DESIGN OF A DATABASE FOR ARCHAEOLOGICAL SITE DATA

Judy Chapman

Information Systems Research Group, Faculty of Mathematics,
Open University, Walton Hall, Milton Keynes, MK7 6AA.

Abstract

This paper defines a data model for archaeological site data which can be used to design a database for implementation on any available database management system. This conceptual data model was produced by applying the techniques of data analysis so it was derived by studying the items of data and the perceived requirements for processing that data i.e. the current information retrieval needs as well as those needs envisaged for the future.

The paper then describes the transformation of the conceptual model to a logical data model which gives the design in a format appropriate for use with the Rapport relational database management system.

The type of data being stored has several awkward properties which impact upon a Rapport implementation. These include a large number of empty fields and some completely unused record types for a particular site.

1. INTRODUCTION

This paper defines a data model for archaeological site data which can be used to design an implementation on any available database management system. The main objective of the design was to produce a general pattern from which the individual archaeologist, unaided by a database expert, can then select those portions applicable to the particular project in prospect. It was also felt that there would be some gain in terms of simplicity and economy during input if selected fields from the general pattern of the database could be suppressed in particular instances. The database system used to implement the general design was RAPPORT-3 [LOGICA (a)]. This system was chosen for a number of reasons which are outside the scope of this paper.

The model was based on a description which was compiled in the Department of Archaeology at Edinburgh University by Trevor Watkins. It was discussed by his colleagues to ensure that it covered the standard range of records and fields required. It was recognised that none of them will use all the fields for one site, but any of them can imagine situations where each field would be necessary. This description was supported by a

list of typical questions which might need to be asked of the database.

The paper describes the general strategy adopted in the database design and then considers each stage in some detail.

2. STRATEGY

It is now generally accepted that the development of a database system involves a number of steps. These stages have been summarised [CHAPMAN 83] and the main steps were used to develop the database for this archaeological site data as follows. Firstly a conceptual data model was produced by applying the techniques of Data Analysis and Functional Analysis [ROBINSON 81] and [OU 80]. So the conceptual data model was derived by studying the items of data and the perceived requirements for processing that data i.e. the current information retrieval needs as well as those needs envisaged for the future. This conceptual data model was defined using a variant of the constructs proposed in [CHEN 76]. The model was then transformed to a logical data model which gives the design in a format appropriate for use with the Rapport relational database management system. Storage schema definitions and storage media definition are subsumed in the definition of the physical implementation. That is the size of files and fields, the access path requirements and indexes to facilitate the information retrieval were added at a later date. The final stage, the operational definition, can be changed as experience with the implemented system is gained.

The final phase of the work was considering how best to load the initial data into the database and then implementing methods of data retrieval and updating of records that could be easily used. This step highlighted a number of problems; the major ones are addressed in this paper.

3. CONCEPTUAL MODEL

Working from the supplied general requirements for an archaeological excavation database an initial data model was drawn. The data-items (or fields) were related to four types of site record:

(1) The written record of the structures and features excavated, their relationships to one another, and supporting notebooks relating to the day-to-day observations.

(2) The artefacts and the physical samples taken from the excavations, together with the written records relating to them.

(3) The photographic record and its index, relating to (1) and (2) above.

(4) The drawn record and its index, relating to (1) and (2) above.

The list of likely retrievals and processing which would use these data-items included postulated uses as well as those required immediately. In essence the following were to be possible. During the post-excavation phase the excavation records would be edited, revised and added to. There would be a need to change existing data in the database, add new data. The ability to modify record-types by adding new field-names and to form new record-types using some existing data were excluded as such dynamic facilities are not currently available in database management systems.

The likely needs for interrogation of the records were given as:

(1) the ability to sort and select both alphabetically and numerically

(2) to answer typical questions, such as:

  (a) all information (contextual, artefactual, photographic and drawing) on a particular context or certain group of contexts.

  (b) any photographs/drawings/finds that refer to a context or group of contexts

  (c) any contexts (plus specified fields on those contexts) which have produced artefacts of a certain material of certain type

  (d) simple listings, e.g. all frames on a particular reel of photographs, or all drawings executed in a certain session.

