29

The development of a bibliographic information retrieval system for archaeological reports using thesauri

A. R. Winterflood
School of Computing, Kingston Polytechnic

G. G. Wilkinson
Department of Computer Science, University College, London

M. Rhodes
Department of Urban Archaeology, Museum of London

29.1 Introduction

This paper outlines a software development project which has been taking place over the last fifteen months to construct a major bibliographic information retrieval system for Greater London archaeology. The aim of this work has been to construct a purpose-built software system from which scholars and members of the public can obtain references to archaeological information contained in the many thousands of published books, papers, technical notes and reports on numerous excavations and finds in the Greater London area, using an extensive and original archaeological keyword thesaurus.

Records of archaeological discoveries in London have been made continually since the 17th century, and are preserved in a large number of books, journals, periodicals and archives. Most lack indexes, and there is not even a comprehensive list of relevant titles. Without a means of accessing archaeological information, its value is much diminished. Library indexes are restricted to publications in stock and refer only to the main topic covered by each item. The Council for British Archaeology Bibliography covers only British publications since the war, and it can take hours to search the thirty or so volumes to locate references to a particular class of find.

In 1982 the Museum of London decided to remedy this situation by creating a computer-based bibliography with the aim of making published information readily accessible to the general public, planning authorities, museum staff etc., and to simplify, accelerate and improve the accuracy of archaeological research and hence the writing of archaeological reports.

The project was initiated by a consultative committee comprising museum staff plus representatives of the Department of the Environment, the Greater London Council, the London and Middlesex Archaeological Society, the Council for British Archaeology and the National Monuments Record. A professional indexer was appointed in 1982 primarily to undertake the

long task of indexing the relevant literature and archive reports for inclusion in a 'publications database'. The indexer (Audrey Adams) has also been working on the production of sophisticated thesauri which will enable finds keywords to be automatically linked to synonyms and related terms in order to make archaeologically-useful searches of the publications database (see Rhodes 1986 for more details). The software development began in the Spring of 1986 with the assistance of Kingston Polytechnic and, more recently, University College London. The software production is now almost complete.

29.2 System overview

The heart of the bibliography system consists of a large database which stores information about publications and the 'finds' (*i.e.* buildings, archaeological features, and loose artifacts) to which those publications refer. Additional details about sites, geographic locations etc., are also to be kept in the database to support a number of different types of bibliographic query.

The basic system structure is shown in Fig. 29.1 below. Besides the publications database it features two thesauri which are used in the search process (a DATE DICTIONARY and an OBJECT DICTIONARY). The software has to support two main system functions: QUERIES by individual researchers (*i.e.* the search process) and MAINTENANCE (*i.e.* the operational updating of the publications database and/or the thesauri). The publications data are stored in

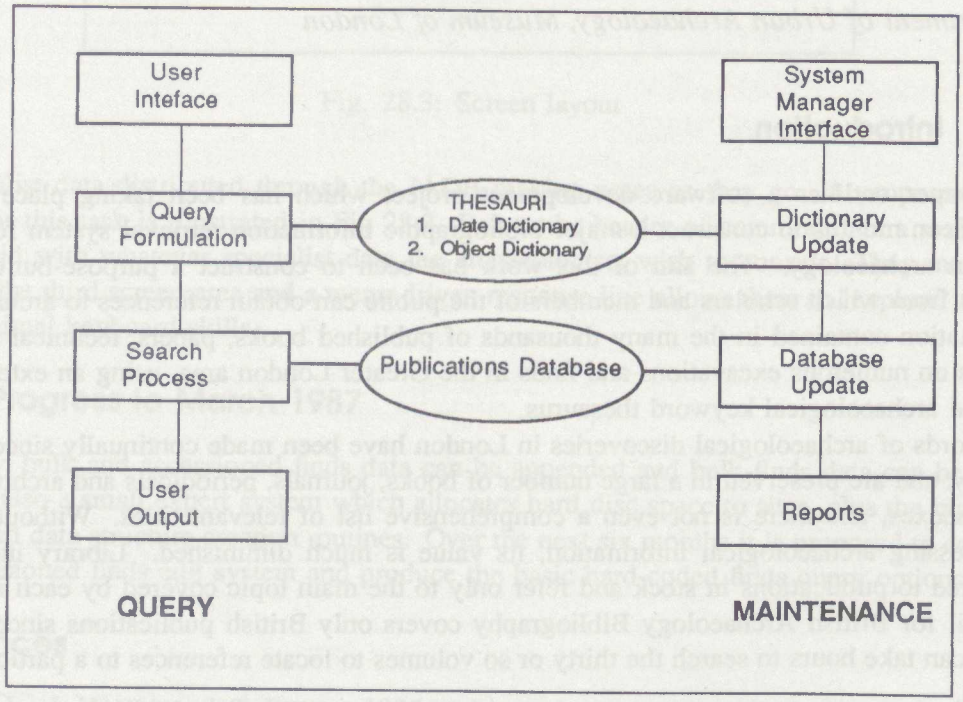

Fig. 29.1: Basic system structure

a hierarchical form as shown in Fig. 29.2.

