33
The excavation archive as hyperdocument?

Nick Ryan
Computing Laboratory, University of Kent at Canterbury, Canterbury, Kent, CT2 7NF, UK
Email: N.S.Ryan@ukc.ac.uk

33.1 Introduction

Archaeologists today work with many different forms of data and software packages. Different members of excavation and post-excavation teams need both to control and to share parts of large data sets. They must maintain their own working data, access data maintained by others and contribute to the creation of shared material at all levels of the archive.

The variety of software used often leads to intractable data management problems Separate, often inconsistent, copies of the same information may be stored in different formats to suit the needs of the various programs, and of the different specialist activities Ensuring that everyone is working with current data and cross-referencing between the disparate systems remains a difficult and unreliable manual task.

It is now both desirable and possible to integrate many of these information handling tasks in a way that enhances the reliability, quality and utility of information, and supports both the individual and collaborative aspects of the process. Richards has suggested that "...computers offer the prospect of breaking down the traditional excavation/post-excavation divide" and that "One of the most exciting developments would be the application of integrated excavation and post-excavation systems" (1991, 182). Software integration is a necessary step in this direction.

The idea of integration is not new. Despite the separation of the manual processing and recording tasks, context and finds records are now commonly held in the same database. Many organisations are now using networks of microcomputers with multitasking capabilities. With centralised database systems these can provide a suitable platform for more extensive forms of integration. Some have already begun to exploit this potential (see, for example, Williams 1991). Others have developed systems that combine conventional database recording of contexts or finds with graphical display of plans and spatial distributions. Recent examples include that of Semeraro at Lecce, Italy (1992).

Multimedia, hyperdocument and document centred approaches to the assembly and presentation of information have gained popularity in recent years. Whilst they provide interesting models for the presentation of archive material, they are generally less suited to the tasks of acquiring, recording and managing the data. Typical authoring systems start from the assumption that the data already exists and provide mechanisms for imposing a structure upon it. However, the end products are often designed to promote freedom of navigation through the assembled information and, in some cases, to enable users to add their own links between items. This contrasts with the more restrictive nature of most applications built on top of database management systems.

Excavation recording systems need to support both the 'production' oriented tasks of structured data acquisition and exploratory research activities. This argues for a merging of the navigational freedom of hypermedia with the structuring and data management capabilities of conventional database systems. However, excavation recording involves a much wider variety of data types than can be easily handled by a combination of these systems. It is not sufficient simply to extend either type of system or just to develop hypermedia interfaces to conventional databases.

This paper begins by examining several possible approaches to building a modern excavation recording and archive system (Section 33.2). Each approach has desireable features and the central question is how best to combine these characteristics. The emphasis throughout will be on minimising the need to develop special purpose components and maximising the use of existing commercial software.

Section 33.3 discusses the use of stratigraphic diagrams as an aid to navigating through complex excavation data, and Section 33.4 introduces *gnet* version 4, a new version of a system for managing stratigraphic data. This program can be used as a core structuring component of an excavation system in which it is able to work co-operatively with a wide range of other software.

33.2 How to build an excavation system

The excavation and post-excavation processes are cyclical. They involve a sequence of stages from evaluation and preparation, through excavation, analysis to archive production (see, for example, English Heritage 1991). In due course, archive material, reports and publications from earlier excavations provide resources for the assessments of potential and preparations for subsequent excavations. There is considerable overlap between the stages, both chronologically and because many interpretative activities continue, and are subject to revision, throughout the process.

A detailed discussion of the design of any particular system should begin with an analysis of requirements. However, presentation of a detailed requirements specification is beyond the scope of the present paper. Instead, the central concerns here are with the range of

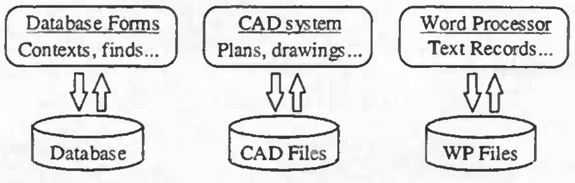


Figure 33.1: In the 'conventional' approach, there are no links between the data handled by different applications.

capabilities that any such system should include and the ways in which the various components should interact.

During the initial stages of evaluation and preparation for excavation requirements include entry and cross-referencing of text and bibliographic material. Most information will be in the form of notes and summaries of external printed sources, although sometimes a facility for capturing and storing plans and photographs, and overlaying information on base maps will also be needed.

Recording requirements during excavation include contexts, finds and basic details of environmental sampling. Plans, drawings, and photographs are also produced and must be catalogued and cross-referenced with contexts, finds and each other. The system should provide facilities for the capture of this information by digitising or scanning. In addition, the introduction of Photo-CD may lead to a rapid increase in the storage of photographs in digital form.

To support the many specialist activities, an information system should provide storage of and access to reference and other private material, and the production of public (i.e. within the unit) summary information for use by others. This distinction is important because few specialists would accept wider access to their working data, with the attendant danger of misinterpretation by non-specialists. However, once they have produced summaries, these should be immediately available to all who need them.

The final product of any excavation must be a coherent archive. It should be designed to serve a wide range of information needs from high level summaries for public consumption to the detail required by many researchers..

The following sections examine various approaches to the software architecture of a system to support these varied requirements. It should be able to support the data collection, exploration, analysis and presentation needs of the excavation team throughout the process.

33.2.1 The 'conventional' approach

Many of the excavation recording systems in use today are based on a variety of software, often one that has been assembled over many years. Each different tool performs a separate task, largely unrelated to that of the others (Figure 33.1). Typically a database management system with purpose-written application programs provides facilities for data entry and the storage of structured records covering contexts and finds. Additional catalogues may also be included to provide easy reference to other important material such as photographs, plans, and sampled material. Written reports, both in final form and as interim working documents, are usually produced with word processing software. The more technologically advanced excavators may also produce drawings using CAD software, store photo-graphs on Photo-CD and use a DTP package for producing publications.