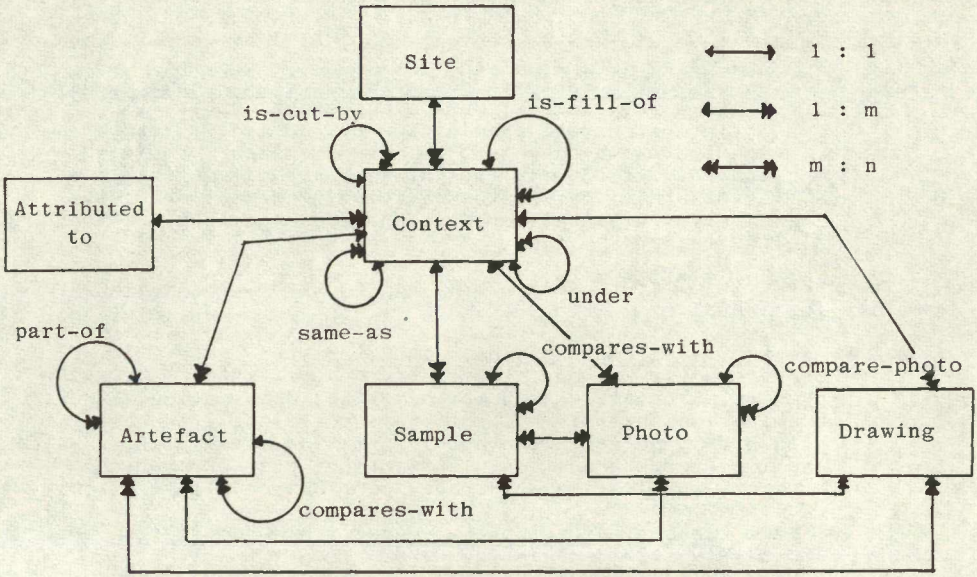
(3) the ability to respond to on-line queries which arise in the course of routine post-excavation work. It should also be possible to dump data on disk or to hard copy on request

(4) to provide those selective listings and full listings necessary to complete the archive record.

The initial attempt at modelling the data in a conceptual model using entities, attributes and relationships disclosed many semantic problems. This very simple model is given diagrammatically in Figure 1 and contains semantic errors. Once the problems had been clarified by discussion the process of normalisation [DATE 81] was applied to the entities to highlight the possible need for futher entities and to prevent updating anomalies by making the model well-formed. Conceptual modelling and normalisation are techniques which are applied to give an implementation free model i.e. no consideration is given to the database management system which will subsequently be used. The techniques help to clarify that the semantics of the data are understood and they lead to a data model which can be used to test that all known (anticipated) functional requirements can be met from the data structured in that form [ROBINSON 81]. The entities which resulted from this process are given below. The attributes underlined form the identifying key. It was decided at an early stage that a database would only hold data for one site so the identifying key need not contain the site signature.

Figure 1

Initial Conceptual Data Model



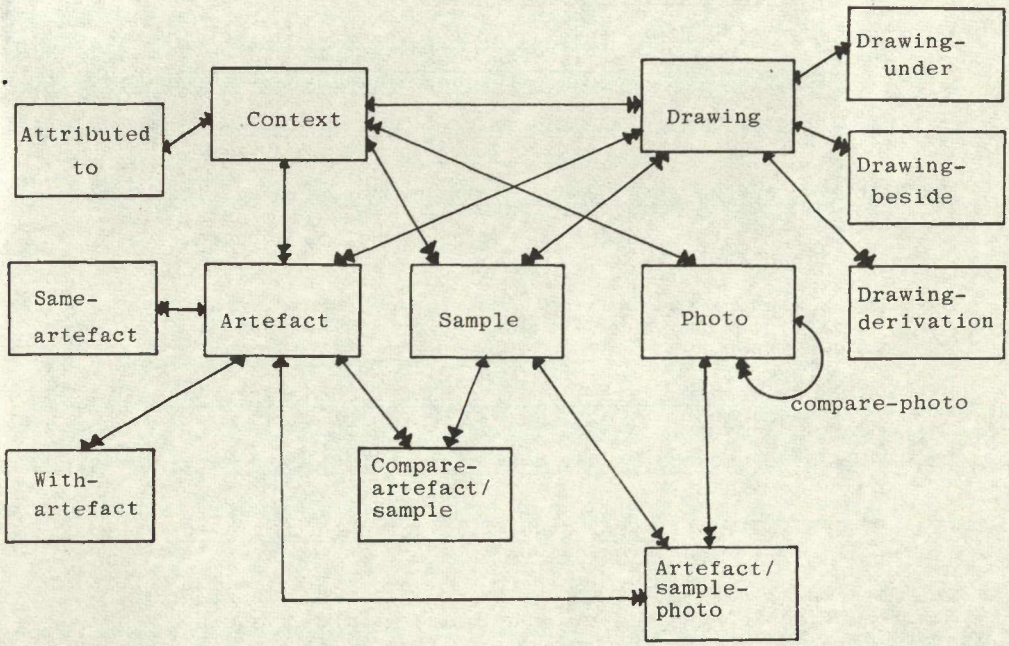


Figure 2

CONTEXT (season, area, context code, type, nature, description, interpretation, length, breadth, thickness, top-height, bottom-height, unit-name, under-context-code, cut-by-context-code, same-as-context-code, fill-of-context-code)

