

ArchE: A Graph Drawing System for Archaeology

Christoph Hundack *
Petra Mutzel **
Igor Pouchkarev
Stefan Thome

Max-Planck-Institut für Informatik, Saarbrücken (Germany)***

Abstract. We present *ArchE* (Archaeological Editor), a system for processing and displaying archaeological data. *ArchE* checks these data for consistency, simplifies and displays them; for each of these steps *ArchE* offers a number of different algorithms. The interactive features (eg input, data editing and modification of the layout) are easy to handle. Furthermore, *ArchE* contains algorithms for focusing on user-defined aspects of the data.

Apart from archaeological applications, *ArchE* can be used as a general graph drawing system.

1 Introduction

In archaeology the dating of objects is not only obtained by absolute dating evidence (as radiocarbon dating) but also by relative dating evidence. This relative dating within archaeological contexts (eg, excavations) is known as stratigraphy. Practical experience has shown that in almost all cases a spatial relationship (“is lying deeper than”) directly translates into a chronological relationship (“is earlier than”).

Stratigraphy was put on a systematic basis in the early 70s by E. Harris when he was confronted with an extremely complex and dense excavation in Winchester (compare [4]). Harris developed the so-called “Harris–Matrix” (or Harris-Diagram) to describe chronological relationships between the archaeological objects.

Since then a number of programs have been developed to facilitate the display and the analysis of the enormous amount of data. One of these programs is part of WinBASP (Bonn Archaeological Software Package). WinBASP was designed

* Research supported by a Graduiertenkolleg graduate fellowship of the Deutsche Forschungsgemeinschaft

** partially supported by DFG-Grant Ju204/7-1, Forschungsschwerpunkt “Effiziente Algorithmen für diskrete Probleme und ihre Anwendungen”

*** {hundack,mutzel,pouchkar,thome}@mpi-sb.mpg.de

in largely by the Rheinisches Amt für Bodendenkmalpflege, Bonn (Germany) and is a system for the support of various archaeological tasks (eg, mapping, seriation, and correspondence analysis of archaeological data). Unfortunately the present program of WinBASP for generating a Harris Matrix (see [5]) still runs under DOS, is only capable of displaying planar structures (or a slight extension of them) and has only limited layout features. *ArchE* will replace this program in the near future.

A *Harris Matrix* is defined as a set with the following relationships between its elements:

- (i) *a* is *later* than *b*;
- (ii) *a* is *earlier* than *b*;
- (iii) *a* and *b* are *contemporary*;
- (iv) *a* and *b* are *equal*;
- (v) *a* and *b* have *no direct relationship*.

Considering only relationships (i),(ii), and (v) a Harris Matrix is a partially ordered set. The relationship (iii) is used to combine objects found in the same layer, whereas (iv) states that parts of one object are found in different locations. From the properties (i) to (v), consistency checks can be derived, eg, testing for directed cycles.

In addition to these checks, the task is to compute and display a simple diagram of a Harris Matrix revealing the chronological order of its elements. In the course of this abstract we adopt standard graph terminology, referring to a Harris Matrix as a directed graph (for a survey on properties and layout of ordered sets compare [9]).

2 The *ArchE* program

At present *ArchE* achieves three goals:

1. it allows easy collection of data which are automatically checked for consistency;
2. it simplifies the input data;
3. it produces a layout which can be further edited.

All these goals have to be achieved sufficiently fast as the input which is given online usually consists of large graphs with up to several thousand vertices. Figure 1 shows the flow diagram of *ArchE*.

As most of the data is collected during the course of an excavation, input errors are very frequent. By searching for (and possibly displaying) directed cycles *ArchE* performs a consistency check of the input graph. Due to properties of the contemporary relationship we had to modify standard techniques and have implemented algorithms for testing the whole input graph and online insertion of edges.

In general the given graphs are not planar. In order to facilitate reading we minimize the number of edges to display all existing relationships between

vertices; this is done by computing the transitive reduction of the input graph. We have implemented different algorithms using depth-first search, the Floyd-Warshall algorithm, and boolean matrix multiplication.

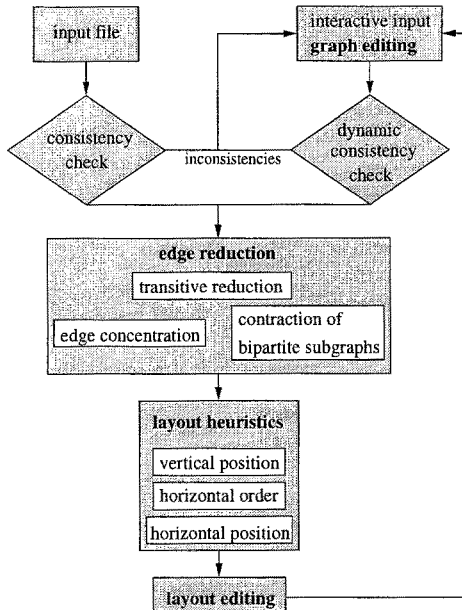


Fig. 1. Flow diagram of *ArchE*

The first one is intended for sparse graphs while the second is based on the well-known Floyd-Warshall algorithm for all-pairs shortest paths on arbitrary graphs. For dense graphs the third algorithm is the fastest; it relies on the fact that boolean matrix multiplication, transitive closure, and transitive reduction are equivalent problems with the same asymptotic running time [1].

