machine resources. However, the operation of the program to carry out the index building was almost totally automatic after appropriate selection of the type of index and range of terms to be found. The major issue for the operator was one of choice of alphabetic range of index terms to make best use of the available computer memory (the entire index segment was built up in memory, then dumped to disk).

Keeping all parts of the system "small" and manually verifiable helped us to overcome several obstacles, even an unusual hardware problem -- a broken resistor -- which arose near the end of the project. After a local shop was unable to locate the problem, we repaired the machine ourselves, though not without a great deal of trouble. However, secondary problems in the catalogue or indices arising from the fault were not difficult to overcome, since the maximum range of damage was limited by the small scope of both data files and programs, complemented by simple but effective manual controls.

We do not feel that the project could have been completed on time if we had used purely manual processes. Nor would we have delivered the catalogue when due if every program was prepared with complete documentation and checking. Use of a commercial database package would have involved some programming of output formats at the very least, and may have been prohibited by the multi-disk nature of the final database of catalogue records.

7. RESPONSIBILITIES

The classification scheme, cataloguing and document layout were carried out by Mary M. Nash. Carole Gaulin performed the data entry and disk control. Software, apart from the Anderson Technoproducts Word Processing package, was written by John C. Nash, who also maintained the hardware.