Archaeological Computing

Harrison Eiteljorg, II

with GIS Chapter by

W. Fredrick Limp

Center for the Study of Architecture
Bryn Mawr, PA
Second Edition, 2008

Archaeological Computing, second edition, by Harrison Eiteljorg, II, with GIS Chapter by W. Fredrick Limp

Published by the Center for the Study of Architecture, P.O. Box 60, Bryn Mawr, PA 19010.

This work has been published as downloadable sections in PDF format. It is available for downloading at http://archcomp.csanet.org.

Preface to the Second Edition

Like most books, this one has been a along time in the making. It has taken some unusual routes from first glimmer to this second edition of the original product, and I think it worth commenting on some aspects of those routes.

First, the participation of Fred Limp as author of the GIS chapter has been crucial. I am not competent to say more than the most general things about GIS, but it is one of the three critical technologies that must be treated here. Knowing that Fred Limp would produce the GIS chapter gave me the impetus to keep plugging away on the rest.

Second, I wish also to acknowledge the invaluable work of Susan Jones, my assistant at CSA. Not only has she read and commented on everything, she has often made truly critical suggestions and comments. We argue amiably and effectively; the result is, I believe, an improved text. In addition, her work over the last several years on the *CSA Newsletter* has provided helpful information that I have put to use here.

Third, the book was originally conceived as a text, but the general absence of archaeological computing courses suggested that the idea of a text was not a good one. Thanks to the good advice of Mitch Allen (then of Alta Mira Press, now of Left Coast Press), the book was recast as more of a users' manual, a book that can be used readily alone as well as in a classroom setting. That, in turn, led to the choice of PDF files as the preferred form of publication. Being more of a manual will require regular updating, something far more easily accomplished with a digital format. But the book may be better used as a manual if printed out, hence PDF instead of HTML. Casting the book as a manual also led to the physical layout. The text column for all but this Preface has been kept relatively narrow – and it has been offset toward the left margin (toward the spine in the version for duplex printing) – to provide more space for margin notes.

Fourth, I realized shortly after the completion of the first edition that there was something important missing. There was almost no discussion of the problems associated with digitizing the records from on older project – either one that had been completed or a project of some duration entering a new phase. The initial work on such a discussion for a new chapter was undertaken fairly quickly, although its completion took longer than expected. That discussion, which makes up the seventh chapter of this new edition, is the major change from the first edition. Other changes are small and generally reflect either my shockingly inadequate proofreading skills as applied to the prior edition or my inability to leave well enough alone.

Several problems that I experienced with generating the PDF files for the first edition were problems only because of my relative inexperience with the process. I am indebted to Dr. Bruce Bevan, a graduate-school colleague at the University of Pennsylvania more years ago than I will admit, for writing to me to suggest what could be done to overcome those problems. This second edition benefits significantly from his helpful comments.

There are many figures in this work that use colors. Those colors are rarely if ever critical to a full understanding; an on-screen look at any figure that seems to involve a critical use of color should suffice. Note, in addition, that on-screen examinations of images will often permit significant – and useful – enlargement of the images as well.

In the long run, this product may be popular enough to be printed. If so, we hope the printed version will have benefitted enormously from critiques from users who download this from the web, use it, and let us know what they think about it. Please be a part of that group, return to the web site from which you obtained this document (archcomp.csanet.org), and let us know what needs to be improved. By the time of the preparation of this second edition the number of such comments has been vanishingly small. So I will again urge readers to assist future readers by commenting on the problems they have found.

Using PDF files as the publication medium, has meant that we can quickly and easily make changes to the "published" form of the book and constantly update the final product. Indeed, this second edition has been produced within about a year and a half of the completion of the first. Continuing changes may be expected.

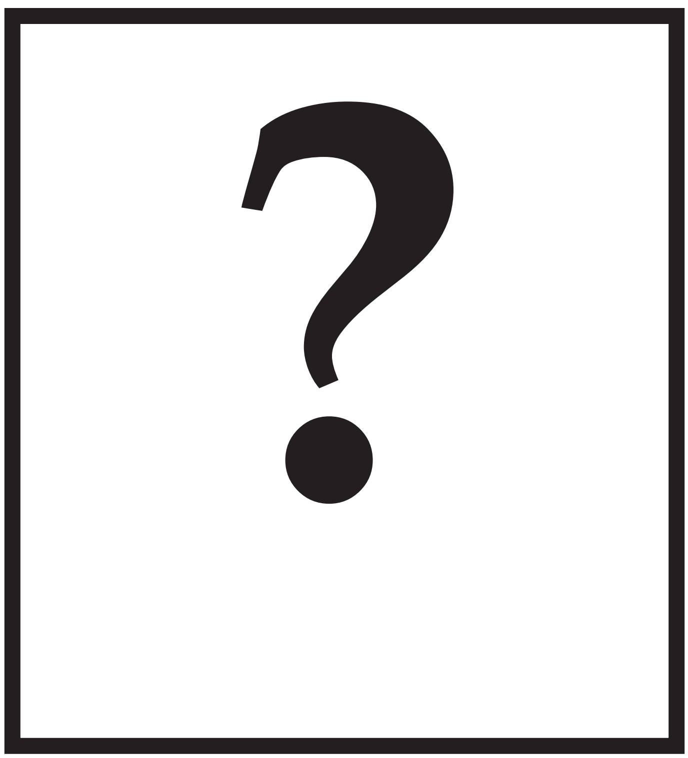
Harrison Eiteljorg, II
Bryn Mawr, PA
December, 2008

Archaeological Computing

Table of Contents

Introduction

Why



Why do we need this book – or a book like it?

That is a question I asked myself at the beginning of this project. The best answer is in the form of a story about a meeting I had with a colleague several years ago. My colleague was preparing a database for his excavation, and he was very excited, both about the database he had made and about the database management software he was using. The software was easy to use and encouraged him to believe that he could record the data from his project well and accurately without outside technical assistance.

When he explained his database to me, I learned that for each archaeological context he had prepared his system to record one, two, or three different ceramic wares found therein, but no more. I asked him about that, and he said that, if there were more than three wares in a given lot, that would be treated as a mixed lot, making the presence of different wares insignificant. I suggested that, taking contemporary imported and local wares together, one could easily have a half dozen or more contemporary wares. He then said, "That's the pottery expert's problem." That ended the conversation.

My colleague's problem was simple. He did not know how to organize his database to handle an unknown and possibly unlimited number of entries for pottery wares from a single archaeological context, nor did he understand the demands of good database design well enough to realize the problems he was creating for himself with his organization of the basic file. In particular, he did not understand how difficult it would be to search the file he had created, precisely because of the way he had structured it. That was something we never discussed, since the conversation closed with his comment that finding more than three contemporary styles of pottery in one context was the pottery expert's problem.

The situation has not changed very much in the years since. Too many archaeologists are still mis-using computer technology regularly – or failing to use the technology at all. In addition, many archaeologists who do use computer technologies are still trying to re-invent the proverbial wheel. Worse yet, archaeologists who create good and useful computer resources have a very small audience of colleagues knowledgeable enough to use those resources. Furthermore, few members of the profession are in a position to assess the quality of the digital resources that are available either for their own use or when writing a review – whether of project results or of a grant proposal.

Things are not looking better for the next generation of archaeologists. A survey done by André Tschan and Patrick Daly[1] showed that computer technologies hardly figure in American archaeology classes at either the graduate or undergraduate level. I had done a more casual survey of archaeology graduate school programs in 1999, and I had also found that there was very little attention paid to important computing issues.[2]

In that same year I had begun asking graduate students about their proficiency with computers on various campuses when I visited to lecture. Their responses, in fact, had been the impetus for my survey of graduate programs, because the students' responses to my questions were very worrisome. Most thought themselves to be reasonably adept with computers. On further questioning, however, that seemed to mean that they could use word processing software, email programs, and web browsers. Time and again, the graduate students I polled would indicate that they had little or no familiarity with database management systems (DBMS), computer-assisted design software (CAD), or geographic information systems (GIS) – while nevertheless thinking themselves to be computer-savvy. Since those

---

1 Tschan, A. and Daly, P., "Is there such a thing as 'Computer Archaeology'?" in Lock, G. and Brown, K. (ed.) *On the Theory and Practice of Archaeological Computing*, Oxford, 2000, pp. 135-138.

