



# Kent Academic Repository

Ryan, Nick S. (2001) *jnet, a successor to gnet*. In: Borner, W., ed. *Archaeologie und Computer, Workshop 6*. .

## Downloaded from

<https://kar.kent.ac.uk/13525/> The University of Kent's Academic Repository KAR

## The version of record is available from

## This document version

UNSPECIFIED

## DOI for this version

## Licence for this version

UNSPECIFIED

## Additional information

## Versions of research works

### Versions of Record

If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

### Author Accepted Manuscripts

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in *Title of Journal*, Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

## Enquiries

If you have questions about this document contact [ResearchSupport@kent.ac.uk](mailto:ResearchSupport@kent.ac.uk). Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our [Take Down policy](https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies) (available from <https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies>).



# **jnet: a successor to gnet.**

Nick Ryan

*n.s.ryan@ukc.ac.uk*

*Computing Laboratory, University of Kent at Canterbury, Canterbury, CT2 7NF, UK.*

---

## **Abstract**

After several years of hoping that more recent programs would allow his graph tool *gnet* to be quietly forgotten, the author has finally bowed to requests from many colleagues to resurrect the project. This paper reviews some aspects of the history of stratigraphic software and presents an early prototype of *jnet*, a successor to *gnet* written in Java. *jnet* is intended to offer similar facilities to its predecessor, but with significant improvements in database connectivity, interoperability with other programs, and to enable access from anywhere. Whereas *gnet* was limited to Windows platforms, *jnet* may be used on a variety of desktop and mobile systems as well as through a Web interface.

## **Background: a brief history of computerised stratigraphy**

In what now seems like an earlier age of archaeological computing, the application of computers to the manipulation and analysis of stratigraphic sequences probably began with Wilcock's *STRATA* program (1975). Although it now seems primitive, *STRATA* demonstrated that a computer program could be used to derive a logical sequence from a collection of relationships between stratigraphic layers. The appearance of this program also led to the start of a debate about the appropriate use of computer based techniques in stratigraphic sequencing and, more generally, in excavation recording that was to continue for some time.

*STRATA*'s input was a complete set of recorded observations of stratigraphic relationships, and its output was a complete sequence for the site. It was designed as a batch process in which all of the available data was interpreted in a single run to produce a deterministic solution. This 'black-box' approach was seen by many as antithetical to the process of developing an understanding of a site and its stratigraphy during excavation (see, for example, Harris 1975). In practice, contemporary hardware and software limitations severely restricted the size of the models that could be handled and processing even very small sites might take several hours. However, it was to be some time before interactive computer methods could offer an alternative approach.

A decade later, two papers presented at the 1985 CAA Conference prepared the foundations for several subsequent developments (Haigh 1985; Ryan 1985a). Haigh, brought a mathematician's view to sequencing by identifying the problem as the ordering of a partially ordered set, or poset. Ryan arrived at a similar position from a computational perspective. He had recently been working on the related problems of drawing genealogical diagrams and graphical representations of computer file stores. He observed that existing system software, the UNIX 'topological sorting' program, *tsort*, was used to solve a similar problem in traversing and ordering the contents of a hierarchical file store. With simple extensions to handle cross-links in a hierarchy—symbolic links in a UNIX filestore, marriages linking lineages in genealogy—the core algorithm of *tsort* became the basis for *gtree*, a generalised program for drawing and manipulating tree-like data structures (Ryan 1985b).

Whilst some concentrated on the stratigraphic diagram, or 'Harris Matrix', as a distinct issue, others began to develop systems to integrate relationships and sequences into more complete excavation recording and analysis tools. Rains, for example, introduced the forerunner of his integrated archaeological database at this time (Rains 1985). Soon after, Alvey presented his *Hindsite* program which used AutoCAD to maintain single-context plans together with a representation of the

relationships between layers (Alvey 1989). Both of these systems represented significant early contributions to the development of excavation recording and visualisation.

Again in 1990, the CAA Conference proceedings included several papers on computer processing of stratigraphic data. Boast and Chapman (1991) presented an approach based on SQL queries against a database, Desachy and Djindjian (1991) discussed a simple sorting algorithm, whilst Huggett and Cooper (1991) reviewed the current state of the art and discussed the practical utility of various approaches. In the same volume, Herzog and Scollar (1991) introduced a fully automated system for producing stratigraphic diagrams. Though able to handle more realistic data volumes with reasonable speed, Herzog's program followed Wilcock's earlier approach of producing a solution as the output of a batch run. However, improving technology was later to make it realistic to run the program whenever new data became available during excavation. In this way, the sequence diagrams produced by the program could evolve as excavation progressed. The main limitation of this approach was that the excavator could exert little influence over the final form of the diagram.