The main benefit of this approach is that if the software has been chosen wisely each tool is used for the task to which it is best suited. Problems are unlikely to arise from trying to store and manipulate information using an inappropriate technology.

During the data collection phase most effort will be directed to filling in forms, preparing drawings, cataloguing finds, plans and photographs. Typically these activities will involve each person working for long periods with a single application, only occasionally needing to consult other types of stored information. However, when cross-referencing is required, as is increasingly the case as a project moves into the post-excavation phase, this convenient compartmental strategy begins to break down and becomes instead a major limitation.

The very separation of the different data types make it difficult to move smoothly between closely related but different representations of important information. For example, when viewing a particular context record it will usually be possible to see related finds records if they are also managed by the same database system. Viewing plans showing physical location, or photographs, or a textual description of a group of related contexts in a partially written report is not always so straightforward. Often this can only be achieved by starting a different application and then searching manually for the required information.

A few years ago when almost all such applications ran in a single-tasking environment this could be extremely tedious. The advent of affordable, if limited, multi-tasking environments has eased this problem and it is now possible to run several of the required applications simultaneously on a conventional desktop machine. This is a considerable step in the right direction, but the link between applications and between different types of data must still be made by the user. The lack of integration prevents automation of searching, or the recording of intelligent links between items.

Many modern packages can import files generated by a wide range of other systems. Word processors and DTP packages can import tabular data from common database packages as well as a variety of vector and raster graphics formats. Similarly, many database packages can now import, store and export graphical and textual data. At first sight this may appear to be a solution, but unfortunately it introduces further problems.

Importing data from another source means making a separate copy of the original. In due course, the original may be changed, but those working with copies may be unaware of changes which could have a material impact on their own work. The need to reconcile differences and, indeed, the high possibility of people working with and deriving conclusions from incomplete or incorrect material, highlight the limitations of this approach.

Although many database systems can now store a wide range of data types, most do so only as simple binary objects. To date, most DBMS have no understanding of the behaviour of more than the most basic data types and thus no facilities for editing or querying these objects (Ryan 1991; Cheetham & Haigh 1991; Stonebraker & Rowe, 1986). It is therefore difficult to build applications that support effective centralised management of all data types.

33.2.2 Hyperdocument approach

The hyperdocument approach is normally based on a single program designed for both authoring and presentation of documents (Figure 33.2). Typical examples include Guide and Hypercard. Early hypertext systems enabled arbitrary links to be inserted in a textual document so that it was possible to look up the meaning of a word in a glossary, or to jump to another related page of text following a predetermined network of links. Modern hyperdocument systems have taken this approach further and many can now handle a wide variety of text, graphics, database records and multimedia data derived from a variety of other sources

The major attraction of this approach is the ease with which a document author can add links between related objects. Unlike the relationships in a database, these links do not depend on predetermined structure or on stored data values and a typical link will lead to only one other object. In other words, links can represent arbitrary relationships, perhaps discovered in the process of browsing through or analysing the stored information. Planning the link structure is perhaps the most challenging task for a hyperdocument author.

The arbitrary complexity of a hyperdocument means that it can be very easy to get lost. Much work has been done on easing these navigational problems, and many systems provide additional information to help users find their way around. Amongst the possible navigational aids are history lists that show how the user reached the current item, and directed graphs showing the arrangements of links in the vicinity of the current page. These act as a 'road map' that can help the user to visualise where they have come from and where to go next. Clearly, when dealing with such a large and complex set of data as that recorded during an excavation, reliable aids to navigation are essential.

Grouping material together within a single program helps to ensure that all aspects of the data can be accessed through a single, consistent user-interface. Those whose work mostly involves one type of data do not need to learn a different style of working in order to understand other, less frequently used, programs. However, this argument is equally applicable to separate programs written to a

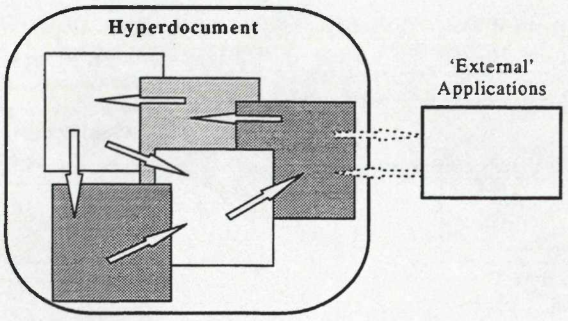


Figure 33.2: In the hyperdocument approach different data types may be linked under the control of a single application. Some systems provide mechanisms for linking with data managed by other software.

common interface standard. Whilst few would dispute the desirability of maximising the common components of an interface, the fact remains that working with different types of data often requires different techniques. There are only so many operations that can be shared by, for example, data entry forms and a CAD package.

It is certainly possible to produce data collection tools using hyperdocument systems, but the limited data management facilities built into most systems restrict this to relatively simple applications. Although some hyperdocument systems have used databases as a storage and retrieval medium (see, for example, Becker & Rowe 1991), such facilities are not yet widely available. Data management facilities and others such as management of CAD data may be accessible through links with other software. When used in this way, the approach takes on some of the features of other approaches described below. Overall, however, the design of most hyperdocument systems supports imposition of structure on a *pre-existing* collection of data, thus emphasising their presentation role rather than use for data collection and management.

33.2.3 Complex document approach

The approach which, for convenience, I refer to here as the 'complex document' includes several techniques that stress the incorporation of different data types within a single enclosing document. Typically, such systems are based on a communication technology such as Microsoft Object Linking and Embedding (OLE) or similar mechanisms developed by IBM for OS/2 and Apple for the Macintosh.