The publication entry at the top level contains the normal bibliographic reference details (*e.g.* journal name, editor, title etc.). Individual parts (such as specialist reports) within a main publication are then specified separately. Details of sites mentioned in each publication part are then given (*e.g.* site no., SMR code, address, OS grid reference). The 'finds' discussed in the publication are then recorded and linked to each site. This database is therefore extensive and

designed flexibly to support a number of different types of bibliographic query. Functionally,

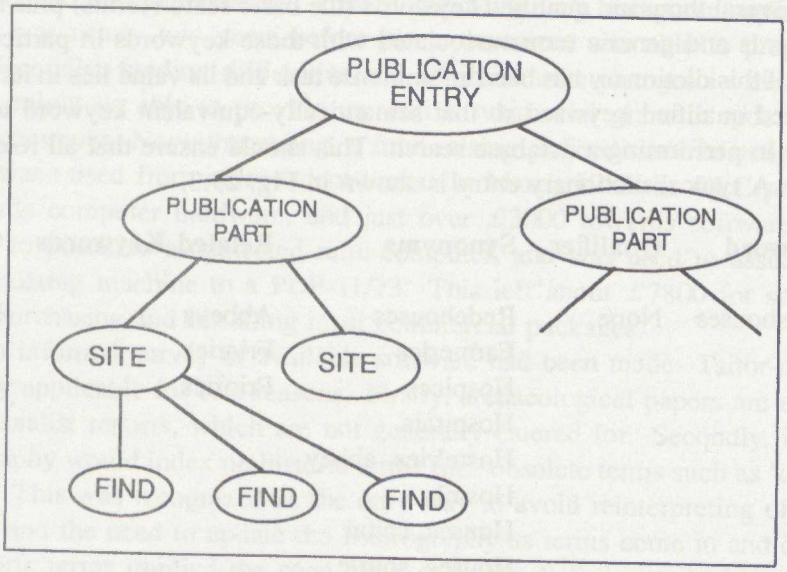

Fig. 29.2: Hierarchical structure of the publications information

the Museum requires support for user queries of the form: 'Output a list of all references to ...

- 'AXES'
- 'ROMAN DAGGERS'
- 'MEDIEVAL ABBEYS'
- 'POST-MEDIEVAL TERRACOTTA TILE'
- 'VALENTINIAN COINS'
- 'RUBBISH PITS 50–200AD'
- 'DRAINAGE GULLIES AT OS GRID REF TQ123456'
- 'BRONZE AGE KNIVES AT SMR REFERENCE 123456 AUTHORED BY J. SMITH'
- 'AGRICULTURE'
- 'ARTICLES ENTITLED 'THE OAK PILES AT THAMES WHARF'

Each find entered in the publications database must be given a specific keyword (*e.g.* AXE, DAGGER, RUBBISH PIT etc.) which is used as the main index term. Such keywords may be modified by the use of up to seven 'qualifiers' (e.g. BRONZE, TERRACOTTA) and must be linked to a specific broad period in history (such as ROMAN, POST-MEDIEVAL). Individual finds are therefore indexed by unique qualified keywords which are period-controlled. More precise date information may be entered for finds to tie them to specific years, ranges of years or dynasties ('sub-periods').

Perhaps the most valuable components of this system are the two thesauri which are to be used to make queries simple to perform and as realistic as possible. One is used to expand 'date' terms (*e.g.* to convert a regent's name into specific dates); and a second, far more extensive one, to

expand finds keyword terms for searches. The latter thesaurus—the OBJECT DICTIONARY—will contain several thousand qualified keywords (the basic search terms) plus lists of synonyms, related keywords and generic terms associated with those keywords in particular periods. The construction of this dictionary has been an immense task and its value lies in its rôle in expanding a user-specified qualified keyword so that semantically-equivalent keyword terms can be used automatically in performing a database search. This should ensure that all relevant publications are retrieved. A typical dictionary entry is shown in Fig. 29.3.