Following Harris and others' earlier reservations about 'black box' methods, Ryan (1998) had developed an interactive system called *gnet*. The design thinking behind *gnet* was quite different to that behind Herzog's program. Whilst accepting that automated layout and the ability to print stratigraphic diagrams were important capabilities, the ability to interact with and explore the diagram—what we would now call visualisation—were seen as the key to supporting the excavator's need to develop and maintain an intimate understanding of the structure of the site as excavation progressed. As Herzog and Scollar noted (*ibid*), however, the program required what was then advanced hardware. Initially, it ran only on UNIX workstations, and even the subsequent PC version required a mouse and a specialised graphics card, items that few archaeologists possessed at that time. Fortunately, time was to make this a less restrictive requirement.

*gnet* was a development of an earlier program, *gtree*, redesigned to handle networks or lattices as its primary data structure, rather than the trees with added cross-links of its predecessor. It was, in fact, a general-purpose graph browser/editor that could be configured to suit a wide range of applications from stratigraphy and genealogy to software engineering design processes. Much of its development was, however, informed by archaeological requirements as this had rapidly become its most widely used application. Intensive development, in collaboration with excavators in several countries, took place in the first half of the 1990s.

Beyond simple error checking and interaction with diagrams, the final system (Ryan 1995) offered a variety of facilities aimed at post-excavation tasks, including methods for grouping contexts to form phase diagrams and other high-level interpretive abstractions. Multiple views could be displayed simultaneously. It was possible to switch rapidly between a complete graph showing all recorded links and those that formed the stratigraphic sequence, or between grouped and expanded views. Where plan data could be provided in a suitable form, a 2.5D view similar to that introduced in Alvey's Hindsite program could also be displayed. Database connectivity enabled *gnet* to be closely integrated with excavation recording databases and several techniques were explored to enable linkage with other widely used programs. Together, these features enabled the program to function as a component in a larger toolkit.

One of the less desirable outcomes of this phase of development was that, although earlier versions had been built for UNIX, Macintosh and PC platforms, by 1995, *gnet* could only be run on Windows machines. The later versions were dependent on several Windows-specific C++ class libraries, on ODBC to provide connectivity with databases, and on OLE for linkage with other programs. Eventually, development ceased with the introduction of 32 bit versions of Windows and the

accompanying changes in ODBC and OLE. Nevertheless, some users report continuing use of the program on modern operating systems, whilst others have reported failure despite many ingenious attempts to make the program work. The author himself has not been able to run the program on any of his machines for more than five years. Since its demise, there has been a steady stream of requests from colleagues to resurrect the *gnet* project and to bring the program up to date. Unfortunately, the considerable effort required merely to reproduce the program in a modern form was always difficult to justify when working in a research environment.

For several years, the author's research focus has been in mobile and ubiquitous computing, and the development of collaborative tools for use in mobile and ad hoc networks. Archaeological applications of this work have centred on data collection and access to remote information resources during field survey campaigns (Ryan *et al.* 1999; van Leusen & Ryan *in press*) and in tourist guides that can adapt to the level of specialist knowledge of the user (Ryan *in press*). Others have investigated the use of mobile and handheld devices in excavation recording (for example, Powlesland & May *this volume*; Ancona *et al.* 1999) but, so far, there has been little work that seeks to integrate models of the stratigraphic sequence into such mobile tools.

During this work on mobile collaborative tools, the need arose for a program to act as a simple test bed for research purposes. This need has now provided the justification to develop a successor to *gnet*. The remainder of this paper outlines the initial design requirements for this new system, and describes the current state of implementation of a prototype<sup>1</sup>.