Further reduction of edges is obtained by either *edge concentration* or search for complete bipartite subgraphs. We choose two arbitrary levels A and B and consider the induced bipartite subgraph $G, V(G) = A \cup B$. In both cases we search for a covering of G by complete bipartite subgraphs $(A_1, B_1), \dots, (A_k, B_k)$, such that each edge belongs to at least one (A_i, B_i) . For a complete bipartite subgraph (A_i, B_i) we insert an additional dummy vertex v' between A_i and B_i , and replace the edges between vertices of A_i and B_i by edges between A_i and v' , B_i and v' , respectively. We thereby reduce the number of edges from $|A_i||B_i|$ to $|A_i| + |B_i|$ and increase the readability of the layout.

In the first case we try to minimize $\sum_{i \leq k} |A_i| + |B_i|$, ie, the number of replacement edges. In the second case we try to minimize k , ie, the number of covering bipartite subgraphs. As these problems are NP-hard, NP-complete respectively, we have implemented heuristics (for details on edge concentration compare [8]).

Additionally *ArchE* includes an algorithm for finding small complete bipartite subgraphs (see [3]) as the existing complete bipartite subgraphs often possess only few vertices.

The main layout procedure uses extensions of the Sugiyama algorithm [11], which consists of three phases. First using topological sort, we determine the vertical order of the vertices (contemporary vertices are placed on the same level). Then their horizontal order is computed by a heuristic (eg, barycenter, median). Finally the coordinates of the vertices are determined following [11].

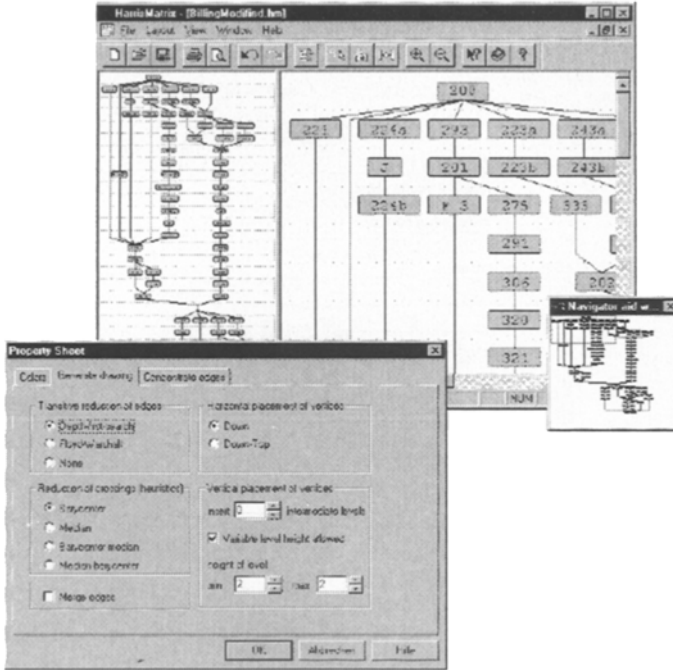


Fig. 2. A screenshot of *ArchE*

Graph editing features include the contraction of subgraphs; they are displayed as single vertices. Further *ArchE* has most of the advanced windows program properties including easy interactive editing, customizing of all user interface objects, zooming function (for different windows), undo/redo functions, printing and print preview functions, export/import facilities (for the archaeological format as well as for several graph editor formats), and a detailed help function (Figure 2).

We have chosen Windows95 and WindowsNT as platforms because these are the most frequently used among archaeologists. The need for an event based environment, the extendable design, and the required speed were reasons for designing *ArchE* in Visual C++ (4.2) using MFC application frame.

3 Practical experience and comparison with other programs

ArchE is one of the few graph drawing systems running under Windows and thereby usable for archaeologists. It has been successfully tested by several ar-

archaeological groups in Europe and is currently adapted to the WinBASP package. Due to the fact that *ArchE* computes edge concentrations and reduces bipartite subgraphs it decreases the number of edges of the graph and therefore receives drawings with a considerably lower number of crossings.