2 Eiteljorg, H., II, "Computing in the Archaeological Curriculum," *Archaeological Computing Newsletter*, 58: Winter, 2002, pp. 21-23.

are the three technologies that are crucial for recording archaeological data from real archaeological projects, that struck me as a very dangerous sign. Indeed, those three technologies are the primary subjects of this book

The absence of formal training and the resulting absence of skills belie the serious need for grounding in computer skills, a need that seems clear to many. As Tschan and Daly put it, "Information Technology in archaeology is subject to a rather urgent agenda and the development of 'computer archaeology' must not be placed on the shelf or put off until later."[3] Scholars must know how to use computer tools if they are to work effectively in the field, the lab, the office, and the classroom. They must also know enough about computer technology to be able to judge the computer work of others. The time to learn is not while in the field and on the job, where mistakes may be very damaging, but in the classroom as a student; hence, this book.

Another short example is relevant here. Visiting a survey project a few years ago, I looked at the database that had been developed for the project during the previous academic year. There was a serious problem. Although someone who was clearly competent had developed it, the system had no provision for displaying the data safely. That is, any time there was information displayed on the computer monitor, everything that could be seen could be changed, whether accidentally or intentionally. All changes were instantaneous and permanent. Nobody working on the project knew that all displayed information was "live" and subject to change. Nor did any of the staff know how to try to recover from an accidental change or erasure.

The last example for now. I visited a long-running excavation some years ago during the summer an excavation architect first began to use AutoCAD®. An architect who knew how to use CAD programs was hired to make the trench plans with AutoCAD. However, he had not become familiar enough with the use of one of AutoCAD's most important features to understand fully how to segregate the data as he drew the trench plans. As a result, all the plans were drawn together in one incredibly complex, impenetrable plan. He could not separately print out a plan of any portion of a trench or of any context. When I explained how to do what he sensed he should have been doing – but did not know how to do – he had to spend hours making the necessary corrections.

The time and place to learn about computers and computer software is in graduate or undergraduate classes where important principles of computing can be taught. Those of us who have been working with computers for a long time know all too well how easy it is to make major mistakes because of a lack of experience and training. We see such mistakes all the time – and we have made most of them ourselves. However, the stories above remain sadly typical. Such things should not happen on the job; they should happen in a classroom or computer lab where the consequences of errors are much less significant.

Can this book remain up-to-date?

In a word, no. Computer technology changes almost as fast as a falling snowflake. The snapshot we take today is wrong before we have finished winding the film for the next image – or, to be more in keeping with our subject, I should say before the image has been recorded on the digital camera's memory device. For that reason, we will provide added information at the web site from which you obtained this document: archcomp.csanet.org. We will use the site for new materials

---

[3] Tschan and Daly, p. 148. Despite using this quotation, I must take exception to the use of the term "computer archaeology." I fear that it suggests a special branch of the discipline, archaeologists who are expert in computing technologies. I believe strongly that ALL archaeologists must have considerable expertise in computer technologies. They will be using those technologies of necessity, and the computer must not be an impenetrable, magic "black box" to scholars who are dependent on them.

and links to relevant sites. In addition, we are prepared to operate a discussion list for those wishing to follow up on topics that are relevant. The web site may offer other features as time passes, depending on the needs of the readers.

What are the aims of this book?

Our primary subjects are the three critical computer technologies widely used to record archaeological field data and then to provide useful access to those data: database management systems (DBMS), geographic information systems (GIS), and computer-aided design software (CAD). For each of these software types, we will try to explain the design and data structure issues that should guide the development of a data recording system, pointing out the need for those procedures that are truly crucial, some that are important if not mandatory, and others that may be theoretically necessary but practically not. Often that will mean that we will be focusing on the eventual output of the system. That is, the most important function of computers for holding archaeological information is not simply holding information but giving that information back to scholars in useful forms.[4] When data are structured in incorrect ways, the retrieval can be hard to use, awkward, or simply wrong. We will often focus on this aspect of data recording – that is, retrieving what has been recorded – to explain why certain recording systems and/or processes are required.

We will also discuss various issues that arise with other forms of digitized data, including storage and access matters. All kinds of digital data, after all, comprise the total results of an archaeological project, whether the data are produced as the project progresses or after the fact.

An active reader should constantly be thinking about data with which he/she is familiar and how those data would be organized most effectively. The examples we present will rarely be as compelling as those from your own experience.

We will also cover some basic information about computer technology, including some moderately technical issues needed to inform the discussions.

We will spend a good deal of time and ink discussing how computer files should be documented and archived. After all, the best files are only useful if there is enough information about them to guide users in the future – and if the files are preserved for those users.

We will do our best to avoid jargon, but when we must lapse into computer-speak, we will try to define our terms carefully. If we have missed a definition and left you hanging about a term, please check the web site. If that does not help, let us know so we can add what is needed to the site. Instructions for communicating with us will be clear at the web site.

It may also be a good idea to say what is not an aim of the book. This is not a manual for a computer program. We will not be teaching you how to create a MySQL® database, an ARC/Info® map with associated data, or an AutoCAD model. When you have digested what is here, I hope you will be ready to read a software manual and begin to put DBMS, GIS, or CAD programs to work thoughtfully and effectively, but there will be a very long road ahead at that point.

This is also not the place for discussions of the Internet, the Web, digital photography (in terms of photographic processes – use and treatment of digital images will be discussed), illustration programs, email, advanced visualization programs, or . . . . Since this book is concerned with the recording of archaeological field data in digital form, the preservation of the recorded data, and providing access to the data, those other subjects are not ours. Similarly, we will not deal with statistical analysis packages here, though there may be an occasional reference to preparing data for use in such programs.

---

[4] As a college classmate once said to me, the good secretary is not the one who files things well but the one who finds quickly and accurately the things that have been filed. So it is with computer data.

One final note. This is not the place to begin if you have not studied archaeology yet. We will try to make our examples clear, but anyone using this book should have studied archaeology at least enough to know the basic terminology and excavation practices and to be comfortable with the core ideas of stratigraphy, artifact analysis, seriation, and archaeological inference. Our aim is to help you use computers in the practice of the discipline, you must bring to this enterprise some understanding of the discipline.

How is this organized?

In order to keep the text flowing in each chapter – or at least to try to do that – a great deal of important information has been separated from the continuing exposition into sidebar discussions. These discussion are often very important; their position outside the primary exposition, should not be taken to indicate a lower level of importance. The separation simply indicates that the subject in the sidebar needs a full discussion on its own, where it will not interrupt the primary exposition.

The organization of the web site may eventually be important to readers as well. It is certain that any text updates – along with descriptions of the update processes and dates of all material – will be available there. Other materials will be added as the needs and desires of users demand. The organization of the site will depend upon the needs of the users.

What are the consequences of making the book available in PDF form?

First, of course, you need not pay for the book, only for printing out the work, or individual portions of it, that you download if you choose to print. Second, you must abide by the agreement you made before downloading – the agreement that you would not print copies for others, whether free or not. The point of that is not to make life difficult for anyone but to make sure that everyone who downloads a copy knows that his/her copy was the current version on the date of the download. You also agreed not to distribute the PDF file – for the same reason and to prevent its being distributed or re-distributed from secondary centers. Since it is free via the archcomp.csanet.org page, there is no reason to make it available elsewhere. (By downloading the document(s), you have agreed to these conditions.) Furthermore, we believe that those who visit the web site personally to download their copies are more likely to participate in the review/critique/suggestion process that will help to make this document evolve and improve.

Second, we can update the document as required; of course, this is already the second edition. That is why you will find the date of the last update at the bottom of each page. The web site will always show the dates and histories of the various versions of the document so that you can be sure you have the most recent iteration. At the outset we made no estimate of the frequency with which the document would require changes, and the appearance of the second edition in about a year and a half may or may not be typical. Time will tell. We will try to make it very clear on the web site not only when changes have been made but how important the changes are.

Third, we can also change formatting and other matters that are separate from content. While that may not be likely, we will be listening carefully to comments about anything that makes it harder to use the documents effectively. (A downside: I must keep using the awkward term *document* since we are not talking about a traditional book or articles.)