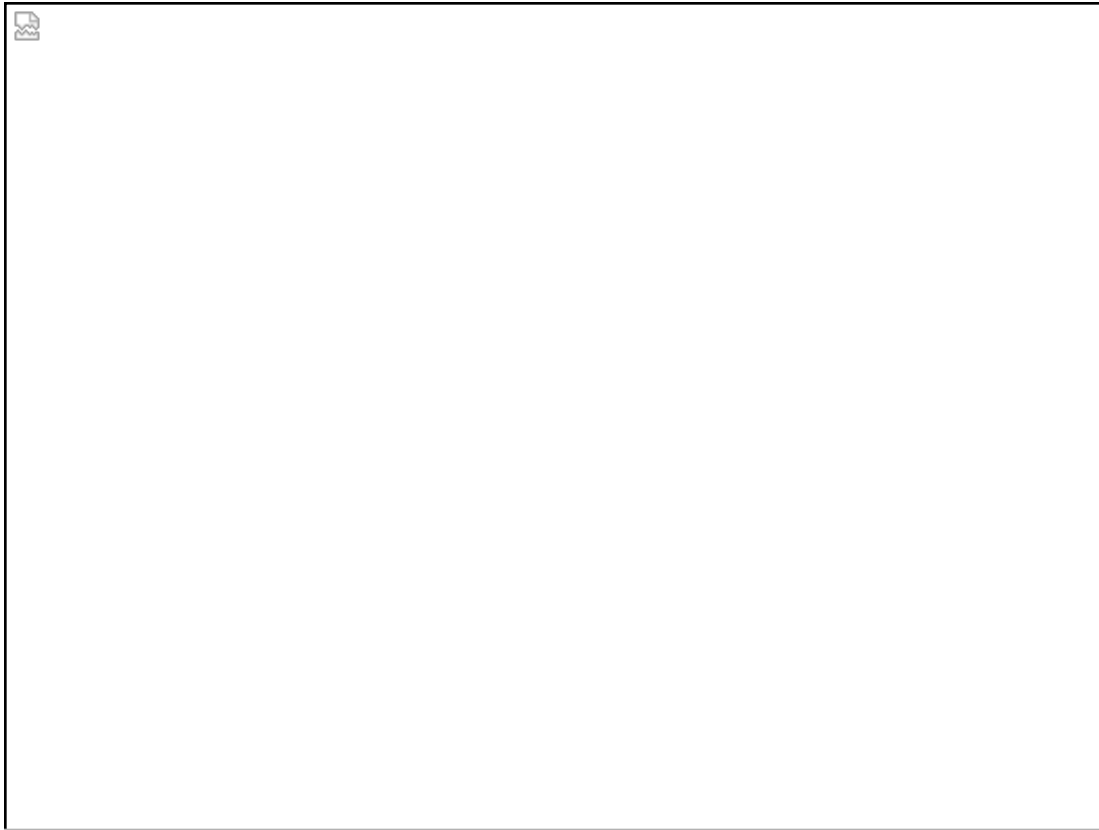
## **Design requirements**

The fundamental requirement for the new system was to provide equivalent graph visualization and editing functions to those of *gnet*. However, unlike its predecessor, it was also required to support a wide range of computing platforms and environments. It should work as a stand-alone application on desktop, laptop and handheld devices, and should support collaborative working in a networked environment. Like *gnet*, it should provide shared access to centralised databases for multiple users.

Mobility implies the need to operate in wireless network environments and hence to be able to survive disconnections from the network. Indeed, the program might be used for long periods with no network connection and so would need to ensure synchronisation during relatively brief periods of connectivity. Combining this with the collaboration and shared data requirement leads to a system that can work both on and off-line, and can resolve conflicts between updates made by different users.

*gnet* had offered only limited output capabilities. Diagrams could be printed, if necessary on multiple sheets of paper, but there was no way of exporting or displaying read-only diagrams either as images for publication or as models that might be used by a CAD system. Similarly, the ability to use the diagram as an exploratory interface for querying underlying data was not separable from its editing and other graph manipulation capabilities. Instead of trying to build all of these capabilities into a single monolithic program, it was decided to maximise the use of existing software and technologies. Diagram export, read-only views and a simple query interface could be provided through web browser interfaces.

Compatibility with existing mobile infrastructure developed for other applications, together with the requirements to work on multiple platforms led to the choice of Java as the implementation language for this new tool. This in turn inspired the choice of the name, *jnet*, reflecting both the wholly new implementation and the ancestry of its core functionality.