Whereas a simple hyperdocument incorporates all material, whatever its source, within a single application, this approach uses other applications as servers to manage the different data types (Figure 33.3). Thus, for example, a drawing package is automatically invoked whenever a graphical object is selected. In this respect it shares with the conventional approach the benefit of applying the most suitable tool to manage each different data type.

There is much less potential for navigational problems with this approach, if only because the structure of these

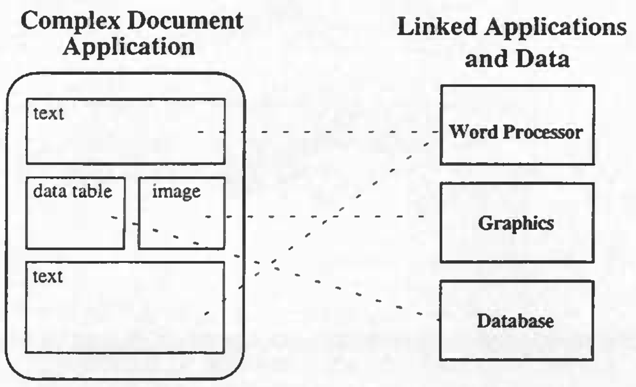


Figure 33.3: The complex document approach provides active links to other applications, or embedding of snapshots.

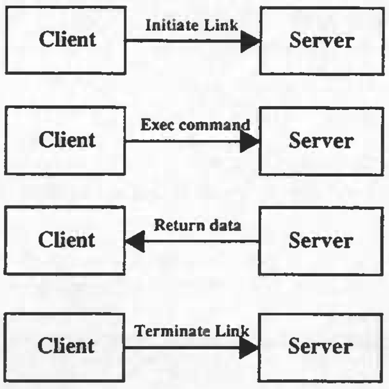


Figure 33.4: A simple DDE conversation sequence (top to bottom). Acknowledgement and error messages omitted for clarity.

documents is typically more regular and less complex than that of many hyperdocuments. A hyperdocument link is a link *between* objects, permitting an arbitrary graph structure in which loops are possible. Here, the links *include* or *embed* objects *within* the enclosing document. Objects nested within objects may form a hierarchical structure but typically such nesting is shallow.

Embedded objects represent copies that are unaffected by subsequent changes to the original. The including or linking approach is in some ways more interesting because it inserts a reference to the external object in the enclosing document. Whenever the document is viewed, the current version of the linked object is accessible.

Again the emphasis of most applications that use these methods is on collecting together and imposing a structure on a variety of existing data sources in order to provide a coherent single presentation. As with hyperdocument systems, there is no inherent reason why a data collection system should not be built in this way, rather that the emphasis is on presentation. As more software able to use this technology becomes available, it may become an acceptable medium for developing simple information systems. However the predominantly hierachical structure that it encourages may limit its applicability.

33.2.4 Single complex application

If none of the alternatives provide a suitable solution, why not develop a special program to handle all of our requirements? Rather than being constrained by the limitations of other software, or having to purchase complex programs with unwanted functions, this could be designed to fit the requirements exactly.

Some program development is invariably required even if most can be achieved using database 4GLs and other 'high-level' development tools. There will always be special requirements that cannot be met directly by commercial software. If programs must be developed to deal with these, then why not extend them to cover most other requirements?

Unfortunately this approach often leads to uncontrolled growth with new features added to an old system as needed. Such applications become increasingly more difficult to maintain and adapt to new requirements. Invariably there will be many compromises, and attempts to extend the system may continue long past the point when it should really have been completely re-designed.

The availability of powerful and easy to use programming environments, such as Visual Basic, together with libraries of suitable re-usable components may encourage adoption of this approach. Such development should be subjected to careful control with an emphasis on providing functionality that is not available in other software. There can be no justification for spending limited archaeological time and money on developing modules that merely reproduce the functionality of existing software. The system described by Rains elsewhere in this volume provides an excellent example of what can be achieved by the carefully controlled application of this approach.

33.2.5 Multiple communicating applications

As with complex documents, this approach is based on a communication technology, but in this case the communication requirements are simpler. All that is required is a mechanism whereby one application can initiate a conversation with another, issue a few simple commands to open documents and locate objects, and terminate the conversation. Inter-process communication and remote procedure call (RPC) mechanisms are well established facets of operating systems such as UNIX but, more recently, they have become available to microcomputer programmers. For systems using Microsoft Windows, the Direct Data Exchange (DDE) mechanism provides the required functionality.

Many Windows based applications can act as a DDE server or client; some can perform both roles. Most servers permit remote execution of commands that are normally invoked interactively by menus (Figure 33.4). For example, a word processor could be instructed to open a specified file, replace occurrences of one word by another, then save and close the file. Commands may also return data to the remote client application. A query can be passed to a database system as part of a command and the result returned to the client application. Many Windows applications with a macro or other programming language can act as DDE clients. Most include statements that can initiate DDE conversations and exchange commands and data with a server.

Provided the applications used support this form of communication it is possible to construct information systems that make extensive use of existing software. Purpose-written programs are then required only for tasks that cannot be performed by off-the-shelf software. Only minimal programming skills are required to set up the link mechanisms between the component applications. Typically this involves writing one small function or procedure for each required communication link. The task is well within the capabilities of users with some design and programming skills and could be performed by anyone with experience of setting up a simple excavation record database.

33.3 Navigation by stratigraphy

A core requirement of an integrated information system is the ability to create links between alternate representations and related information of different types. For example to access a stratigraphic diagram while viewing context or finds records, or to view text notes, plans or photographs relating to the contexts displayed by a stratigraphy editor. The ability to move easily between alternate representations, or between related data items is reminiscent of the link mechanism of a hyperdocument. Such navigational and exploratory capabilities are exactly what is needed in this type of information system.