A worrisome negative consequence of publishing these documents as PDF files is the potential for alteration. It is certainly possible for someone to break the protective shell of the PDF format and change one or more of the chapters. If that were done, our work would no longer be truly ours. That is why we asked you to agree not to alter the PDF files in any way. We hope to prevent anyone from ascribing to the authors either ideas or opinions that actually originate with someone else.

If the frequency of use of the PDF file seems to justify the investment, a bound version of this work may ultimately appear. While that would certainly change the nature of the web site, the site will continue to function in the event of a paper publication. Change will continue in the use of computers; the web site will attempt to mitigate the problems created by that change, whether this work is a paper book or a PDF file.

I

Some Basics

Introduction

To begin, we need to understand why computers are useful in archaeology, what they bring to the enterprise. Often the benefits of computers are taken to be so obvious that they need not be stated. Quite the contrary. We need to make sure the benefits are explicit so that we know how to evaluate the various applications and processes we discuss. Much of the discussion to come will be based very directly on the need to achieve explicitly defined benefits. In this beginning chapter we will only get a foretaste of those benefits, but in the coming chapters we will try to define more fully the capabilities of the systems we describe.

The simplest description of the benefits computers bring to archaeology is this: computers enable us to manage data more effectively and efficiently. That is true not only of archaeology but of business and most other scholarly disciplines in which computers are widely used.

It is more and more important for archaeologists to manage information effectively as the quantity of data collected on excavations and surveys grows. Carbon-14 dating, flotation, pollen analysis, micro-stratigraphy, dendrochronology, and other modern techniques have made the data load staggering, as have our modern habits of retaining and recording virtually everything, not just whole objects and large fragments. Without computers, it is nearly impossible to manage everything.

Manage, in turn, means several things. It means that information must be gathered and entered into the system accurately, speedily, and efficiently. This is not different from the requirements for a pen-and-pencil system of data recording. In either case the aim is to be sure that the information we have gained is accurately recorded and is recorded with as little waste of time and effort as possible. Whether using a pencil or a computer keyboard (or one of the other digital entry systems to be discussed later), good design eliminates duplication of effort to the greatest extent possible.

Entering Data

We use forms for entering data into paper systems, and we use forms in computer systems as well. People fill in forms with pen and ink or at the keyboard, and whether using pens or keyboards, those people will make mistakes. That is

Figure 1

A hand may work with a computer keyboard, a pen, a pencil, a paint brush, or . . . No matter the implement, though, the hand is connected to a human brain; so mistakes will happen. Guaranteed!

inevitable. Paper systems can include many kinds of subsidiary information to help reduce error, but computer systems can do much more. They can enter some things automatically (e.g., the current date), and they can be designed to include checks on data to prevent many kinds of common errors. Entry of a certain class of data can be restricted to a prescribed list of possibilities while the computer serves as silent watch guard to make certain that the pottery style entered is one of those permitted (and is spelled correctly). The computer might also check the arithmetic to be certain that the top of an archaeological context entered into a database is, as it should be, higher than the bottom. Indeed, a variety of such checks and cross-checks can be used by computers to limit the kinds of errors that can creep into data.

Well-designed computer systems also guard against entering the same information more than once – not only to make data entry more efficient but to prevent the existence of multiple, potentially conflicting versions of the same information in the data files. If a given piece of information such as the description of a catalogued object exists in only one place, it need only be entered once and can only be changed in a single place – even if it may ultimately appear on many different forms and reports. Similarly, a context or feature in a map or drawing may be used in many different printed maps and drawings without being drawn – or altered – more than once.

Storing Data

Once information has been entered, managing data includes storing the information in a systematic and accessible fashion. The most familiar paper-based recording system for archaeology is based on the use of file cards. Information about an individual artifact is stored on a card, and the cards for all the artifacts are stored in drawers or on shelves according to some simple system, commonly by catalog number – and often with cards for different kinds of artifacts in separate systems / drawers / cabinets. Computer data are different in terms of the technology, but the end results are similar. The storage is usually instantaneous on the computer, the automatic last step of data entry. Alphabetizing or arranging by numeric order is not only automatic, but the system can have many indices for the same data so that the user can access the information with any of those indices, singly or in groups – by catalog number, by excavation unit, by date of

Some of the benefits of a well-designed computer system used to record information from an archaeological project:

Data entry:

1. Automated checking of data entries can prevent errors, as can using preselected lists of allowed entries to prevent misspellings, unacceptable entries, etc.
2. Virtually any data-entry error that can be imagined should be preventable.
3. Geometric checks can monitor size and placement of graphic elements.
4. Many graphic items in CAD or GIS can be entered automatically from survey coordinates.

Data editing:

1. Editing of any specific data item can involve but a single change to it in the only place it exists in the system.
2. Changes to the data can be tracked so that all changes can be identified and the original entries preserved for reference.

Data Storage:

1. Storage of all data in a central repository can keep files safe from loss or damage.
2. Additional copies – kept in another place where harm that befalls one set cannot affect the other – can guarantee the permanence of the data.

Data Retrieval and Presentation:

1. Data can be presented to different users in forms desired by those users.
2. Each user can specify the form he/she needs for any data selection, including various forms for publication.

Logic enforcer:

1. Using computers encourages a clear, logical, and unambiguous system of data organization.
2. Individual modifications that might violate the logic of the system can be made all but impossible.

Better, more varied access to the data:

1. Many statistics can be produced.
2. Maps based on sophisticated forms of analysis can be produced.
3. Three-dimensional representations of sites and structures can be created.

excavation, and so on. Access to the information, of course, requires a computer and the right program(s).

Maps and drawings are also stored in different ways on computers. They are kept in larger, more complex and complete forms than the usual paper maps and drawings. Rather than making many individual paper maps or drawings at a variety of scales, each one a part of the whole, everything can be kept in a single, complex computer file, one for which scale is not a factor. Once stored in large and complex computer files, the individual pieces of the maps and drawings can be separated or combined for output to create virtually any combination of the individual parts – and printed at any scale desired. Colors, line weights, and line types can be altered on command. A variety of effects can be produced to illustrate specific points through the use of color, line weights, points of view, and so on.

Regardless of the forms in which data have been stored, digital data have a huge advantage over paper data that is often missed. Making copies of all the data files – for preservation or for sharing – is trivial. Indeed, part of any computer system plan must be the nature of the repository and the off-site secondary repository to prevent accidental loss or destruction. But making the copies is truly a trivial process and costs next to nothing.

Retrieving and Presenting Data

Once stored, the information in a computer must be retrievable in much the same way the cards in a paper-based catalog can be retrieved. However, there is far more flexibility with the computer. The cards in a paper system are exactly the cards filled out for information storage. The computer data, on the other hand, can be presented in the same forms used for data entry, but information can also be presented in other ways, using forms that meet specific needs of specific project personnel or other scholars for an individual job. A wide variety of reports, drawings, tables, maps, and charts can be generated by a good computer system. Indeed, the variety of data retrieval possibilities is so extensive that the initial design specifications of any system should include the requirements for data retrieval and reporting so that they are built into the system from the outset. The design of a good system, for instance, should include the capacity to produce a printed catalog of all object types – in forms desired by the project directors. In fact, there should be two quite different ways of presenting any catalog, one for display on screen and one for printing onto paper.

As indicated above, computer systems are also more flexible in terms of the kinds of indices that can be used to find items. With a card catalog, one must be content to find items by the index. With a computer catalog, on the other hand, multiple indices can be supplied; searches and sequencing are automatic even if one wants to search or order data via a category that has not been indexed in advance. In addition, computers make it easy to search for records according to multiple criteria, rather than a single one, pottery of a specific style and from a specific selection of contexts for example.

There are other important differences between managing data on paper and managing data with a computer. A paper system requires constant re-typing or re-writing of the information. Photocopying will substitute for simple copying of the information on a given form, but not for putting that information into another form for a new purpose, e.g., a published catalog. Computer data, though, can be re-purposed with ease and without re-typing; the artifact registry, for instance, can be printed out as a catalog, made into a simple list for a shorter discussion, or used to generate a selection based on specific criteria. Similarly, maps and drawings can be reproduced at various scales, with different colors or keys, and with different selections of the available information – all at the command of the skilled computer user. In each case, moreover, there is virtually no chance to introduce unintentional transcription or copying error; so the result is not only more easily achieved, the information is far less likely to have been corrupted in the process.