**Figure 1:** the basic architecture of *jnet*.

### ***jnet* Architecture**

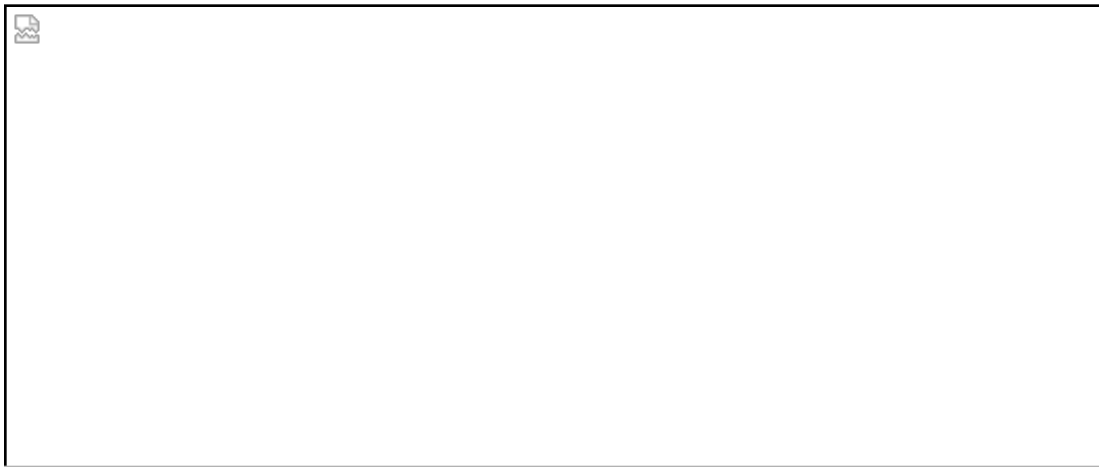
The implementation of *jnet* is based on the well-known Model–View–Controller (MVC) pattern (Figure 1). The GraphModel component holds a representation of the graph in memory and provides methods for layout, manipulation and editing. The Controller links the model with the view and routes messages between them. The view is represented by Painter and Canvas objects. The Canvas is a Java ‘interface’, a generalised specification of object behaviour which may be implemented in different specialised forms. Several specialised implementations of the Canvas interface may be plugged-in to render the graph in various formats. Depending on which version of the Canvas is used, the graph may be rendered as part of a fully interactive display, or as a stream of graphical commands in a language such as SVG<sup>2</sup>, VRML or X3D<sup>3</sup>.

A second interface provides similar flexibility in the storage of graph data. The GraphStore provides generalised behaviour for fetching and saving entire graphs or their component parts. Specialised implementations support local files, local and remote database connections (using JDBC), and other remote stores using an XML<sup>4</sup> serialisation of the data for transport over the intervening network.

Using these generalised interfaces and specialised implementations it is possible to compose different versions of the program for different platforms and operating environments. Figure 2 shows one of the simplest forms, a stand-alone interactive program running on a single desktop or laptop machine with data stored in local files or a local database. In this mode, the full range of graph editing and manipulation functions are available through an entirely conventional user interface.



**Figure 2:** *jnet* stand-alone configuration; an interactive program running on a single desktop or laptop machine with data stored in local files or a local database.



**Figure 3:** *jnet* client-server configuration provides multiple users with shared access to a central database on a remote server

Simply by changing the parameters used to establish the JDBC database connection, the same program can be used in a networked environment to provide multiple users with shared access to a central database on a remote server (Figure 3). An easier to maintain configuration can be achieved by running *jnet* as an applet in a Java-enabled web browser. This latter approach avoids the need to install the program on each client machine and provides a means to ensure that all users have access to the current version of the program. However, the security restrictions imposed on applets require that both the database and the web server used to deliver the applet must reside on the same machine.

More flexible network configurations may be achieved using Java servlets. Here, HTTP requests sent to a web server are redirected to an executable Java program, the servlet. Two such servlet configurations have been built using the same *jnet* components as described above. In both of these, a client program interacts with a web server which, in turn, communicates with a database server. Unlike the applet approach, however, there is no requirement that both servers reside on the same machine.



**Figure 4:** *jnet* servlet configuration delivering the graph as a stream of graphical commands for display in a web browser.



**Figure 5:** 2D representation of graph delivered to a web browser in XML/SVG format. Clicking on nodes gives access to underlying database records in a separate browser window.