Keyword	Qualifier	Synonyms	Related Keywords	Generics
Almshouses	None	Redehouses	Abbeys	Religion
		Farmeries	Friaries	
		Hospices	Priories	
		Hospitals		
		Hostelries, abbey		
		Hostels		
		Houses, spital		
		Houses, spittle		
		Infirmaries		
		Maison-dier		

Fig. 29.3: A typical OBJECT DICTIONARY entry

Synonyms are used automatically in expanding a given search term. Since finds in the publications database may only be indexed using valid terms contained in the dictionary, all synonyms and related keywords must themselves be present to avoid searches which generate a null result. Careful controls have therefore been built into the database maintenance functions to ensure that all finds entries are controlled by the dictionaries and that new dictionary entries are checked for consistency.

29.3 System development issues

The system which the Museum originally set out to construct was an ambitious one in the sense that it was intended to support a large publications database which should be searchable by complex and archaeologically-realistic queries. The Museum therefore had to carry out an initial feasibility study to see if it was possible to run such a system on the existing hardware configuration. Also difficult choices had to be made about the best approach to system development within budgetary constraints. Essentially, any project of this nature can be tackled in one of four ways:

1. buy-in a complete system (*e.g.* from another museum);
2. buy-in a relevant package;
3. develop totally in-house; or
4. develop in-house with external assistance.

The initial plan was to purchase a dedicated mini-computer running XENIX, which could handle a database package such as STAIRS. This would be located in the Museum's library.

It was intended to choose a computer system during the year 1984–5, although in the event the project faced two serious setbacks. Firstly, the Museum's computer supervisor resigned in 1983, and it was nearly two years before his replacement was able to prepare a systems specification. Secondly, funding difficulties and administrative changes in the GLC wrecked hopes that they might be able to provide us with a substantial grant in order to purchase a dedicated mini-computer. Notwithstanding, a fund-raising project was initiated in 1984, and a sum of £4500 was raised from independent trusts. In March 1985 the GLC provided a grant of £9000 towards computer hardware, and just over £3000 towards software. The £9000 was insufficient to purchase a dedicated mini-computer, and was used to assist in upgrading the Museum's existing machine to a PDP-11/73. This left about £7800 for software, a sum insufficient for purchasing and installing most commercial packages.

Meanwhile an informal survey of available software had been made. Tailor-made packages were not directly applicable for two reasons. Firstly, archaeological papers are unusual in that they include specialist reports, which are not generally catered for. Secondly, it was decided that the Bibliography would index on historic terms (*i.e.* obsolete terms such as 'celt' would not be modernized). This was recognized as the only way to avoid reinterpreting old publications (full of pitfalls), and the need to update the Bibliography as terms come in and out of fashion. The use of historic terms implied the need for a complex dictionary which was likewise not catered for by existing packages.

Museum cataloguing software was also examined. Although GOS was too basic, STIPPLE, albeit incomplete, offered a number of powerful cross-referencing features (Dixon 1983). However, this software could be utilized only by hiring space on a mainframe run by a specific company, which might enjoy a limited life and would involve high rental charges. BRS/SEARCH (developed from STAIRS) could work on a UNIX[1]-based mini-computer, and seemed to fulfil many of our basic requirements. However, in addition to the purchase price, installation costs would be high in view of the complexity of our dictionary, and on top of this, an annual rent would be charged. In view of the expense of these options, it was decided to explore the possibility of programming the Bibliography from scratch.

The third development option (in-house production) was ruled-out owing to the lack of a programmer team capable of tackling this scale of project. The fourth option was therefore selected as the best approach and proceeded with the temporary employment of computer science sandwich students under academic supervision. This had the advantage of offering useful experience to the students in real-life software engineering and provided the Museum with highly-trained programmers.

Since this work was intended to lead to a valuable software product which might need to be readily enhanced, modified and ported as necessary, it was decided at the outset to develop the system in a modular fashion using a formal software engineering methodology. The Constantine-Yourdon structured design method was chosen for this (see Page-Jones 1980, De Marco 1978) and proved extremely worthwhile. The C programming language was selected for coding efficiency and the system has been implemented on the Museum's PDP-11/73 running the XENIX-2 V3.2 operating system. We review the system design below and outline some of the experiences gained in using this software engineering approach in the sections which follow.

29.4 The development phase

The main phases of system development in the Constantine-Yourdon approach involve:

[1]UNIX is a trademark of Bell Laboratories

1. requirements specification;
2. system analysis;
3. detailed design; and
4. implementation and testing.

The requirements specification document was drawn up in the early part of 1986 and was agreed with the relevant staff within the Museum. This specified the main functional requirements (query functions and maintenance functions) and acted as the basis for subsequent design work.