Flight	Airline	Destination	Departs	Remarks	Gate
121	USAir	Philadelphia	3:20	on time	B1
124	United	Los Angeles	3:07	on time	B14
135	Delta	New Orleans	3:30	on time	B6
140	United	San Francisco	3:41	on time	B11
145	Delta	Miami	3:00	on time	B3
152	United	Denver	3:38	on time	B5
157	USAir	Charlotte	3:14	delayed	B8
162	American	Dallas/Ft. Worth	3:46	on time	B21
165	USAir	New York - LaGuardia	3:12	on time	B2
168	United	Seattle	3:40	see agent	B15
173	American	Montreal	3:27	on time	B9
176	United	Vancouver	3:50	7:00	B20
183	USAir	Pittsburgh	3:34	delayed	B2
210	American	Atlanta	3:22	on time	B4

Departs	Airline	Flight	Destination	Gate	Remarks
3:00	Delta	145	Miami	B3	on time
3:07	United	124	Los Angeles	B14	on time
3:12	USAir	165	New York - LaGuardia	B2	on time
3:14	USAir	157	Charlotte	B8	delayed
3:20	USAir	121	Philadelphia	B1	on time
3:27	American	173	Montreal	B9	on time
3:22	American	210	Atlanta	B4	on time
3:30	Delta	135	New Orleans	B6	on time
3:34	USAir	183	Pittsburgh	B2	delayed
3:38	United	152	Denver	B5	on time
3:40	United	168	Seattle	B15	see agent
3:41	United	140	San Francisco	B11	on time
3:46	American	162	Dallas/Ft. Worth	B21	on time
3:50	United	176	Vancouver	B20	7:00

Figure 2

Two versions of a very simple data set, a listing of airline flights such as might show on a monitor at O'Hare Airport in Chicago.

The upper version of the flights shows them in flight number order (the left-most column), which might be useful to airline personnel who need information about equipment or delays and will seek that information based upon flight numbers. Passengers, though, will generally arrive at the airport with different information in mind and different needs. The second list shows one arrangement that might be better for passengers, with the flights arranged by departure time, from earliest to latest. This organization is commonly seen in European airports. More examples follow on the next page.

Note that in these and the following examples the data are always the same. The data are ordered differently, and the column on the left always is the one put in order – so the user of the information can most easily find what is needed.

Destination	Departs	Airline & Flight No.	Gate	Remarks
Atlanta	3:22	210 - American	B4	on time
Charlotte	3:14	157 - USAir	B8	delayed
Dallas/Ft. Worth	3:46	162 - American	B21	on time
Denver	3:38	152 - United	B5	on time
Los Angeles	3:07	124 - United	B14	on time
Miami	3:00	145 - Delta	B3	on time
Montreal	3:27	173 - American	B9	on time
New Orleans	3:30	135 - Delta	B6	on time
New York - LaGuardia	3:12	165 - USAir	B2	on time
Philadelphia	3:20	121 - USAir	B1	on time
Pittsburgh	3:34	183 - USAir	B2	delayed
San Francisco	3:41	140 - United	B11	on time
Seattle	3:40	168 - United	B15	see agent
Vancouver	3:50	176 - United	B20	7:00

Airline/Flight	Destination	Departs	Gate	Remarks
American/210	Atlanta	3:22	B4	on time
American/162	Dallas/Ft. Worth	3:46	B21	on time
American/173	Montreal	3:27	B9	on time
Delta/145	Miami	3:00	B3	on time
Delta/135	New Orleans	3:30	B6	on time
United/152	Denver	3:38	B5	on time
United/124	Los Angeles	3:07	B14	on time
United/140	San Francisco	3:41	B11	on time
United/168	Seattle	3:40	B15	see agent
United/176	Vancouver	3:50	B20	7:00
USAir/157	Charlotte	3:14	B8	delayed
USAir/165	New York - LaGuardia	3:12	B2	on time
USAir/121	Philadelphia	3:20	B1	on time
USAir/183	Pittsburgh	3:34	B2	delayed

Figure 3

Two additional versions of the list of flights.

The upper version of the flight listing shows the flights in destination-city order (the left-most column), which might be the most useful for passengers. It is the order most often seen in U.S. airports The bottom arrangement is ordered by airline and then by destination city, with the flight number made less prominent.

Again, the data are the same in all the listings. Changing the appearance of the data is simply a matter of changing the way the data are ordered and presented.

Sharing Data

Another aspect of managing archaeological data is the need to make the data available to others who may want access to it. In the case of paper data, that is done with photocopying, a relatively simple and inexpensive possibility. The only alternative is re-writing all the data. In practice, though, even photocopying is sufficiently expensive and time-consuming that archaeologists wanting access to excavation records must visit the home institution of the project and examine the records in person unless they want only a very small, well-defined selection from the records. Even then, archaeologists must be able to define their needs in terms that permit the appropriate records to be located. Copying digital data is, on the other hand, quick, simple, and inexpensive. Virtually the entire record of an excavation can be copied with little expenditure of time or money – and then mailed in nothing larger than a padded envelope or sent over the Internet for free, and those records can be searched thoroughly by the recipients without specifying their needs in advance.

Managing data also includes storing and preserving the data for future users. Photocopying (on acid-free paper) is the method of choice for careful management of paper data, but it is, as noted above, expensive and time-consuming when