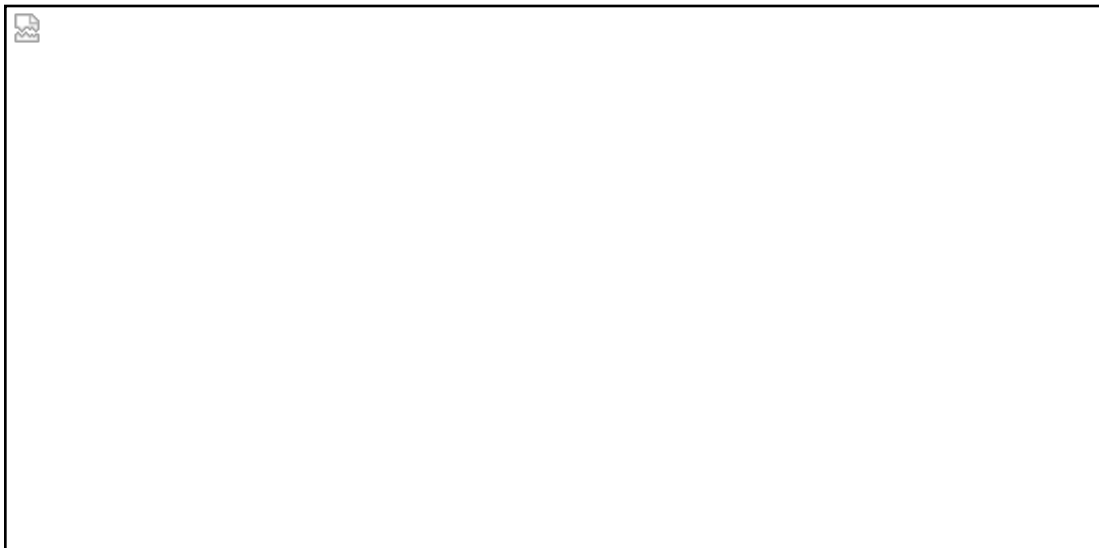
The second servlet configuration (Figure 6) offers a fully interactive editing environment suitable for use during the excavation and post-excavation processes. This version is intended to support collaborative working by members of an excavation team, possibly using handheld computers on a site served by a wireless network. Here, there is no requirement to render the graph in a graphical form as this role is performed by the client. The servlet uses the Controller and GraphStore components to deliver data to the client in an XML format, and to pass updates received from the client to the database server. The GraphModel component is only used when the requested data must undergo some initial processing before delivery, possibly to reduce the amount of data that needs to be passed over the network. For example, a client might request that the server removes all edges (stratigraphic



relationships) that do not form part of the stratigraphic sequence.