The system analysis proceeded with the construction of a conceptual model and definitions of the logical data structures required and entity relationships. A model of the system was represented as a hierarchical set of 'Data Flow Diagrams' (DFDs) which described on a conceptual level how the system was to behave. These diagrams represent individual processing tasks as 'bubbles' and data flows between them as labelled arrows. Sources and sinks of data are shown as boxes and files as names over solid lines.

A typical example of one of our DFDs is shown in Fig. 29.4. Note that the DFD only shows logical data flows. Sequencing information is not made explicit by such diagrams—that is dealt with in the detailed design. Also it is important to realise that DFDs can be refined to greater levels of detail. For example the 'Validation Process' bubble could itself be turned into a more detailed DFD with separate bubbles for the different validation functions it performs.

Overall we found that the DFD's provided a very useful graphical way of representing the software functions and they proved an invaluable aid to project team discussions for refining the system design. Such a notation is readily comprehensible to non-computer literate archaeologists and was of great benefit in this work.

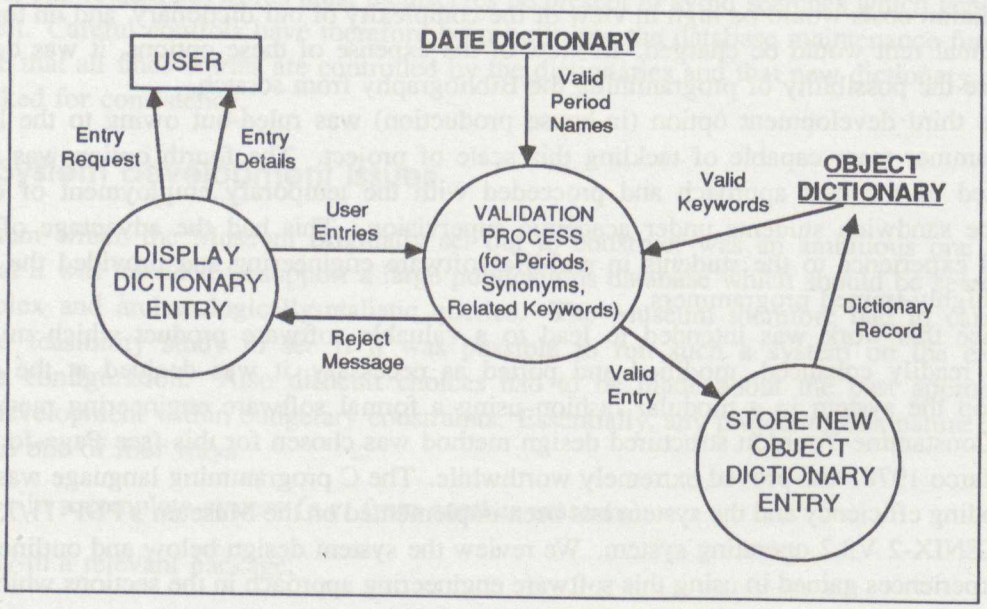

Fig. 29.4: DFD For Object Dictionary record entry

After completing the system analysis with levelled DFD's the detailed design was carried out using the 'structure chart' notation (to identify individual program modules and their relationships). A 'Data Dictionary' and 'Procedure Dictionary' were also produced to lay down

actual data field lengths/types etc and precise procedure interfaces (*i.e.* calling parameters required and functions etc)

29.5 Problems encountered in software engineering

A formal software engineering methodology like the Constantine-Yourdon approach is essential to produce code which can be modified and extended with ease at a later date. Indeed we chose this method for that very reason. This kind of approach does however demand rigorous adherence to the development plan if the software is to be produced efficiently. One of the main problems we encountered in the implementation phase was that some of the museum staff who were closely-involved with the project (but not on the programming team) suggested functional changes at various stages well into the coding phase. This kind of occurrence breaks one of the most fundamental rules of software engineering and necessitates considerable re-design of software. A number of workers have pointed to this problem in the past (see Fig. 29.5 for example) and shown that the later functional changes are made the more costly and time consuming the changes become.

In such a changing context, additional computer-based tools are necessary to relieve the tedious process of updating the complete documentation set (DFDs, Structure Charts, Data- and Process-Dictionaries). Careful management is also necessary to ensure that programmers update the documentation *before* continuing the coding, respect the modular organisation of the program and test and debug modules thoroughly before integrating them in to other units. Unfortunately close project control was not possible during the main coding phase and some of the benefits of this structured system development approach were lost.

However we believe that such problems are likely to plague any system development in an archaeological context—primarily because the end-users and project-initiators (expert archaeologists) are likely to be unfamiliar with the software engineering formalisms and the disciplined development cycle required by effective software development.

