

AN ARCHAEOLOGICAL NETWORK DATABASE MANAGEMENT  
SYSTEM

Jonathan C. Moffett

Institute of Archaeology, London

ABSTRACT

This paper describes a Network Database Management System (N.D.B.M.S.). It has been designed to be used with data stored using the Simple Data Entry and Retrieval (S.D.E.R.) suite of programs, which are described elsewhere (Catton et al, 1981; Moffett, 1984). However, it is also possible to use it on its own.

The N.D.B.M.S. is based on a general model of archaeological data which has a direct equivalent in a network database structure. This paper describes this model and describes the D.B.M.S. operation.

1. Introduction

Archaeology is a discipline concerned with large quantities of data, obtained by a variety of data recovery techniques, such as excavation, survey and attribute analysis. All the data that are obtained are in one of three forms:

numbers, e.g. dimensions and percentages  
characters, e.g. descriptions and documents  
images, e.g. photographs

Most archaeological research is involved in the study of these three types of data to aid our understanding of what has happened in the past.

Computers can handle large quantities of data and are capable of handling the three types of data referred to above. Thus archaeology should be a natural beneficiary of computer technology for both storage and analysis.

2. Archaeological Data

Archaeology is primarily concerned with the study of objects (or entities, but as this word is used in database terminology to avoid confusion it will not be used here). However, the definition of an object is not confined to artefacts, but also features, sites and

regions. It is anything which is unique and can be identified by a unique code and can be studied in isolation. Therefore, a sherd of pot can be considered as a single object as can a complete jar, while a particular site such as Mucking, in Essex, is an object in that it can be studied separately from the site of Charlton, in Hampshire.

An archaeological attribute is some facet of an object which is to be described. An object usually has many attributes:

OBJECT ---> ATTRIBUTE

where --> = "has a 1-to-many relationship with"

It is the attributes which are used to distinguish one object from another. For example, one attribute of a complete jar is that it is complete, whereas the sherd is part of an incomplete vessel. However, if the sherd is part of a rim then both objects can have "rim diameters" as an attribute, which makes them directly comparable. With the two sites of Charlton and Mucking, one attribute which refers to both is period, another is area, and others will involve types and quantities of artefacts and features.

Archaeological data are actual values of attributes, and refer to a specific object. For an object any single attribute will normally have only one value:

OBJECT ----> ATTRIBUTE ----> VALUE

where ----> = "has a 1-to-1 relationship with"

Since an object has several attributes it therefore has several values associated with it. Rearranging the diagram above it is possible to say that an object has several values derived from it and that each value refers to one attribute:

OBJECT --->> VALUE <---- ATTRIBUTE

Since each value is either a number, a character or group of characters or an image then it is possible to produce the following structure:

OBJECT

NUMBER      CHARACTER      IMAGES

ATTRIBUTE

It is possible to generate a network model using these three definitions. Given a single object there are many values referring to it of either of the three types of data, but each value refers to only one attribute although there are many attributes. However, given

several objects one value might refer to more than one object, for instance the same rim diameter may refer to several objects, while one attribute may have more than one possible value. Figure 1 illustrates this more complex arrangement.

### 3. A Network Database Structure

Having developed a general model for archaeological data it is possible to develop a database structure to mirror this, but first it is necessary to briefly describe the d.b.m.s. used.

The database management system used in this research is M.D.B.S. 1, which is a commercial network d.b.m.s. (M.D.B.S., 1981). One use of this package has been described in Moffett, 1984, although this current paper represents something of an update of that paper.

M.D.B.S. has a data definition language (d.d.l.) which allows the user to specify the database structure. Data are stored in fields called "items" by M.D.B.S. and these are grouped together into "records". It is possible to have several different record types, that is records with different groups of items in the same database file. The records are related to each other by sets, which are assumed to have an owner and a member, and ownership can be in one of the following four forms (and are shown with their diagrammatic equivalent):

```
One-to-one   = ----->
One-to-many  = ----->
Many-to-one  = <<-----
Many-to-many = <<----->
```

It is also possible to define the order of the member records with respect to its owner record, such as numerically or alphabetically sorted, last in first out (LIFO), or first in first out (FIFO). With the many-to-one and many-to-many sets it is possible to define the order of the owner records with respect to a member record.

There is a data manipulation language (d.m.l.) which can be embedded in a number of computer languages. These allow the user to get at the data in the database and manipulate it as required and which takes care of finding the data in the database file. Thus the programmer only needs to know what data is wanted and then issue the relevant instructions for the d.m.l. to do the rest.

The database file is organised in pages, of a size specified by the programmer. When a program uses the database it reads pages of the database into the memory and writes them back when it has finished. However, the number of pages that can be put into the memory at any one time depends on the amount of memory available. With large database files the program has to keep reading blocks of pages into memory until it locates the information it requires. The file can be spread over several disk drives, the d.m.l. takes care of any drive changes which may be necessary.

Figure 2 shows the M.D.B.S. database structure used to model the

archaeological model described above. It is not possible to store images using the computer system which is available, so this type of data is not considered in this d.b.m.s.

The database structure modifies the original model to aid the d.b.m.s. There are five record types in the database:

1. RECORD IDENTNO = object identification  
ITEM object id code
2. RECORD NUMERICAL = numerical  
ITEM value
3. RECORD DESCRIPT = description  
ITEM descriptive value
4. RECORD COMMENT = comment  
ITEM character array
5. RECORD FIELD = attribute type  
ITEM attribute description

There are two types of textual data, descriptive comments which are usually single word descriptions or abbreviations and longer more verbose sentences. The field record is here used to refer to the attribute type.