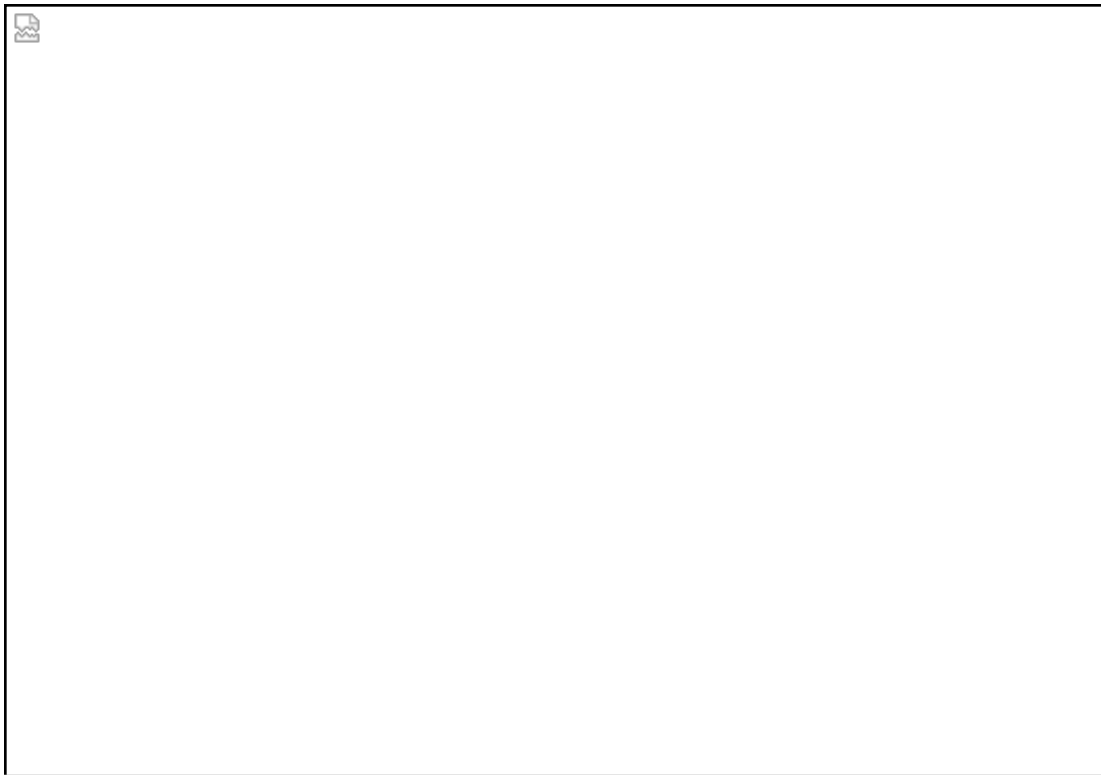


**Figure 6:** *jnet* servlet configuration delivering the graph as XML data for use by a *jnet* client.

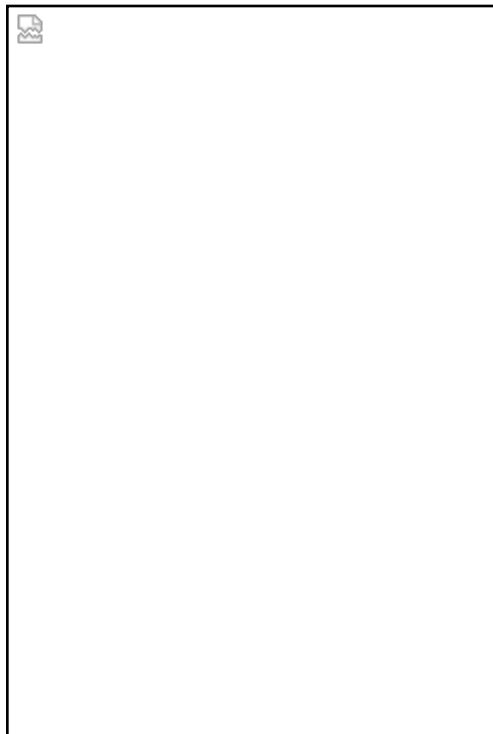


**Figure 7:** *jnet* client configuration.

The client device uses a complete interactive version of *jnet* configured to use a GraphStore component that exchanges XML data with the remote servlet (Figure 7). Applications working in a wireless environment must be capable of surviving frequent and extended periods of disconnection. Here, a local storage capability is added to cache requests and updates during periods of disconnection, and the GraphStore component manages this cache ensuring synchronisation with the server whenever a connection is available.



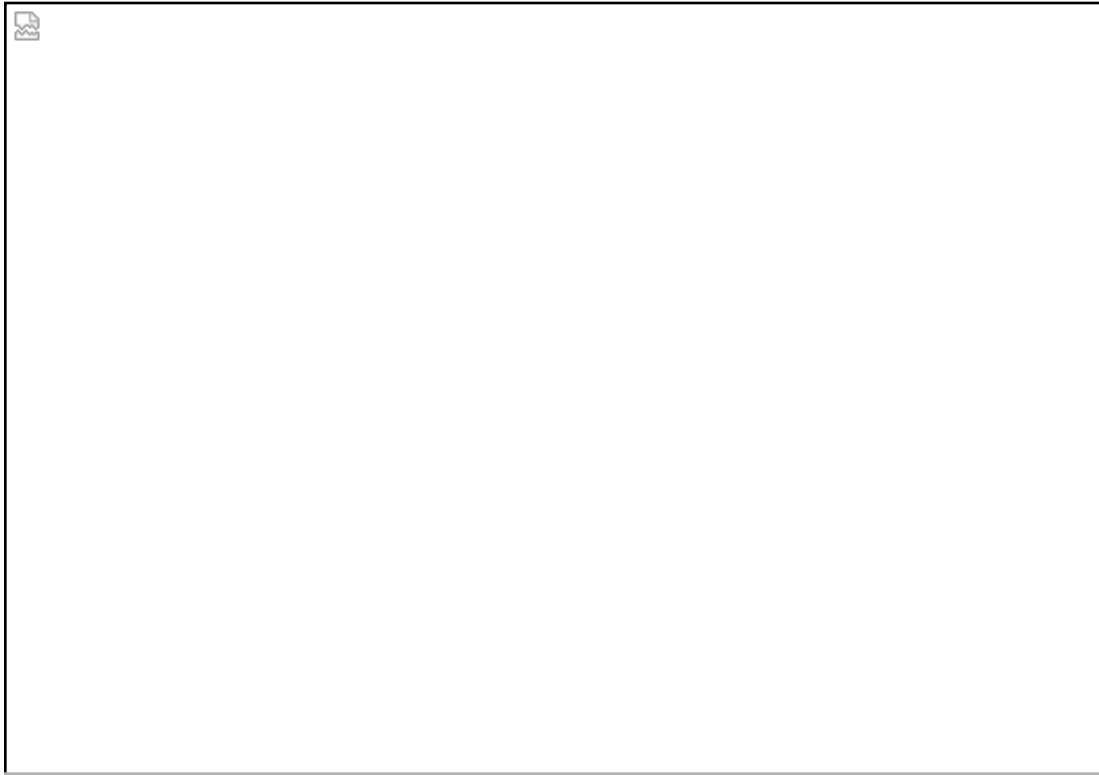
**Figure 8:** a list of available graphs in XML format.