Indeed it is becoming increasingly apparent that the effectiveness/failings of customer-software team interactions significantly affect overall system development efficiency. In highly-organised bureaucratic organisations where a software system is simply replacing a clearly specified paper procedure the rewards of adopting a formal software engineering methodology are high. Our experience has shown that in an environment where a totally new research tool is being created with the assistance of archaeologists who are new to software 'engineering' the approach may not work as efficiently as it otherwise might. Hopefully these problems will gradually disappear as the archaeology community becomes more used to formal approaches.

29.6 Concluding remarks

In the previous pages we have outlined how the Bibliography system originated and shown some of its novel features—particularly the construction of a keyword thesaurus of significant proportions. The software development is now almost complete and it is clear that the use of a formal methodology has been very beneficial despite some setbacks. The main benefit of structured design has been the production of a modular system which can be easily enhanced and/or ported. A full operational keyword thesaurus will be loaded into the system soon and entry of publications data will commence in the near future. We now look forward to many successful years of system operation and are confident that it will provide an invaluable research tool for scholars in London. We believe that our choice of implementation method

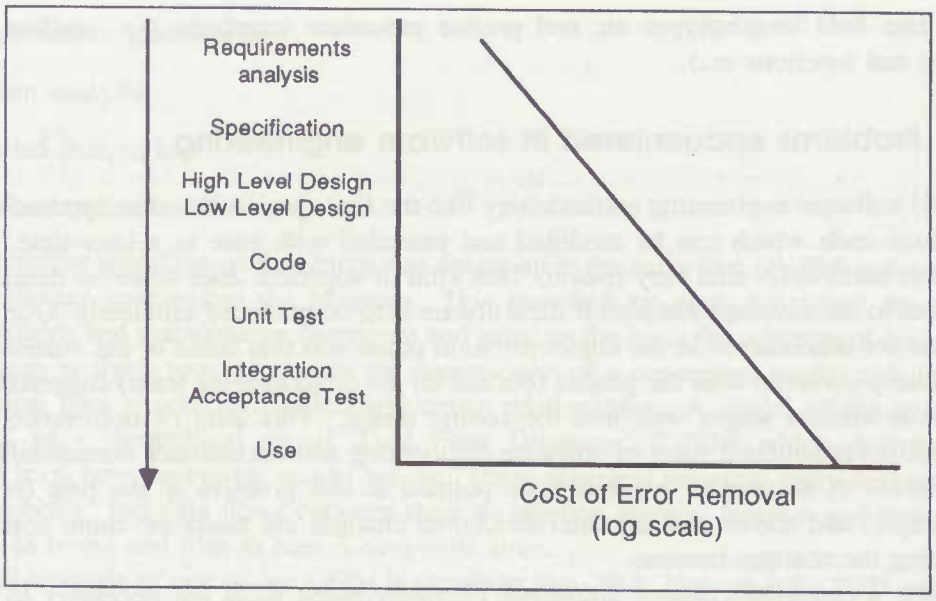

Fig. 29.5: Relationship between cost of error removal/functional change and development stage (Adapted from Cohen *et al.* 1986)

was the right one but it has become clear to us that there are many factors affecting eventual system development efficiency which are often not apparent at the outset. Nevertheless we thoroughly recommend the Constantine-Yourdon approach to other groups wishing to develop similar systems.

Acknowledgements

We are grateful to the Deputy Director of the Museum of London and to the bodies mentioned in the introduction to this paper for their continued support of this project. The project would have been impossible without financial assistance from the GLC, the HBMC, the Baring Foundation and the Pilgrim Trust. We also wish to record our gratitude to the programmers who have put in considerable effort on this project, namely: Leena Ruparelia, Tim Bunce, George Michaelson, Mark Riddoch and Barbara Segal. This paper would also not be complete without an acknowledgement to the substantial level of commitment shown by our indexer Audrey Adams who has taken primary responsibility for painstakingly producing the thesauri and the publications data.

References

COHEN, B., W. T. HARWOOD, & M. I. JACKSON 1986. *The Specification of Complex Systems*, Addison-Wesley, Reading, Mass.

DE MARCO, T. 1978. *Structured Analysis and System Specification*, Yourdon Press, New York.

DIXON, R. 1983. 'A modern computer-cataloguing and administration system for museums', *Int. J. Museum Management and Curatorship*, 2, pp. 335–346.

PAGE-JONES, M. 1980. *The Practical Guide to Structured Systems Design*, Yourdon Press, New York.

RHODES, M. 1986. 'Preparation of the post-excavation archive in london with special reference to finds', *in Dust to Dust*, pp. 25–36, Field Archaeology and Museums Conf. Proc., Society of Museum Archaeologists.