ARTEFACT (season, area, context-code, field register number, day-book temporary number, museum accession number, description, expert examination by, comments, length/height, breadth/diameter, thickness, weight, position, grid east, grid north, heighting, part-of-context-code, site plan/section, post-excavation plan/section, post-excavation drawing, publication reference, publication illustration)

Although very similar to ARTEFACT it was decided to have a separate entity for SAMPLE.

SAMPLE (season, area,context-code, field-register number, day-book temporary number, laboratory number, museum accession number, material, reason for sampling, description, expert examination/analysis by, comments, position, grid east, grid north, heighting, other information, site plan/section, post-excavation plan/section, post-excavation drawing, publication reference, publication illustration)

WITH-ARTEFACT (season, area, context-code, field-register-number, with-field-register-number)

SAME-ARTEFACT (season, area, context-code, field-register-number, same-field-register-number)

COMPARE-ARTEFACT/SAMPLE (season, area, context-code, field-register-number, cfseason, cfarea, cfcontext-code, cffield-register-number)

ARTEFACT/SAMPLE-PHOTO (season, area, context-code, field-register-number, reel number, frame number)

PHOTOGRAPH (season, reel number, frame number, camera, lens, aperture, shutter speed, film type, film details, taken-by, subject, detail, viewpoint, compare, published as)

DRAWING (season, area, drawing number, type, scale, drawn by, date, medium, subject, detail, grid area/squares, published as)

DRAWING-UNDER (season, area, drawing number, under-drawing-number)

DRAWING-BESIDE (season, area, drawing number, lies-beside-drawing-number)

DRAWING-DERIVATION (season, area, drawing number, derived-from-drawing-number)

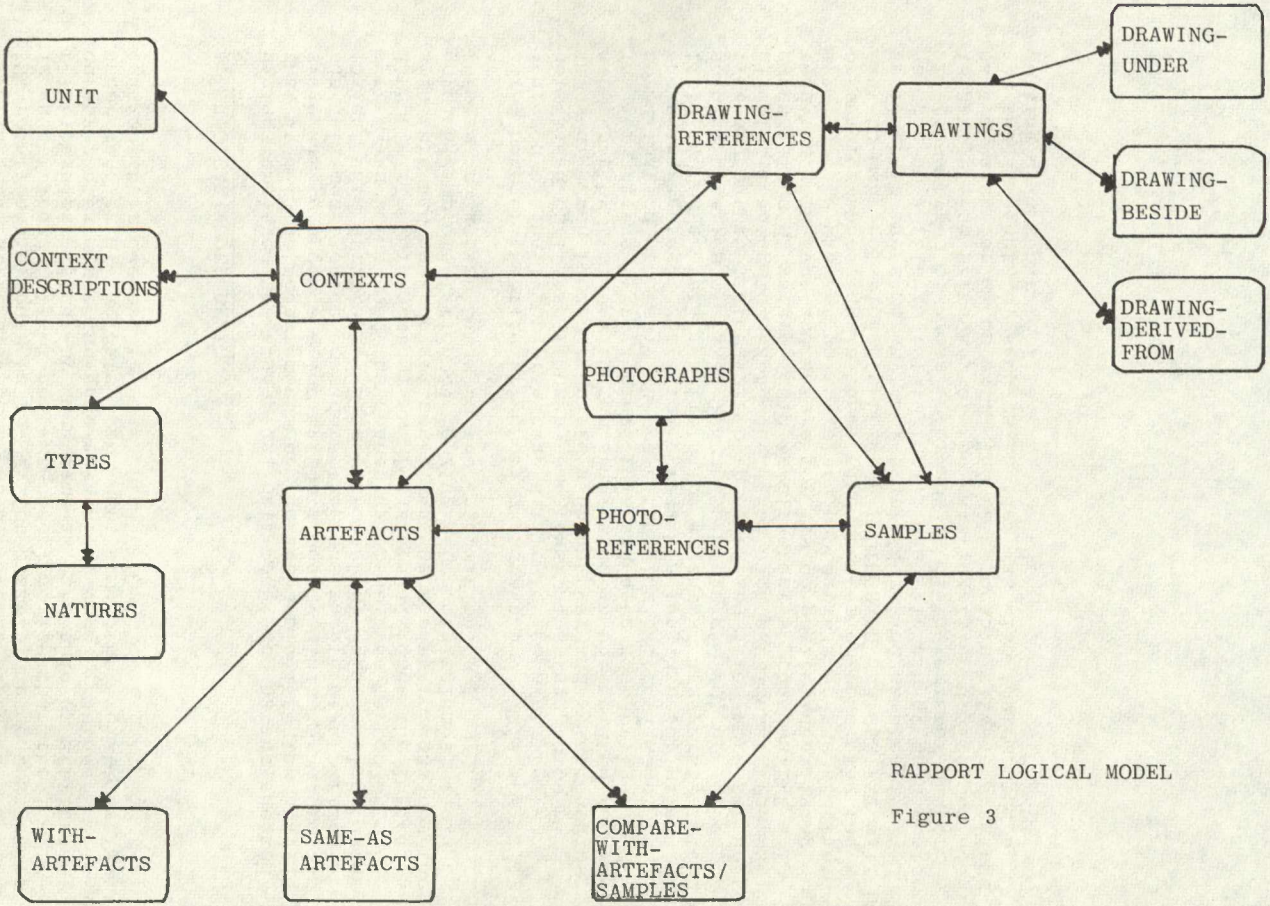
The data model diagram is given in Figure 2. The apparent discrepancies between the two models are mainly caused by