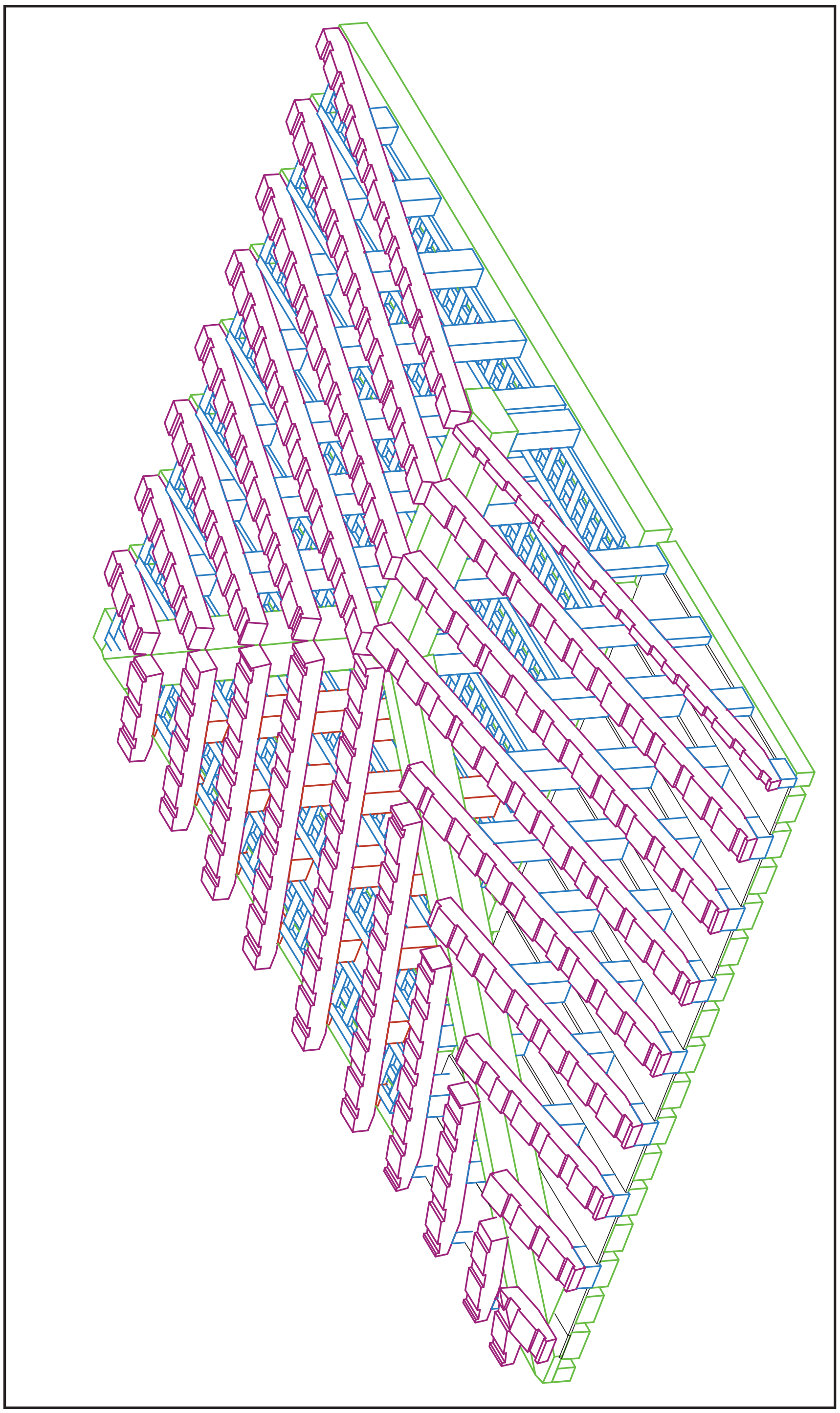


Figure 4

A reconstructed ancient roofing system – line drawing in color. Compare the two black-and-white drawings in figure 5.

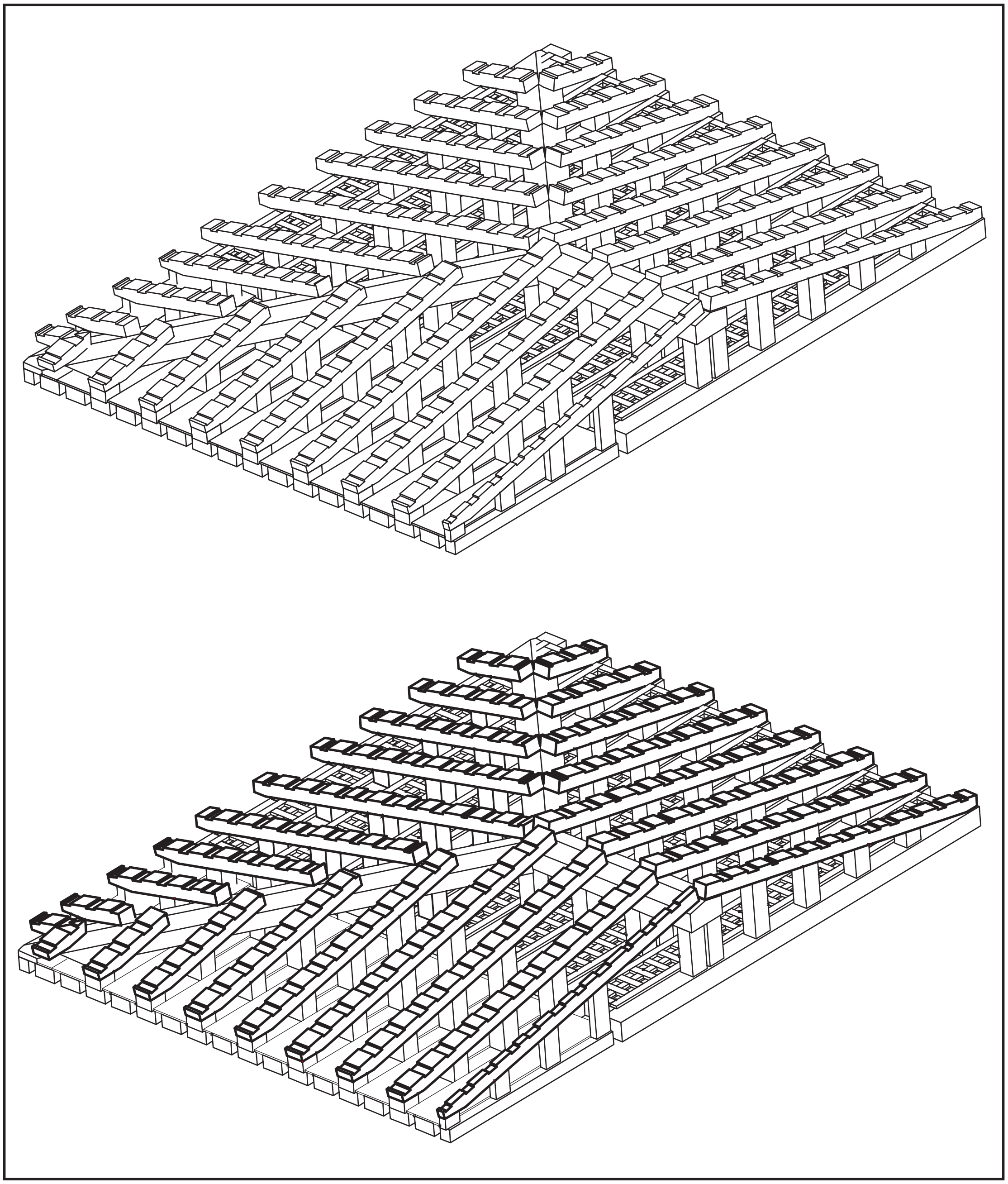


Figure 5

Two more drawings of the same roofing system. Both are black-and-white, but the lower one has differing line weights to aid understanding.

many pages must be copied. In addition, there is another step for the very long term: microfilming of archival materials, a system that has, when tested carefully, proved less reliable than frequently assumed. When the data are in computer form, copying the data is much simpler, but archival preservation for the long run is more complicated. Data must be actively archived; that is, archival care involves managing and altering the files on a periodic basis, not simply keeping them cool and dry; so it adds costs. Of course, microfilm preservation also requires specialists and adds costs. We will return to the archival preservation issue later in this book; it is an important subject that needs a full treatment.

To summarize, a good computer system for archaeological data has important advantages: data entry can be more effectively monitored and controlled, with data-recording errors reduced but certainly not eliminated. Storage and retrieval can be easier, safer, and more flexible, with information retrieved in many forms, even in the form of camera-ready copy for publication or electronic files ready for the printer. Re-purposing the information is easy and accurate, requiring no re-copying of information and eliminating the potential for adding errors when re-copying. Copying the information for use by other scholars is trivial, as is sending the information to others either on disk or over the Internet. Archival preservation is a less clear advantage for computer files.

The foregoing paragraph states some advantages of "a good computer system for excavation." It should be made quite explicit here that those advantages disappear if the system is poor. That, of course, is one of the reasons for this book.

The key requirements for developing any computer system are:
1. advance preparation with honest appraisals of the needs and demands imposed on budget and personnel by its creation,
2. committing the project – and the budget – to continuing maintenance of the system, and
3. documenting the system as it is planned and as it evolves.

Resources available at participating institutions should be utilized to the fullest, and planning should, from the outset, take those resources into account. Of course, a well-designed system requires another step: preservation of the data in useful forms for future scholars.

One of the most important aspects of this design and development process is making certain that the data are viewed holistically as the data for the entire

Building a Good Computer System

The process of creating a good computer system for an archaeological project is more than creating a good database, geographic information system, or CAD model – and connecting the parts effectively. First, the project director(s) must understand what they want and need. There should be no doubt about the reasons for using computers on the project. The rationale for using computers – and the specific data recording techniques – should be clearly and explicitly understood at the outset.

Second, the directors should be prepared to commit the financial and intellectual resources required. A good computer system will likely raise the early costs of a project and only later provide the most significant benefits, both intellectual and monetary. So there must be a realistic appraisal of those initial costs, as well as the on-going costs of system and data maintenance. Similarly, the intellectual rigor required to create an effective computer system is significant, and time is required to bring that rigor to bear. Project directors must be prepared to devote the necessary time and effort early in the planning period, before the project has begun, and they must be committed to spending time themselves, not simply setting tasks for others.

Third, everything done to prepare the system, from the first planning to the last minor change, must be documented. Descriptions of all the developmental processes must be retained, and project personnel must be committed to that work, even though it may seem unproductive. It is crucial to the long-term utility of the data.