Where the requirements differ from a conventional approach to hyperdocuments is that many links between items are present as structural or value-based relationships in the underlying database. A hyperdocument system provides the means to link essentially static frames of information. In the type of system envisaged here it should be possible for the user to add links for their own purposes, but many others will be derived dynamically from, and maintained by, the database.

The use of a directed graph as a 'road map' was mentioned above as a navigational aid employed in some hyperdocument systems. Such a map already exists, either implicitly or explicitly, within any excavation information system. The stratigraphic diagram is a directed graph and, because of its central importance in linking contexts, can be used for similar purposes in an excavation information system.

Most current tools for analysis of stratigraphic sequences (e.g. Herzog 1993) or display and editing of stratigraphic diagrams (e.g. Ryan 1988; Boast & Chapman 1991) are separate from or, at best, loosely integrated with excavation databases. To fulfil this navigational requirement, they need to be more closely integrated with the collections of related data, and extended to provide support for nested and overlapping groups of contexts.

The techniques used in stratigraphic diagramming tools can also be used to support a wider range of excavation and post-excavation analyses. Andresen and Madsen (1991) have emphasised the need to preserve the integrity of original observations and to maintain their separation from subsequent interpretative constructs. This has long been standard practice with manual recording, but is all too easily subverted in computer based systems. In the same article, they showed how such constructs may be represented in a relational data model as links or relationships between objects, both contexts and finds. It is no coincidence that the link structure they chose is one that can also be used to represent stratigraphic relationships between contexts. Stratigraphy is just one example of their idea of construct, albeit one that is established during excavation rather than in later interpretative stages.

The objects linked together by interpretative constructs may include other constructs and thus they may be combined into hierarchies. The construction of the structural history of a site may produce many such hierarchies, often overlapping each other where, for example, a ditch continues to mark a boundary through several phases. Some of these constructs will be temporary, serving only to test ideas that are later abandoned. Others will form an integral part of the interpretations recorded in the site archive. As with other information in the system there will be both private constructs used by individual specialists in their detailed work, and public constructs used to encapsulate structuring interpretation of interest to others.

The excavation archive can therefore be envisaged as a large collection of different types of data, interconnected by a number of overlapping networks of interpretative constructs. A diagram editor that can be coupled with database systems and other tools could provide a means of navigating through this complex structure. A prototype of such a system is described in the next section.

33.4 *gnet* version 4

Despite the use of an old name to emphasise its ancestry, *gnet* version 4 is an entirely new program. It has been built to address several limitations of earlier versions and to explore how the stratigraphic diagram and related representational forms can be used as core components of an excavation information system. Earlier versions ran on a variety of hardware and operating systems (Ryan 1988). The most well-known of these ancestors ran under MS-DOS and was subject to many of the constraints of that environment (Madsen 1990). Comments from many users indicated that its main limitations were the number of nodes (contexts) and edges (stratigraphic relationships) that could be handled in a single diagram, and the lack of an effective 'undo' mechanism.

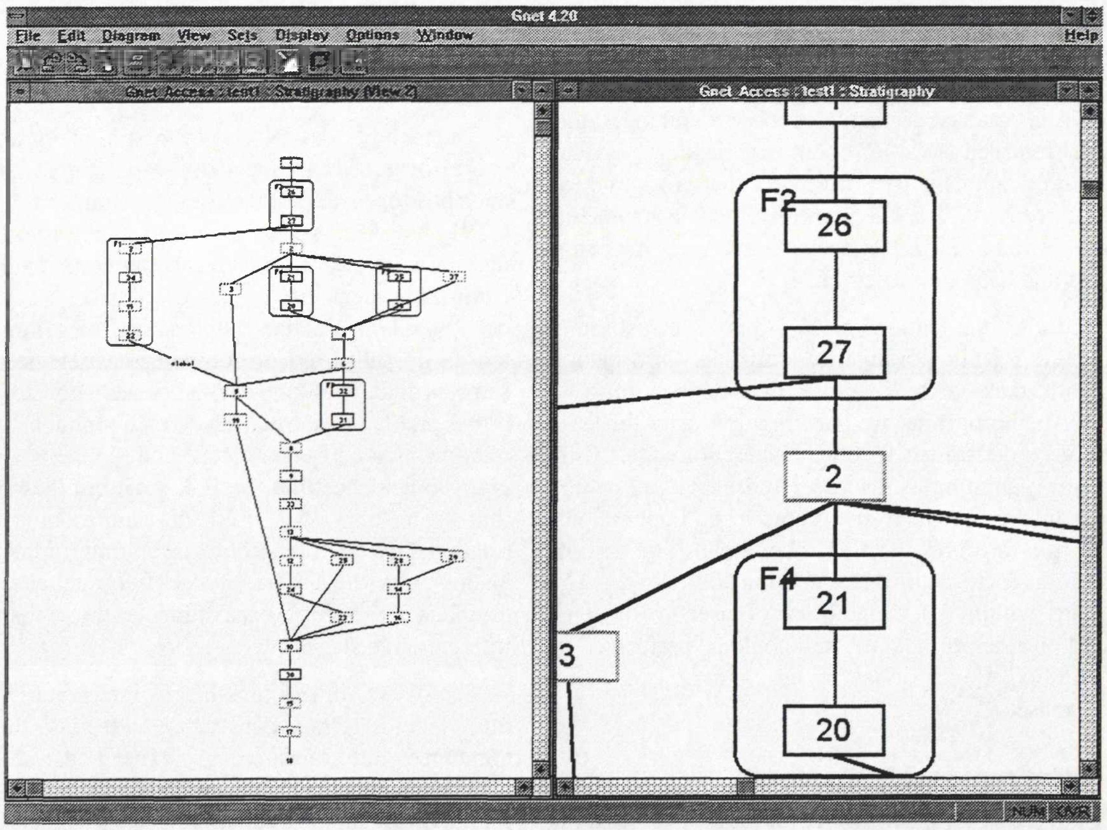

Gnet 4.20
File Edit Diagram View Sets Display Options Window Help
Gnet Access : test1 : Stratigraphy (View 2)
Gnet Access : test1 : Stratigraphy
F2
26
27
2
F4
21
3
20
NUM OVR

Figure 33.5: Typical *gnet* display: the right hand window shows an enlarged section of the diagram.