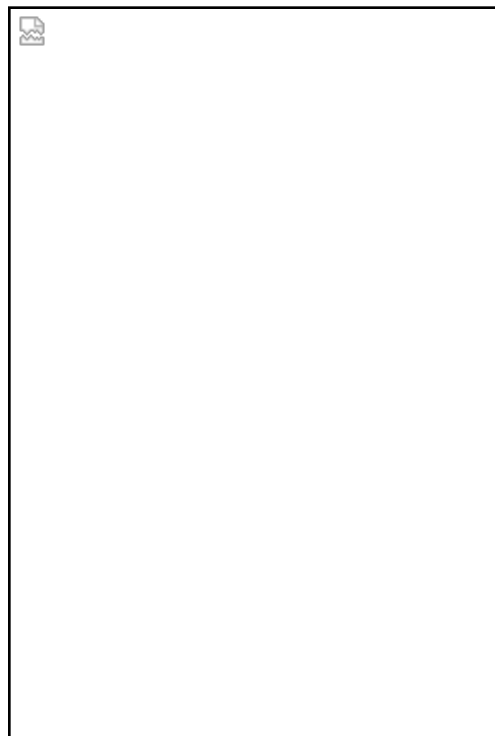
**Figure 9:** a list of available graphs displayed on a *jnet* handheld client.

Two examples of client requests are shown here to illustrate typical use of the system. Initially, the client may request a list of available graphs so that the user may chose one with which to work. If this request originated from a web browser, the user would probably see only the resulting XML data returned by the server (Figure 8). Using the *jnet* client, however, the data is used to build a pick-list

from which the user may choose the required graph. Figure 9 shows this step on a *jnet* client running on a handheld computer. Next, the graph data is requested and the server returns XML data describing the nodes and edges (Figure 10). The user may now manipulate and edit the graph using the client program (Figure 11).



**Figure 10:** graph data in XML format.



**Figure 11:** a graph displayed on a *jnet* handheld client.

The detailed working of the collaborative aspects of the system is still in development and is very experimental. The intention of the current model is as follows. Users will typically work with a subset of the graph, but may choose to display a larger subset or even the complete graph. For example, several users might be responsible for particular areas of an excavation but their displays could also show the contexts and sequence from adjacent areas, or the entire site. Access controls will be applied to define ownership of parts of the graph and so restrict which parts individual users may edit. These access controls will permit shared ownership so that parts of the graph may be edited collaboratively by two or more users. When nodes or edges are added or deleted, these changes are made available to all other users as soon as possible. Initially, this is achieved by adding update information to the server response following the next request from each client, but the benefits of immediate broadcasting will also be investigated. Each user may have a private layout (i.e. node positions) of the part of the graph under their control but, for collaborative use, several users may choose to share a common layout.

## Conclusions

Computer-based tools for visualising, editing and manipulating the stratigraphic sequence have a comparatively long history. Existing approaches have demonstrated their utility, as witnessed by the continuing widespread use of Herzog's programs, ongoing requests for an updated version of *gnet*, and the increasing popularity of the more recent *ArchEd* program (Mutzel *et al. this volume*). However, other than the highly integrated GIS-oriented approach demonstrated by Powlesland and May (*this volume*) and the reawakening of interest in 2.5D and 3D representations (Barceló *et al. this volume*), relatively little innovation has been apparent during the last decade.

The *jnet* system described here arose not from an archaeological requirement, but from a need for an application in which to test ideas about collaborative working in a mobile wireless networked environment. In its current form, *jnet* is not intended as a major step forward in the methods of producing, processing and manipulating stratigraphic sequences. Instead, it recasts the capabilities of the earlier *gnet* program in a form that is more appropriate to modern networked and distributed computing environments. Nevertheless, it demonstrates the potential for using handheld devices on site to support real time data collection and interrogation, collaborative working amongst members of excavation and post-excavation teams, and new ways of publishing and disseminating excavation data via the Internet. All of these rely on an apparently unique capability derived from its predecessor of using the sequence diagram as an exploratory interface to further excavation data such as the context records.

## Notes

1. The system described is an early prototype and should be considered as a work in progress. The current status of this project can be seen at <http://www.cs.ukc.ac.uk/people/staff/nsr/arch/jnet/>
2. SVG, *Scalable Vector Graphics (SVG) 1.0 Specification*, W3C Proposed Recommendation, 19 July, 2001, <http://www.w3.org/TR/SVG>
3. X3D, *X3D™ - Extensible 3D: New-Generation Open Web3D Standard*, <http://www.web3d.org/x3d/>
4. XML, *eXtensible Markup Language (XML)*, <http://www.w3.org/XML/>