Fourth, there must be a commitment to long-term care for the system. Project directors must not only ensure long-term care but remain involved in the general maintenance of the system so that the system and those responsible for it are seen as important to the continuing success of the project. Specific matters include such simple things as upgrading hardware and software over the life of a project and the much more difficult challenge of keeping appropriate personnel on staff for the duration – someone who can monitor the systems for necessary changes, repair problems, add capabilities, and so on. Long-term care also includes regular inspections of the data for anomalies that suggest data errors or terminological inconsistencies and sloppiness. Finally, long-term care includes an obligation to prepare the data for archival storage at the conclusion of the project as well as to arrange for and secure that archival storage.

project, not as isolated groups of information. That means not only including all specialists in the design process but making sure that the needs of the specialists, while honored, are never permitted to compromise the needs of the project as a whole. No conflict between the needs of the specialist and those of the project should be insoluble, but there are likely to be serious conflicts between various users. The project director must be involved in the development of the system if those potential problems are to be headed off successfully.

Treating the data as a complex whole also requires that the databases, GIS data sets, and CAD models be developed in such a way that, to the extent possible, there is no duplication and each piece of the whole is strengthened by each other piece. It is all-too-easy to develop the three technologically distinct data sets as if they were independent, but they should not be so conceived. To the contrary, they should be as fully integrated into a uniform whole as possible. One of the most important advantages of a well-constructed computer system is the recording of each piece of information in one place and one place only, thereby preventing any confusion that might arise from different versions of the same "fact." If the parts of a system are not well-constructed, that advantage can easily be lost.

Steps in Developing a Good Computer System
(Document them all!)

1. Define the system. What are the intentions for the digital data – only text data?, graphics as well?, what kinds of graphics?

2. Define the personnel. How many people will be available, for what portion of their project work, to develop and care for the computer system?

3. Fit the system definition to the personnel. The more complex the system, the more personnel will be required and the better trained they must be.

4. Involve local computer expertise – the campus computer center or information technology professionals in any sponsoring institutions. Networking intentions, temporary data storage, and long-term data archiving may all require the involvement of such colleagues.

5. Determine local support for hardware and software, whether from the computer center or elsewhere – and, of course, on-site issues that will impact hardware choices.

6. Involve the specialists from the beginning. All site specialists – pottery experts, lithics experts, etc. – should be consulted to ascertain their needs, and consultations with those specialists should continue until the system is working to everyone's satisfaction.

7. Examine the excavation/survey system and prepare data organization that adequately represents the data and contexts, in all their complexity. (An iterative process that will require refinement over time, this should not be rushed. The specialists should be intimately involved in this process.)

8. Examine the excavation/survey system for its impact on CAD or GIS applications.

9. Plan the relationships among databases, GIS data sets, and CAD models.

10. Choose hardware, software, and networking intentions, subject to change with technological advances, based on the support anticipated and the data organization determined in prior steps.

11. Specify data organization, create sample files, design sample forms (on paper and on screen) critique them, test with project personnel, repeat until all are satisfied.

12. Plan data back-up and off-season storage in concert with computer center and with networking plans (if any).

13. Add networking to the system, if that seems desirable, in careful steps and test for unexpected complications that may arise.

15. Test with as much sample data as possible, keeping those data for regular testing and refinement in the future.

Designing, developing, and testing a good system takes a long time. Less than a year will press the computer experts and may indicate that the project directors did not understand the importance of advance preparation.

The Real World

Again and again in the following pages we will speak of the ways problems should be approached, the kinds of software that should be used, the ways data should be stored, and so on. These comments will reflect the ideal, and it must be admitted here at the outset that the ideal is, as in ancient Greece, the unattainable perfection that provides a beacon, a guiding star. In the real world, we rarely reach the ideal, but when we take aim at perfection, two important things should follow. First, our results should be better, not perfect but better. Second, we should have made compromises consciously and knowingly. The latter is especially important. When time requirements, budget limits, personnel shortcomings, or any other problem prevents perfection, the issue is rarely that a given thing cannot be done. Rather, it cannot be done without spending too much or too long to do it. When the trade-offs are clear, explicit, and well-understood, everyone is better served. It is critical that you understand what you must sacrifice and why that is a necessary choice – and no sacrifice should put the data from a project at risk.

The Computer as Enforcer

There is another advantage to using computers to record data – a very different kind of advantage and one that often appears to be a disadvantage at the outset. If a computer specialist is involved in planning the data recording system and does his/her job well, the recording system will be better. That seems a bold statement. It is, but it reflects the added rigor brought to the task by the not-very-bright computer. Computers will, after all, do exactly as they are told, and they will make only those assumptions made explicit in advance by the people directing the process; understand only the explicit, not the implicit; catch no hints; tolerate no ambiguity. As much as that can be annoying, it enforces a level of discipline that can be very beneficial. It also encourages a design that is more complete and explicit, since nothing can be left to chance – or to the computer's nonexistent common sense. In the end, the computer understands archaeology and archaeological data only to the extent that the system designers have designed the data structures, rules, and procedures to mimic understanding. An example of this should be useful.

An excavator once explained to me the recording system for a site; it involved a very typical division of the site into units and sub-units, with the last, smallest unit in the hierarchy being the key archaeological unit, the volumetric shape that, for whatever reason, must be treated as a discrete, indivisible context. All the material from any such excavation unit is collected together and treated as contemporary in the sense that all were deposited at the same time. Individual items might be mapped if important and found in situ, but sherds and other small finds would have only the excavation unit identification to provide their locations.

There are occasions when a given excavation unit turns out to have been misunderstood during the process of excavation. Such an excavation unit may ultimately turn out to be two or three excavation units, erroneously thought to have been only one until excavation showed the full circumstances. The normal response to the discovery of such a unit is to stop, close the excavation unit being excavated (clean the boundaries and finish bagging, labeling, and record-keeping for that excavation unit). Assuming the multiple excavation units that should have been recognized have not been completely excavated, the remainder of each is then excavated as a new excavation unit, with all the appropriate records, bagging of artifacts, and record-keeping. The result is one more excavation unit than there should have been – the one including parts of each of the others – and smaller versions of the true excavation units than there should have been. Of course, the original, ill-defined excavation unit loses some value, since its contents may have been deposited at different times and/or under different circumstances. That happens with some regularity on any excavation because the distinctions necessary to define excavation units are rarely so clear that every excavator will recognize them all immediately.

For the excavator with whom I was consulting, most excavation units were recorded as just that, individual excavation units. However, an excavation unit later recognized to have been incorrectly identified and subsequently closed, with any remainder divided into more than one was also treated as an excavation unit, with the new, true excavation units that were then separately excavated called sub-units. There was apparently no other distinction between excavation units and sub-units. Excavation units were either true units or ill-defined ones, depending on whether or not they included sub-units. Sub-units presumably were all true units. (Though a sub-unit might itself have sub-units, at least in theory.)

In a paper system the excavation unit and sub-unit coexisted very peacefully. Anyone using the paper records could and would treat excavation units and sub-units as equivalent, and the analytic process could go forward without a problem. In that analytic process, those excavation units that had sub-units might have their value and importance reduced if the sub-units were found not to be contemporary.

When the excavator wanted to devise a computer system for the excavation, the existence of a separate entity called a sub-unit required a new category that would be truly distinct from the unit. If the sub-unit is different from the excavation unit, the computer must treat it as such – and probably must be prepared for a sub-sub-unit, which must be different yet. Therefore, a computer specialist designing this system would need to define the distinction between an excavation unit and a sub-unit. But there is no distinction. Each is a discrete, defined, undifferentiated volumetric unit from which artifacts have been collected; each is a context unit. Ultimately, the designer of the computer system would help the excavator to recognize the proper way to handle the problem – by treating all excavation units as equivalent units for data recording purposes. The fact that some excavation units should have been recognized as more than one is certainly of interest and must be recorded so that it will be taken into account in considering the stratigraphy and the history of the site. However, the question here is not one of two different categories of collection units, but one of qualitatively different excavation units – those that were misunderstood for a time and those that were properly defined from the outset. That distinction is not maintained by calling some excavation units sub-units since most excavations units are not tainted.