Experimental versions exploiting various extended memory techniques have been tried in order to increase the number of contexts that could be handled. None of these approaches offered more than a few of the benefits available by moving to a Windows environment, and attempts to expand earlier versions have long been abandoned.

The principal distinguishing features of this new version are its greater capacity, different appearance, its method of data management, the provision of additional modules to display other representations of data, and an ability to work co-operatively with other applications. The increase in capacity is entirely due to the ability of Windows programs to use extended memory and avoiding the limitations of MS-DOS.

As to its appearance, *gnet* uses the Windows Multiple Document Interface (MDI) to provide views of one or more stratigraphic diagrams each in a separate window. These views can be scaled independently so that it is possible to view an enlarged part of a diagram in one window, whilst the overall diagram is displayed in another (Figure 33.5). These views may also be used to display different aspects of the same information side by side. For example, one window might show all recorded relationships between contexts, whereas a second could show only those necessary to establish the stratigraphic sequence.

Because *gnet* is essentially a general purpose system for manipulating directed graphs that has been enhanced with specifically archaeological functions, its application is not limited to stratigraphic diagrams. Additional relationship types may be defined to support the constructs discussed in the preceding section. Once such a construct has been defined and documented, it can be used in the same way as any other relationship in the database. Given constructs called, say, 'Phase_1' or 'Hoard_5' it will be possible to construct queries to display context records or plans of all members of Phase_1, and to highlight contexts containing coins from Hoard_5 on the stratigraphic sequence diagram.

Most earlier versions of *gnet* have used simple text files to store details of nodes, links and the definitions of their appearance. Version 4 uses the Microsoft ODBC library to provide a consistent interface to several different database management systems. It will therefore work with any DBMS for which suitable driver software is available. At the time of writing, these include dBase, Access, Paradox, Ingres, Sybase and Oracle. The program generates all of the tables needed for its own operation, and can import data from the text files used by earlier versions. In addition to the usual data management benefits gained from using a DBMS, this approach means that *gnet* can be integrated more easily with existing excavation databases. In the past it has always been necessary to write special routines to export node and edge

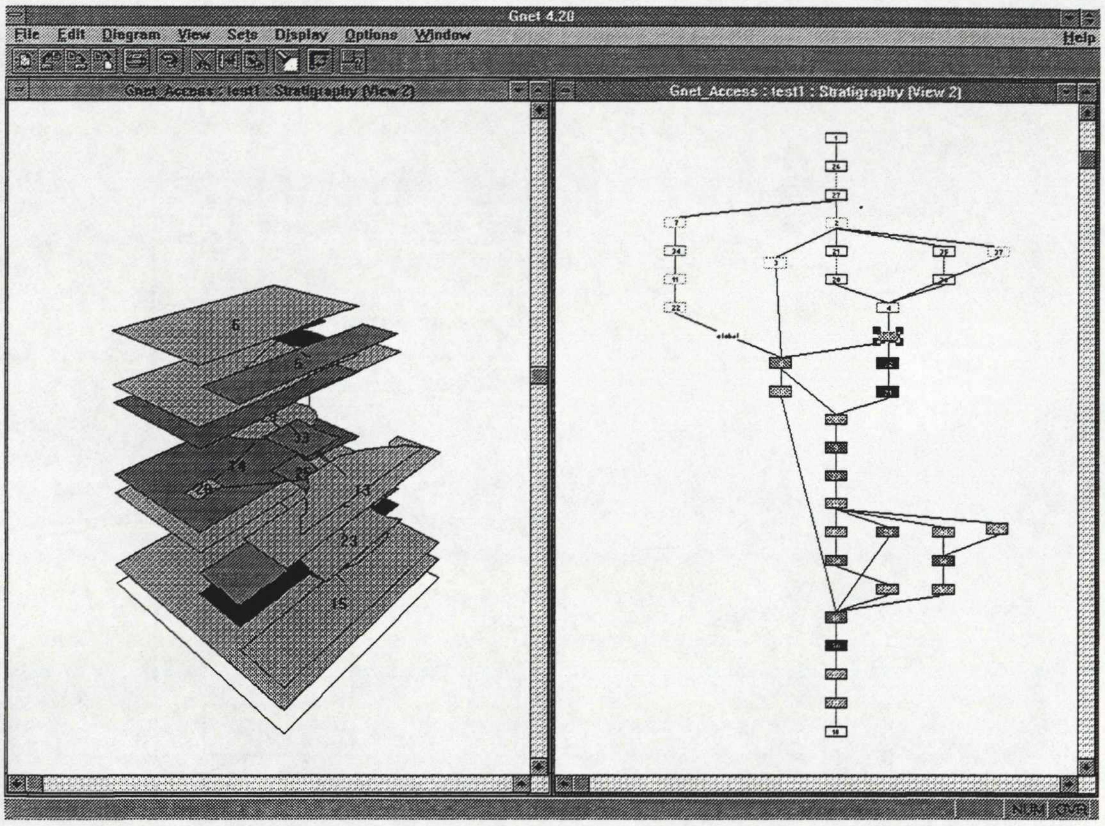

Gnet 4.20
File Edit Diagram View Sets Display Options Window Help
Gnet_Access : test1 : Stratigraphy (View 2)
Gnet_Access : test1 : Stratigraphy (View 2)
NUM OVR

Figure 33.6: *gnet* display showing stratigraphic diagram and associated 3D display of context outlines. The 3D display shows only the highlighted contexts in the stratigraphic diagram.

data from a database for use by the program, and *vice versa.*

The design combines elements of the single complex application and multiple communicating applications approaches. In addition to the directed graph representations of stratigraphic relationships and other analytical constructs, the version presented here also includes a simple module for displaying two and three dimensional models of the excavated contexts (Figure 33.6). This was included because, at the time, a suitable CAD system with DDE capabilities was not available to perform these tasks. However, unlike most CAD systems, the outline shapes of contexts are stored in the same database as the stratigraphic information. A second minor module provides a simple text editing facility for making notes and attaching them to diagrams.