treating several relationships as attributes since in practice the relationship would be 1 - 1; for example, in context there is an attribute fill-of-context-code. Notice that the 1 - 1 relationships, is-cut-by and same-as, are held by attributes cut-by-context-code and same-as-context-code. Further understanding of the data also meant some other relationships changed their degree.

4. LOGICAL MODEL DESIGN

Having completed the conceptual data model and having checked that it met the requirements of the archaeologists the next stage was to produce a logical data model. This is a form of the data model which considers the constraints of a particular logical data structure. Since the RAPPORT relational database system was being used the conceptual model was mapped onto RAPPORT files of records, [LOGICA 82 (a)]. Once the logical structure had been determined it was necessary to consider the physical implementation by looking at record volumes, field formats and access paths to the information.

First however the file structure was drawn up. This was achieved basically by producing a file for each of the normalised entities although some constraints had to be imposed. Each record requires a unique key and relationships are shown by fields with the same value in the related records. The work necessary to achieve this had been done by the normalisation process. The resulting logical data model is shown in Figure 3. To allow a variable length description field and interpretation field for a context a separate file holding a series of fixed length segments was set up. The type and nature fields were both coded and files were set up to give the meaning for each code. For each particular type the coded values of nature would be different so that NATURE required both type and nature for its composite identifying key. A file was set up to relate photographs to the artefacts or samples appearing in them. The problem of resolving the many-to-many relationship between PHOTOGRAPH and CONTEXT will be achieved in the same way as for artefacts and samples i.e. setting up a file to relate them. One file cannot be used for all these types of cross reference as field-register-number is part of the prime key of the existing file but it does not have a value for a context. The relationship between drawings and the artefacts or samples in them was also implemented by setting up a separate file.



RAPPORT LOGICAL MODEL

Figure 3

5. PHYSICAL MODEL DESIGN

The translation to the physical model was performed on the basis of the access and storage patterns of the data. The main work here was in deciding the size of each field and its storage format so that maximum information could be held without making the records too large. It was discovered that the RAPPORT configuration used had a limit on its buffer which was exceeded in the initial implementation so two fields were deleted to reduce the size to within the limit. The two fields removed were 'comments' from ARTEFACT and 'other information' from SAMPLE, neither appearing to be essential.

Constraints were imposed on the length of descriptive fields. In particular the description field in records in the artefact file was decomposed into six separate fields each of a restricted size. Where several occurrences of a field were being held as a vector the number of repetitions was kept as small as possible and in several cases it was decided that in practice only one value would be needed so repetitions were excluded. For example in an ARTEFACT record it is only possible to record that it has been examined by one expert and in PHOTOGRAPH one only comparison is allowed in the 'compare' field. It was at this stage too that it was decided to remove the area field from many of the records as it contained no information additional to that in the context code.