Computers on Small Projects

Even small projects involving only one person may employ computers to record, store, retrieve, and preserve data. The need for care and documentation does not change, and most of the steps included in “Building a Good Computer System” and “Steps in Developing a Good Computer System” are required; planning and making realistic assessments of the requirements for computing remain crucial. There are differences, though. If the archaeologists are not personally knowledgeable about the software tools needed and require advice about them, they should be certain to consult with people who will help to develop practical, down-to-earth solutions, not pie-in-the-sky grandiose ones. They must seek out computer experts who are accustomed to working with relatively simple tools. Overkill is surely a greater potential problem than the reverse when a small project requires computing technology.

Simpler processes can be used for data entry, and analytical routines need not be designed in advance. For instance, if only one or two people will be entering data, it is possible to dispense with many procedures designed to make data entry easier, quicker, or less error-prone. In such cases, though, the scholars involved must do regular checking of the data for errors; though that may be time-consuming, it is much less so than making error-prevention routines. Simply putting data in alphabetical order, for example, can easily show spelling errors or similar data entry problems. Similarly, analyzing data can be done as necessary with tools prepared for specific needs; it is not necessary to prepare tools for analytic procedures that may never be wanted.

In the final analysis, investigators working on small projects must be practical, but they must also

This example is a simple one that shows how, in a paper system, some ambiguity can easily be masked and become inconsequential. However, when the excavator tried – with considerable difficulty – to graft a computer system onto the paper one that had been in use for some time, a problem arose. Had a computer system been used from the beginning, the ambiguity of the underlying system would have been far easier to bring to the attention of the excavator. The system designer would have had to ask a good many questions about the nature of the sub-unit, and eventually it would have been obvious that unit and sub-unit were equivalent, even though the excavator wanted to treat them differently. Perhaps the result would have been an indicator in the records that some units were what the excavator would have called sub-units and others were “parent” units, but the use of a strictly rule-based system would have required the elimination of all ambiguity and the use of clear, explicit rules. In the final analysis, that is best for any recording system, of course; so having the computer there at your side demanding explicit answers to the simplest of questions can be remarkably beneficial.

The benefits thus far discussed show that computers add value to the excavation process, making data storage and retrieval easier, faster, more versatile, and less prone to error. In addition, their help in making the whole process more open and explicit is valuable in surprising ways. The whole story, however, is far more complex. Computers often bring new capabilities to the archaeologist by virtue of the speed and completeness of the data recording. For instance, good digital records will permit a variety of statistics to be generated automatically. Good field recording systems will provide feedback very quickly – potentially even in real time – to guide excavators. Complex relationships between and among objects, contexts, lab results, conservation, personnel, and comparanda can be established and used easily for any number of purposes. The site conservator, for instance, could have immediate access to all that has been put into the system about an object the moment it enters the lab, and his/her work, as documented in progress, should be available to others on site as the conservation goes forward.

Figure 6

An excavation trench.

How does the computer treat a poorly-excavated excavation unit? Just the way it treats a well-excavated one like this. But the recorded data should include information to make that distinction between a well-excavated unit and poorly-excavated one clear to any user.

Providing these benefits with a computer is not without cost. A good computer system must be efficient, stable, reliable, and robust. Making such a system is very time-consuming and requires substantial expertise. The end result, however, provides benefits to all users. Equally important, a good system requires no added time on the part of the individual user to tap into the strength of that system.

We have continued to discuss relatively simple gains in efficiency. In many cases, though, the most important capabilities of digital data simply have no analog in a paper-based system. In those cases the advantages are not increased speed or efficiency, they are totally new capabilities.

New Capabilities

Consider, for example, the case of a scholar working on French Iron Age hill forts (Dr. Scott Madry of Informatics International, Inc., and Research Associate Professor at the University of North Carolina at Chapel Hill). With complex maps and associated data he was able to show the relationships between and among the hill forts, but only by using a computer and a geographic information system (GIS) was he able to find the connecting roads. The GIS system made it possible to learn "that the old Celtic road network connecting the hillforts of the area tended to follow within the line-of-sight of the hillforts, rather than take more direct paths (as originally proposed in Madry and Crumley 1990)."[1]

Technically, a person should be able to do what the computer did in that case. After all, people created the equations that were used, and people designed all the pieces that eventually produced the result. Nonetheless, it is safe to consider this result to have been something simply beyond current human capabilities without a computer. The time required would have been too great to permit making the calculations again and again until a useful result was achieved.

Another example: I fully measured the remains of a building and, instead of making paper drawings, made a computer model with the aid of a computer-assisted design (CAD) system. The result was not simply a group of related drawings but a truly three-dimensional representation of the structure, and I could manipulate the model to see different parts in a variety of combinations and in 3D views. When I requested only the marble portions, it was clear that two vertical marble blocks – separated from the other marble blocks by a group of five horizontal soft limestone blocks – must have been adjacent to the other marble blocks in an earlier phase of the structure. When placed there in the model (impossible to do on site, of course), they not only fit correctly, but the 3D model let me see that the diminution of the lower block's thickness matched the shape of the cutting on the block originally adjacent to it. Computer copies of the marble blocks could then be added to the model in their original positions for the prior phase of the structure and displayed when the reconstruction of that phase was displayed.

One final example: a database of pottery. The database used to manage all the information for analysis was constructed so it could also serve as the source for camera-ready copy for the final publication. In doing so, the different parts of the data system – the catalog, the list of shapes, the list of design motifs, and so on – could be used separately but also could be related to one another to create a unified whole.

That same pottery database might well become part of a larger data aggregation – perhaps being included with other pottery from other sites (assuming common terms and analytic processes). The pottery might also be included in

---

1 Madry, Scott, "GIS and Remote Sensing for Archaeology: Burgundy, France," basic page, leading to www.informatics.org/france/gis.html, Line-Of-Sight Analysis, first paragraph, last accessed November 17, 2006. [The Madry and Crumley reference: Scott Madry and Carole Crumley, "An Application of Remote Sensing and GIS in a Regional Archaeological Survey" in *Interpreting Space: GIS and Archaeology*, K. Allen, S. Green, and E. Zubrow, eds. Taylor & Francis, London: 1990.]

a large-scale GIS data set so that scholars could use maps to access information about pottery found in specified locations. Indeed, that integration of the results of particular projects into a larger universe is one of the most exciting uses of computer technology, promising the potential to guide people far more quickly and efficiently to the resources of interest to them.

Other examples could be provided, but the point is that computers give us capacities that we simply do not have without them. They bring us new capabilities that, once understood, we do not want to be without.

Aggregated Data – The Computer as Encyclopedia

There is one more expectation about computing systems that we need to discuss here. Many people assume, not unreasonably, that computers will permit us to access all computer data in some easy-to-use and roughly universal form. Thus, one might expect that it will soon be possible to ask a computer for all occurrences of Mimbres pottery in excavation contexts or all Terra Sigillata pottery, or all Neolithic pollen samples. The possibilities are endless. Unfortunately, the possibilities are also remote.

There are two major problems with these hopes. First, of course, there is the simple fact that much of our information is not now in digital form and is not likely to be in digital form for a very long time. The costs of digitizing data from old excavations is truly staggering, and the likelihood of funding is small.

The second problem arises from the nature of the practice of archaeology. We do not use carefully controlled language that is uniform from time to time, place to place, and scholar to scholar. Sometimes the differences are interpretive: one scholar's *amulet* is another's *jewelry*. Sometimes the differences are terminological: one scholar's *trefoil-mouthed jug* is another's *oinochoe*. Differences may also be language-based: is the architect of the Propylaea *Mnesicles* of *Mnesikles*? That distinction arises from transliteration only; throw in real cross-language issues, and the problems grow exponentially. We are an international discipline, and we speak – and record our information in – many different tongues, confounding any attempt at true uniformity. The terminological inconsistency is a problem many have hoped computers could overcome for us; others have proposed ways for us to overcome them ourselves so that the computers can do simpler tasks. Nobody, however, has found a magic wand. I am among those who do not believe the magic wand exists.

We also do not excavate in the same way in all times and places. Some excavators count sherds; some weigh them; some try to determine the maximum number of pots that could account for the sherds found. That such a basic part of the fruits of excavations could be treated differently expresses just how variable the recording processes can be. Nor do we conduct surveys in the same way. As a result of these terminological and methodological issues, the data – assuming that they are in digital form already – are very difficult to aggregate, to treat as a single large unit rather than many related small units.