Both of these additional components were constructed using re-usable components, the first from another software project, the second from a commercial library. Work is under way to add further modules providing database access to enable arbitrary queries, and a image viewer. In both cases these use existing re-usable components whose functionality can be made available with a minimum of effort. Indeed, in most cases adding such components involves little more than adding a suitable menu command. The database access module is somewhat more complex because it requires registration of any data tables of interest together with the SQL statements needed to link them to the stratigraphic node data.

These minor components have been included to increase the utility of the system when used as a stand-alone application, or in combination with existing context and finds databases. However, far greater functionality can be gained by making use of other applications to provide these necessary services. Linkage to other software is provided by a simple DDE mechanism. For an application to support communication with *gnet* it must provide commands accessible through a DDE conversation to enable it to open a document, search for and display a named object. Similarly, *gnet* responds to a similar set of commands enabling it to open a named diagram and to find and display named nodes. In due course, access to many of the program's interactive commands will be made available for use in DDE conversations. The present set is, however, quite adequate to provide a simple equivalent of a hyperdocument link between, rather than within, applications.

To illustrate how these links work, Figure 33.7 shows a typical *gnet* display representing the stratigraphic relationships in a small trench. One node (context 2) has been selected and this is indicated by the enclosing rectangle with 'sizing handles' at each corner. Many operations can be applied to one or more selected nodes either through conventional menu selections, or from the context-sensitive menu shown whenever the right mouse

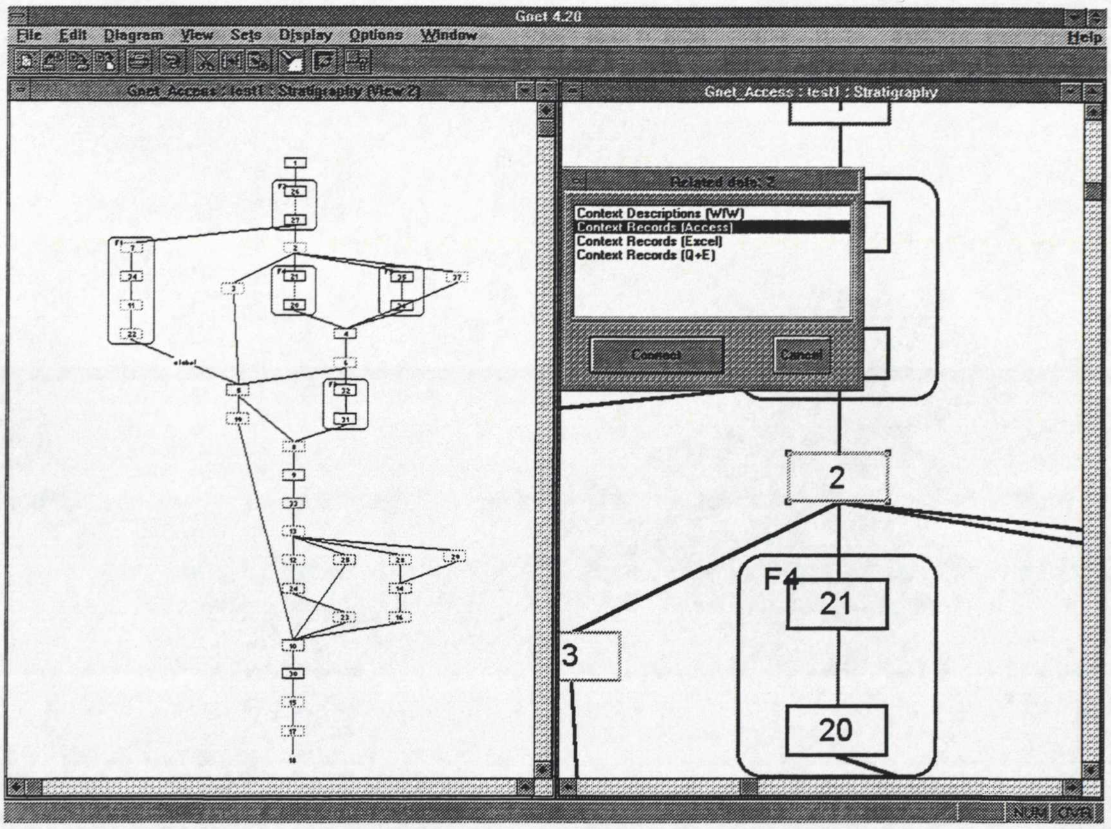

Gnet 4.20
File Edit Diagram View Sets Display Options Window
Help
Gnet_Access : test1 : Stratigraphy (View 2)
Gnet_Access : test1 : Stratigraphy
Related data: 2
Context Descriptions [W/W]
Context Records [Access]
Context Records [Excel]
Context Records [Q+E]
Connect
Cancel
2
F4
21
3
20
NUM OVR

Figure 33.7: Following a link from a context in the stratigraphic diagram to the associated context record in an Access database.

button is clicked. The 'Related Data' item in this menu leads to a sub-menu showing all registered sources of linked data.