Access paths were considered in order to decide what indexes would be needed to enable the envisaged retrievals to be made most efficiently. For the CONTEXT file indexes on context code, type and nature were set up. For ARTEFACT, together with an index on context code were ones on field register number, and on certain fields within the description. PHOTOGRAPH has an index on reel number and SAMPLE has one on material (which is part of the description field). Indexes on under-context-code, cut-by-context-code, same-as-context-code and fill-of-context-code were also required on the CONTEXT file but in many occurrences the value in the field to be indexed would not be present although it might be added later. Hence producing these index files would result in a high number of collisions. This would mean that retrieval using them would not be very efficient and a complete file search might be equally efficient. The number of collisions on an empty field, denoted by spaces, was considerable and there seems to be no easy solution to the problem as it is not possible to use the Rapport index facility on a file so that records with an empty field can be excluded from the index. This problem has still to be solved satisfactorily.

Further problems were encountered in determining the hashing values to be used with the key fields in order to spread the records across the files minimising the number of collisions occurring. It is hoped that with experience and carefully controlled experiments suitable values could be determined from a pilot version of a database. However it is clear that the distribution of the keys will vary from site to site and may need adjusting for each database set up using the general model.

The layout of the files on the available disks needs to be specified. This is achieved using channels. For the pilot database each file was placed on a separate channel.

6. USING THE GENERAL MODEL

A further large set of problems has to be addressed. The physical model, transformed via the logical model from the conceptual model, is very general and in practice the complete model is unlikely to be required. As well as determining file sizes of the correct order for a particular implementation to improve efficiency it is necessary to consider which files are required for data from a particular site. If files that will remain empty are included then valuable storage space is wasted. The alternative is to recompile the logical model (the data definition file, DDF) having removed the unnecessary files for that particular site. All programs using that DDF would also need to be recompiled. This would impose additional problems for an archaeologist trying to set up a database with little or no help from a Rapport 'expert'.

Another consequence of this need to make the system for setting up and using a database one which could be easily applied was the problem of loading the data; much of the interrogation and updating of the database could be handled relatively easily using the interactive query language, IQL, [LOGICA (c)] once the main constructs in that query language had been learnt. Loading the database was a significant problem since it was essential to minimise the imput needed. If the query language (or the load utility [LOGICA (b)]) was used to load the data then every field had to be input for each record. For example full keys needed to be typed in even when the records all belonged to the same 'owner' i.e. all artefacts for one context code. Basically more time consuming was the need to enter a space for every empty data-item in a record so significant effort would be spent on meaningless key depressions. These problems made it desirable to write a load program. The interface language available is FORTRAN IV which causes some similar input difficulties but by producing a suite of modules which used default values it is possible to decrease the number of fields needing to be entered for any particular record type. The modules are parameterised so that they can be used for any configuration of the database once it is known which files are required and which fields, if any, are known to be totally unnecessary. However if loading is carried out using this purpose built program then the input is constrained by a strict ordering of the input records unless the full keys are to be included in every record. These modules can be linked so that they can either be used in interactive mode from a terminal or indirectly with the data coming from a file and the program used in batch mode.

7. SUMMARY

This paper had defined a general data model for archaeological site data. This data model was transformed into a physical model, via a logical model, using Rapport to show how a conceptual data model can be used to design a database in the appropriate data structures for any database management system. Some problems inherent in the physical design of a Rapport database for site data were addressed as was the need for an efficient, easy to use, load program. Interrogation and updating of such a database have not been considered. A summary of the facilities availble in a Rapport system can be found in [SCHMIDT 83]. The length of the paper unfortunately precludes the provision of the complete Rapport Data Definition File.

Acknowledgements

My thanks are due to Trevor Watkins in the Department of Archaeology at Edinburgh University who instigated the work and was most patient in explaining the requirements of archaeologists; and to members of the Database Systems Unit of Edinburgh Regional Computing Centre where I carried out this work while on secondment from the Open University.

References

[CHAPMAN 83] Chapman J and Robinson H, A database administration support environment, Real-Time Programming 1983, Pergamon Press, 1983

[CHEN 76] Chen P P S, The Entity - Relationship Model - Toward a Unified View of Data, ACM TODS, Vol. 1, No. 1, March 1976

[DATE 81] Date C J, An Introduction to Database Systems, Third Edition, Addison-Wesley, 1981.

[LOGICA 82 (a)] RAPPORT-3 Designing and Using a Database, LOGICA 1982

[LOGICA 82 (b)] RAPPORT-3 FORTRAN User Manual, LOGICA 1982

[LOGICA 82 (c)] RAPPORT-3 Interactive Query Language User Manual, LOGICA 1982

[OU 80] The Open University, M352 Computer-based Information Systems, The Open University Press, 1980

[ROBINSON 81] Robinson H M, Database Analysis and Design, Chartwell-Bratt, 1981

[SCHMIDT 83] Schmidt J W and Brodie M L, editors, Relational Database Systems, Springer-Verlag, 1983