## References:

- B. Alvey, 1989, 'Hindsite', *Archaeological Computing Newsletter*, 19, pp4–5.
- M. Ancona, G. Dodero, M. Mongiardino and A. Traverso, 1999, 'Taking Digital Notes in the Field', in J.A. Barceló, I. Briz and A. Vila (eds.), *New Techniques for Old Times, Computer Applications and Quantitative Methods in Archaeology*, 1998, pp117–121, BAR International Series, 757, Archaeopress, Oxford, UK.
- J. A. Barcdló, O. Vicente and O. de Castro, this volume, 'Towards a 3D Representation of Archaeological Layers.'
- R. Boast and D. Chapman, 1991, 'SQL and hypertext generation of stratigraphic adjacency matrices', in K. Lockyear and S.P.Q. Rahtz (eds.) *Computer Applications and Quantitative Methods in Archaeology*, 1990, pp43–51, British Archaeological Reports, Oxford, UK.
- B. Desachy and F. Djindjian, 1991, 'Matrix processing of stratigraphic graphs: a new method', in K. Lockyear and S.P.Q. Rahtz (eds.) *Computer Applications and Quantitative Methods in Archaeology*, 1990, pp29–37, British Archaeological Reports, Oxford, UK.
- J.G.B. Haigh 1985, 'The Harris Matrix as a Partially Ordered Set', in E. Webb (ed.) *Computer Applications in Archaeology*, 1985, pp 81–90, University of London Institute of Archaeology, London, UK.
- E.C. Harris 1975, 'Stratigraphic Analysis and the Computer', *Computer Applications in Archaeology*, 1975, pp33–40.
- I. Herzog and I. Scollar 1991, 'A New Graph Theoretic Oriented Program for Harris Matrix Analysis', in K. Lockyear and S.P.Q. Rahtz (eds.) *Computer Applications and Quantitative Methods in Archaeology*, 1990, pp53–59, British Archaeological Reports, Oxford, UK.
- J.W. Huggett and M.A. Cooper, 1991, 'The computer representation of space in urban archaeology', in K. Lockyear and S.P.Q. Rahtz (eds.) *Computer Applications and Quantitative Methods in Archaeology*, 1990, pp39–42, British Archaeological Reports, Oxford, UK.
- M. van Leusen and N.S. Ryan, in press, 'Educating the Fieldwork Assistant', in G. Bürenhult (ed.) CAA2001: Computer Applications and Quantitative Methods in Archaeology, Visby, Gotland, April 25–29, 2001.
- P. Mutzel, B. Reitgruber, and B. Schuhmacher, this volume, 'ArchEd: an Interactive Tool for Visualizing Harris Matrices.'
- D. Powlesland and K. May, this volume, 'Integrating the Matrix: Synchronised Management of Excavation Data at West Heslerton—Matrix, Plan, Database and Archive.'
- M.J. Rains, 1985, 'Home Computers in Archaeology', in E. Webb (ed.) *Computer Applications in Archaeology*, 1985, pp 15–26, University of London Institute of Archaeology, London, UK.
- N.S. Ryan 1985a, 'Some Thoughts on an Archaeologist's Toolkit', in E. Webb (ed.) *Computer Applications in Archaeology*, 1985, pp 126–132, University of London Institute of Archaeology, London, UK.
- N.S. Ryan 1985b, 'Interactive Tools in the Social Sciences', in P. Johnson and S. Cook (eds.) *People and Computers: Designing the Interface, Proc. British Computer Society Human Computer Interaction Specialist Group Conference*, University of East Anglia, 17–20 September, 1985, pp 404–414, Cambridge University Press, Cambridge, UK.
- N.S. Ryan 1988, 'Browsing through the Stratigraphic Record', in S.P.Q. Rahtz (ed.) *Computer and Quantitative Methods in Archaeology*, 1988, pp327–332, British Archaeological Reports, Oxford, UK.
- N.S. Ryan 1995, 'The Excavation Archive as Hyperdocument?', in J. Huggett and N.S. Ryan (eds.) *Computer Applications and Quantitative Methods in Archaeology*, 1994, pp 211–220, British Archaeological Reports, Oxford, UK.
- N.S. Ryan, J. Pascoe and D.R. Morse, 1999, 'FieldNote: extending a GIS into the field', in J.A. Barceló, I. Briz and A. Vila (eds.), *New Techniques for Old Times, Computer Applications and Quantitative Methods in Archaeology*, 1998, pp127–131, BAR International Series, 757, Archaeopress, Oxford, UK.
- N.S. Ryan (in press), 'Back to Reality: Augmented Reality from Field Survey to Tourist Guide', in F. Niccolucci (ed.), *Virtual Archaeology between Scientific Research and Territorial Marketing*, proceedings of the VAST EuroConference, Arezzo, Italy, November 24–25, 2000.
- J.D. Wilcock 1975, 'Archaeological Context Sorting by Computer', *Computer Applications in Archaeology*, 1975, pp 93–97.