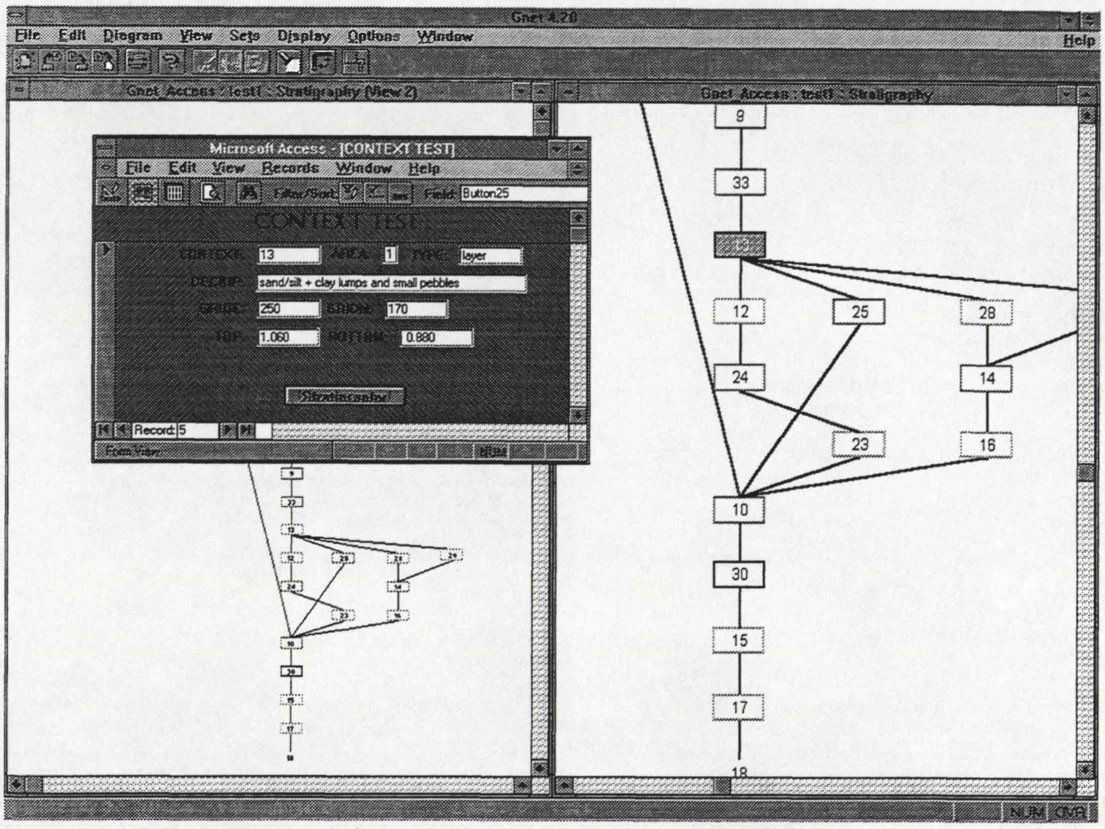
Details of all these related data items, the programs that provide data services and commands that are needed to request these services are held in an initialisation file. In later versions some of this information will be moved to the main database and greater use will be made of the Windows 'Registration Database' to identify available services. Selecting the item 'Context Records [Access]' initiates a DDE conversation with the Access DBMS. If not already running, Access is started and a sequence of commands are sent to open the database, display the context record form and seek to the record corresponding to the selected node in the *gnet* diagram.

Once the form is displayed within Access, the user may choose to remove the search filter constraint imposed by *gnet*, and browse through the database. The form includes a button labelled 'Stratigraphy'. When this is pressed another DDE conversation is started that requests *gnet* to locate and display the node corresponding to the context currently shown on the form (Figure 33.8). The button invokes a simple function written in Access Basic (Figure 33.9) that connects to the *gnet* service and issues 'open' and 'search' commands similar to those passed from *gnet* to Access. Thus it is possible to move freely between the stratigraphic diagram and database representations of contexts through a mechanism reminiscent of a hyperdocument link.

Similar links have been established with several other applications, including Microsoft Word documents, Excel spreadsheets and the Q+E database interface. In principal, the system can be extended to include any application that supports equivalent DDE commands. The examples shown here concentrate on the central role of the stratigraphic diagram as a core structuring component at the heart of an excavation information system. However, it is equally straightforward to create similar links between the other applications. For example, similar Access Basic and Word Basic functions to those used to communicate with *gnet* can be written to establish links between Access and Word. A complete network of such links would provide considerable navigational freedom between multiple representations of excavation data in a manner analogous with the links of a hyperdocument. Unlike a conventional hyperdocument, however, this approach ensures that it is always possible to use the most appropriate tool for any particular task.

33.5 Conclusion

Each of the approaches described in Section 33.2 might be employed in building parts of an excavation recording and archive system. The multiple communicating applications approach was used to inform the design of *gnet* version 4, because it appears to best combine several important


Gnet 4.20
File Edit Diagram View Sets Display Options Window Help
Gnet Access : test1 : Stratigraphy (View 2)
Gnet Access : test1 : Stratigraphy
Microsoft Access - [CONTEXT TEST]
File Edit View Records Window Help
Filter/Sort:
Field: Button25
CONTEXT TEST
Stratigraphy
Record: 5
Form View
NUM
NUM OVR
9
33
13
12
25
28
24
14
23
16
10
30
15
17
18

CONTEXT: 13 AREA: 1 TYPE: layer
DESCRIP.: sand/silt + clay lumps and small pebbles
GRIDE: 250 GRIDN: 170
TOP: 1.060 BOTTOM: 0.880

Figure 33.8: Following a link from a context record in an Access database to the associated context in the stratigraphic diagram.

benefits of the other approaches. It emphasises the use of appropriate existing tools rather than their unnecessary replication, and it provides a mechanism analogous with hyperdocument linkage to provide structure and to enable navigation between related representational forms.