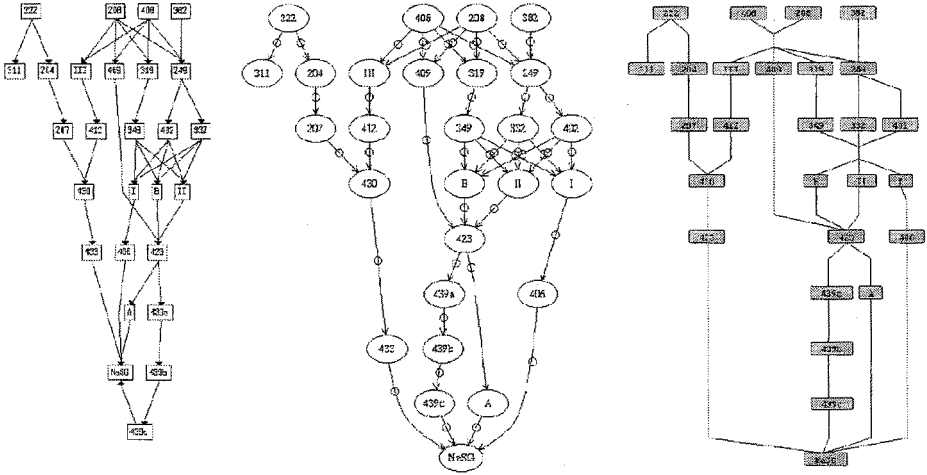


Fig. 3. Layouts of the data set *billing* by *VCG*, *dotty*, and *ArchE*

An example comparing the layouts of *VCG* [10], *dotty* [6], [7], and *ArchE* is shown in Figure 3.

We have tested *ArchE* on real data with up to 1056 vertices, 2624 edges and on even larger randomly generated graphs (up to 3000 vertices, 7000 edges).

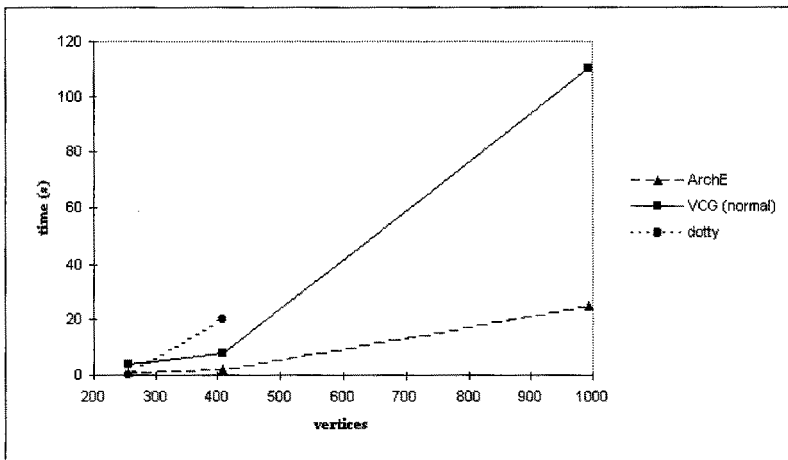


Fig. 4. run time comparisons with *VCG*, *dotty*, and *ArchE*

Comparisons with other graph drawing tools are listed below. They show that *ArchE* is one of the fastest existing systems (Figure 4). We received the

data courtesy of I. Herzog (Rheinisches Amt für Bodendenkmalpflege, Bonn), S. Lütgert, and A. Fuller (Museum of London Archaeology Service). The tests were performed on a Pentium 133 with 40 MB RAM under Windows 95. For *VCG* we used layout speed “normal”; for layout speed “fast and ugly” *VCG* and *ArchE* achieved similar running times.

Currently we are extending *ArchE* beyond the archaeological context to a user interface for the AGD–Graph–Library [2].

References

1. A. V. Aho, M. R. Garey, and J. Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972.
2. D. Alberts, C. Gutwenger, P. Mutzel, and S. Näher. AGD–library: A library of algorithms for graph drawing. Technical report, Max-Planck-Institut für Informatik, Saarbrücken, 1997.
3. M. de Berg, M. Overmars, and M. van Kreveld. Finding complete bipartite subgraphs in bipartite graphs. *ALCOM: Algorithms Review, Newsletter of the ESPRIT II Basic Research Actions Program Project no. 3075 (ALCOM)*, 1, 1990.
4. E. C. Harris. *Principles of Archaeological Stratigraphy*. Academic Press, London, San Diego, 2nd edition, 1989.
5. I. Herzog. Computer-aided harris matrix generation. In E. Harris, editor, *Practices of archaeological stratigraphy*, pages 201–217, London, 1993.
6. E. Koutsofios and S. C. North. Drawing graphs with dot. Technical report, AT&T Bell Laboratories, Murray Hill NJ, 1992.
7. E. Koutsofios and S. C. North. Applications of graph visualisation. Technical report, AT&T Bell Laboratories, Murray Hill NJ, 1994.
8. F. Newbery Paulisch. *The Design of an Extendible Graph Editor*. Number 704 in Lecture Notes in Computer Science. Springer-Verlag, Berlin, 1993.
9. I. Rival. Reading, drawing, and order. In I. Rosenberg and G. Sabidussi, editors, *Algebras and orders*, volume 389 of *NATO ASI Series [Series C: Mathematical and physical Sciences]*, pages 359–404, Dordrecht, 1993.
10. G. Sander. Graph layout through the VCG tool. Technical report, FB14 Informatik, Universität des Saarlandes, Saarbrücken, 1994.
11. K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, & Cybernetics*, 11(2):109–125, Feb. 1981.