There are two types of set used by M.D.B.S.; those which link two records together, and those which are owned by the "system". These sets are entrances into the database, thus in figure 2 the sets "fields" and "idents" are both entrance sets and have to be traversed before the sets "fldnums" or "nums" can be used.

#### 4. A Database Management System

The database management system is written in Pascal with M.D.B.S. 1 data manipulation commands embedded within it. The d.b.m.s. is actually made up of about 30 programs which can chain together in predefined sequences, depending on the command entered by the user. The user is not aware of this division, except possibly for the short delay as one program replaces another in the memory of the computer.

The reason for this division is mainly due to considerations of computer memory. There is only 64 Kilobytes of memory available to the computer, into which not only the program must fit but also the pages of the database. Thus the longer the program the less memory there is available for the database, and so the greater the number of disk read and writes necessary to search through the database to find a given item. Therefore, to maximise the amount of memory available for the data and thereby quickening the search time, the programs have been made small. One other advantage of this is that it allows greater control over the individual programs and speeds up compilation time should any changes be necessary in any one program.

The programs are mostly menu driven. Whenever the user is presented with a choice of options a list of these is displayed. The d.b.m.s. can only be entered by one program, called the entrance program, which asks for the user's name and password. These can be



predefined before the database is loaded or added to the database at a later stage. It is not possible to enter the database by any other program in the hierarchy. If this is attempted the program in question automatically chains to the entrance program.

Once the user has successfully entered the d.b.m.s. they are presented with the following menu:

Database Management Control Menu

Choose option:

- A - Search database
- B - Edit database
- C - List information about database
- D - Database contents statistics
  
- \* - Set output destination
  
- ! - HELP!!!!
- \$ - Leave database

(A-D/!/\$)?

#### 4.1. Searching

This option allows the user to find objects which fulfill certain conditions. It is possible to specify either descriptions and/or numerical values in the conditions. Two logical operations are allowed, "AND" and "OR" for intersection and union of sets. With numerical values it is also possible to specify a combination of less than, greater than and equal to in the condition statement. Once the statement has been entered it is checked to see whether it is correct. If a description is not found the user is able to choose from the seven closest descriptions in spelling.

It will be noticed in figure 2 that the record DESCRIP is pointed to itself by the set "dindex". This is to allow the use of synonyms, so that the owner set is the most commonly used description, but others which are also used are also stored in the database, thus forming a data dictionary.

#### 4.2. Editing

This option allows the user to modify the contents of the database in some way. This includes the adding, deleting or changing of fields, descriptions, numerical values or objects.

#### 4.3. Listing

This allows the user to list information about the database, such as listing all the descriptions, with or without synonyms, and the types of object contained in the database.

#### 4.4. Statistics

This option allows the user to find out how much data is actually stored in the database, in order to monitor its size.

#### 4.5. General options

There are four general options which are available with most menus of the d.b.m.s. The first is to set the output destination which can be either the console screen, the printer or a file on the disk. At most levels there are help commands which display the contents of a help file on the screen. It is possible to move to other modes as well as leave the d.b.m.s.

#### 4.6. Extensions

The above describes the general options available. It is also possible to add other programs to the d.b.m.s. hierarchy to do specific tasks and control specific output devices, such as a digital plotter or graphics terminal. This really depends on the contents of the database, and is decided by the programmer and the users of the database.

#### 5. An Example

The Mucking excavation data (Catton et al, 1982) has been used to test this database structure. There are several large sets of data which have been recorded on to floppy disks and these are being integrated by loading them into a single database file. This will enable questioning of the data about the site to be carried out much more easily than having to do it manually. Several extension programs have been written to take into account different sets of data stored in the database, as well as allowing the use of different output devices, such as a digital plotter.

#### 6. Conclusions

This work began in 1981 when 8 bit microcomputers were the most widely available with considerable amounts of software available for them. However, more recent developments in computer technology are making these systems look out of date especially as the new 16 bit microcomputers are now being sold in the same quantities. Thus in the future problems with the amount of memory available for a process will no longer exist, especially when the memory is of the order of megabytes rather than kilobytes. Since a version of M.D.B.S. exists for 16 bit microcomputers it is hoped to upgrade for these microcomputers the system described here.

## References

- Catton, J.P.J., Jones, M.U., and Moffett, J.C., 1982,  
The 1965-1978 Mucking Excavation Computer Database, Graham, I and  
Webb, R.E., Eds Computer Applications in Archaeology, 1981, pp  
36-43.
- M.D.B.S., 1981  
User's Manual and Documentation.
- Moffett, J.C., 1984,  
The generation of archaeological excavation databases, in Insti-  
tute of Archaeology Bulletin.

Figure 1: A General Data Model

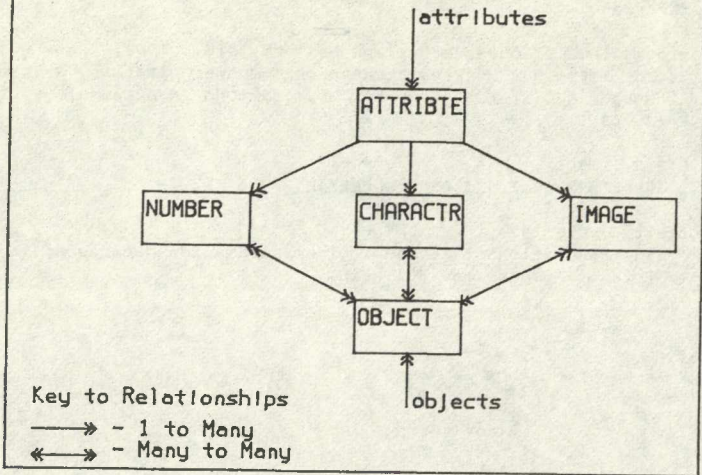


Figure 2: A General Network Database Structure