The same mechanism can also be used to support internal structural links between applications that display different representations of information stored in a database. As with the complex document approach it can support links with 'live' data rather than merely including copies of data managed by other programs. Finally, it is an incremental approach in that new components can be added with minimal disruption. The only requirement being that they support the chosen communication method and implement a few very basic commands.

The stratigraphic and other diagrams provided by *gnet* can act as important navigational aids in addition to their more normal analytical roles. However, *gnet* is only one component in such a system and the principal of linkage between applications can be applied whether or not it is used. In almost all cases this linkage can be achieved using very simple functions written in the programming or macro languages build into many modern applications.

References

ANDRESEN, J. & MADSEN, T. 1991. 'Data structures for excavation recording: a case for complex information management', in C. Larsen (ed), *Sites and Monuments: National Archaeological Records*, National Museum of Denmark, Copenhagen, 49–67.

BECKER, B. S. & ROWE, L. A., 1991. 'HIP: a hypermedia extension of the picasso application framework', *Proc. NIST Advanced Information Interfaces: Making Data Accessible*, June 1991.

BOAST, R. & CHAPMAN, D. 1991. 'SQL and hypertext generation of stratigraphic adjacency matrices', in K. Lockyear & S. Rahtz (eds), *Computer Applications and Quantitative Methods in Archaeology 1990*, BAR International Series 565, Tempvs Reparatvm, Oxford, 43–51.

CHEETHAM, P. N. & HAIGH, J. G. B. 1991. 'The archaeological database – new relations?', in G. Lock & J. Moffett (eds.), *Computer Applications and Quantitative Methods in Archaeology 1991*, BAR International Series 577, British Archaeological Reports, Oxford, 7–14.

ENGLISH HERITAGE 1991 *The Management of Archaeological Projects*, Historic Buildings and Monuments Commission for England.

HERTZOG, I. 1993. 'Combining stratigraphic information and finds' in J. Wilcock & K. Lockyear (eds.) *Computer Applications and Quantitative Methods in Archaeology 1993*, BAR International Series 598, Tempvs Reparatvm, Oxford, 109–114.

MADSEN, T. 1990. 'Stratigrafianalyse og EDB', *KARK Nyhedsbrev*, nr 3, Moesgård, 15–36.

RAINS, M. 1994. 'Towards a computerised desktop' (this volume, 207–210).

```
' CallGnet
'   Simple DDE conversation with Gnet used to locate
'   the current context in a stratigraphic diagram.
'   The parameter 'diagram' contains required Diagram name
'   Returns 0 if DDE conversation not started, else 1
'
Function CallGnet (diagram As String)
    On Error Resume Next                        ' Set up error handler.
    Dim channel                                 ' DDE channel
    Dim F As Form                               ' CONTEXT form
    Dim ctxt As String                          ' Current context id
    Dim opencmd As String, findcmd As String    ' DDE commands

    Set F = Forms!CONTEXT                       ' Get id of current context into ctxt
    ctxt = F![CONTEXT]

    opencmd = "[Open " + diagram + "]"          ' Construct Gnet commands to open diagram
    findcmd = "[Find " + ctxt + "]"             ' and locate the context

    channel = DDEInitiate("Gnet", "SYSTEM")     ' Initiate DDE channel
    If Err Then                                 ' If error, Gnet isn't running
        Err = 0                                 '    Reset error
        task = Shell("Gnet", 1)                 '    and start Gnet.
        If Err Then                             '    If it fails to start
            CallGnet = 0                        '        Return value is 0
            Exit Function                       '        Exit
        End If
        channel = DDEInitiate("Gnet", "System") '    Initiate DDE channel
    End If

    DDEExecute channel, opencmd                 ' Open the Gnet diagram. Gnet ignores this
                                                ' command if the diagram is already open.
    DDEExecute channel, findcmd                 ' Exec remote command to find required context
    DDETerminate channel                        ' Close DDE channel (leaves Gnet running)
    CallGnet = 1                                ' Return value is 1
End Function
```

Figure 33.9: Access Basic function used to establish link with *gnet* and locate a context within a stratigraphic diagram.

RICHARDS, J. D. 1991 'Computers as an aid to post-excavation interpretation', in S. Ross, J. Moffett & J. Henderson (eds.), *Computing for Archaeologists*, Monograph no. 18, Oxford University Committee for Archaeology, Oxford, 171–186.

RYAN, N. S. 1988. 'Browsing Through the Stratigraphic Record', in S. P. Q. Rahtz (ed.) *Computer Applications and Quantitative Methods in Archaeology 1988*, BAR International Series 446, British Archaeological Reports, Oxford, 327–334.

RYAN, N. S. 1991. 'Beyond the relational database: managing the variety and complexity of archaeological data', in G. Lock & J. Moffett (eds.) *Computer Applications and Quantitative Methods in Archaeology 1991*, BAR International Series 577, British Archaeological Reports, Oxford, 1–6.

SEMERARO, D. 1992. 'The excavation archive: an integrated system for the management of cartographic and alphanumeric data', in J. Andresen, T. Madsen, & I. Scollar (eds.), *Computing the Past: Caomputer Applications and Quantitative Methods in Archaeology, CAA92*, Århus University Press, 1992, 205–212.

STONEBRAKER, M. & Rowe, L. A. 1986. 'The design of POSTGRES', *Proc. ACM SIGMOD International Conference on Management of Data 1986.*

WILLIAMS, T. 1991. 'The use of computers in post-excavation and publication work at the Department of Urban Archaeology, Museum of London', in S. Ross, J. Moffett & J. Henderson (eds.), *Computing for Archaeologists*, Monograph no. 18, Oxford University Committee for Archaeology, Oxford, 187–200.