The point, I hope, is clear. As a profession we do not have sufficiently consistent standards either for vocabulary or for excavation systems to permit us to lump information together and make sense of the result because we cannot be sure that the individual pieces are truly consistent. It is very unlikely that this situation will change, since every archaeological project differs in significant ways from every other archaeological project. That is not to say that uniformity of vocabulary is impossible; there are good and useful guides to archaeological vocabulary, ranging from A. O. Shephard's 1957 work, *Ceramics for the Archaeologist,* to any number of specialist ceramics studies, from the Getty *Art and Architecture Thesaurus* to specialized architectural studies, and so on. Using such guides will help enormously, but, even should they become standards, there will be much old data expressed with less precise terminology, and even well-defined terminology is not truly static. To add to the problems, other languages will have sources unreconciled to those used in English Correct use of the terms, of course, cannot be guaranteed either. Thus, while moving to standard prescribed vocabularies is critical to the effective use of computers for aggregated data, it will remain necessary for scholars to look carefully at all aggregated data, even when standard vocabularies have been widely adopted.

In fact, our inability to lump all the digital material together and somehow treat it as a unit is one of the reasons this book is needed. If we all used the same terms, dug in the same way, and recorded our information in the same systems, it would be much simpler to deal with the results. We would all be using common tools, and we would not need as thorough a grounding in the basics of computer systems in order to create, access, or evaluate digital data files. As things stand, however, we are not using the same tools, just the same categories of tools. Nor are we recording the same data with the same terms. So we really need to know a good bit about the categories of computer tools if we are to use them well. For each project, in fact, we need to be able to understand the data recording systems just as we need to be able to understand the excavation or survey system.

Organization of the Text

The remainder of this book consists of eight additional chapters. In Chapter Two we will introduce some very basic computer terms and concepts, some issues that relate to choices of computer hardware and software, and some technical issues of importance to scholarly computing. The following three chapters are discussions of the specific computer technologies that are our primary subjects, one per chapter – database management and database management systems (DBMS), geographic information systems (GIS), and computer-assisted design software (CAD). These are the crucial technologies used for recording archaeological data in the course of excavation or survey.

Not all information from a modern project will be stored in databases, GIS data sets, or CAD models. The sixth chapter will be concerned with the data types that may be part of a modern archaeological project's total data set. We will discuss a variety of issues surrounding the use of digital images, audio, video, and text, but our concern will be with how to organize and manage the files, not how to take digital photographs, make recordings or videos, or enter text.

The seventh chapter, new to this second edition, is concerned with a group of problems unique to digitizing older projects – either projects that have been completed and whose information needs to be better recorded for access or projects still on-going and preparing to convert to computer-based data recording. There are significant differences between the work of a scholar preparing to record data digitally from a new project and that of a scholar trying to digitize extant, paper-based records.

In the eighth chapter we will deal with issues of data protection and preservation, especially those issues involving archiving and making certain that data are available to scholars for **effective** access in perpetuity. The discussion will cover issues surrounding short- and long-term storage of digital records, archiving, and access to those records. Documentation of the recording system will be discussed at length there and in the chapters on the base recording technologies. There are serious issues involved, ones few scholars have fully come to grips with at this time. In the final analysis, it is a truism that archaeologists destroy their evidence as they dig, leaving only the artifacts and the records, not the contexts. If the records are not kept, the loss is enormous. If the records are kept but not accessible, the loss is the same.

A conclusion follows to wrap up the discussion, returning to some important themes and adding some practical considerations that apply broadly to archaeological computing. In that final chapter will also be considerations of the ways the three primary data-gathering technologies should be used – singly and in combination – in archaeology.

CAD vs. GIS

CAD and GIS are often confused and can easily be taken to be more similar than they are. Both involve representations of the real world similar to the maps and plans with which archaeologists are so familiar. Both have mechanisms to connect map/plan data to attribute information about objects in those maps and plans. The differences between CAD and GIS have also been blurred by the fact that people have often used CAD to create maps that are, in turn, used by a GIS program.

Yet CAD and GIS are quite different. First, CAD systems rely on a Cartesian gird (x and y or x, y, and z) that assumes a Euclidian geometer's view of the world. GISs, on the other hand, approach the world as geographers have – by conceiving of drawings as if they were draped onto a model of the earth. The earth, of course, is far more complex than a simple sphere, not to mention the relatively simple Euclidian world.

Second, GISs have been designed to deal with the world, again as geographers do, as if all maps were simply lines on the surface of the globe. Those lines may have differing elevations, but they do not require a fully three-dimensional approach to the world and its geometric features. The underlying globe, with its elevations indicated, takes care of any necessary three-dimensional information. CAD systems have different needs and have been developed to represent fully three-dimensional objects.

Third, CAD programs have focused on the drawing as output because the architects and engineers who use CAD programs rely upon those drawings themselves. GIS programs have been intended to produce drawings as well, but they have also been designed to produce maps that are built in response to questions that, in the simplest expression, invoke set theory to determine what will be included in a given map. Because GIS maps are so often produced in answer to specific queries of the data in the system, they are more often used on-screen.

Fourth, CAD systems rely upon a line-art approach to drawing (more properly *vector graphics,* a term that will be defined in the next chapter) whereas GIS software can utilize either line-art style drawings or photographic-style imagery (*raster graphics*, also defined in the next chapter).

Finally, CAD programs provide a rather elementary view of their world, even if that world is fully three-dimensional. They do not have the built-in intelligence to understand that two objects are adjacent to one another, even if they share an edge. GIS program have been explicitly designed to understand many relationships between/among objects. Thus, a GIS system will understand that one object is within another, adjacent to another, crosses another, and so on.

These differences have combined to make two quite different tools. CAD excels at modeling real-world objects – structures or excavations – and can be very effective in making maps of relatively small areas (where the earth's shape is not consequential). GIS is very powerful at combining what is known about bounded areas on the earth (or points) with information about those areas and points. GIS data sets often yield maps combined with other kinds of information in new and meaningful ways.

The distinction between CAD and GIS seems strong today, but there are already programs moving to bridge the gap, to provide the kind of real-world modeling available in CAD programs with the robust data linkage of GIS. It may be only a matter of a few years before the two types of software merge, though there are market forces that may slow the progress. In any case, today's archaeologist needs to understand both technologies at least well enough to know when to use each – or both.

Where appropriate, there will be a glossary at the outset of the chapter so that the terms needed for the discussion can be clearly defined before they are used. Even if you believe that you do not need those glossaries, you should at least go through the list quickly to be sure that there are no terms being used in ways foreign to your experience. (The combined glossary is available as a separate PDF file for downloading so that all the terms may be found in one place.)

In each of the chapters dealing with the core computer technologies – database management, geographic information systems, and computer-aided design – there will be examples woven throughout the discussion to provide a real-world use of the particular technology. It will be important for you to try to understand the examples thoroughly. The better you understand the needs of archaeologists in particular circumstances, the better you will understand why the software is used as it is or, in some cases, why current software is inadequate. It will be to your advantage to consider how archaeological data you know well and understand thoroughly may be fitted to the systems described – to ask yourself how those data could be treated by the software in question. You will understand the systems best when you can see how to make them function with your data in order to meet your needs. Whereas our examples may be compelling, they cannot carry the force of your own real needs.

The book as a whole and each technology chapter will also have information intended for different categories of readers. Those who need to use a digital resource need to know something about the software used and the data structure. Of course, those who must create such resources need a much broader and deeper understanding of the software. Lying somewhere in the continuum between the resource user and the resource creator are two other categories of users – those who need digital records for their work but must hire specialists to create them and those who need to evaluate digital resources for their own use or the use of others. Each of those categories of computer users – and they are certainly not mutually exclusive – needs different kinds of information, and this book is intended to help each, which implies, of course, that most readers will find some parts of the discussion to be more valuable than others.

Gathering data to put into the computer can be difficult, especially with GIS and CAD systems. Therefore, the CAD and GIS chapters will include comments about obtaining data. Specific information about the ways data can be collected and put into the computer should be helpful. In some cases, this is not only an important part of the process but a truly critical one, providing the basic limits on precision and accuracy that will affect the utility of the data.

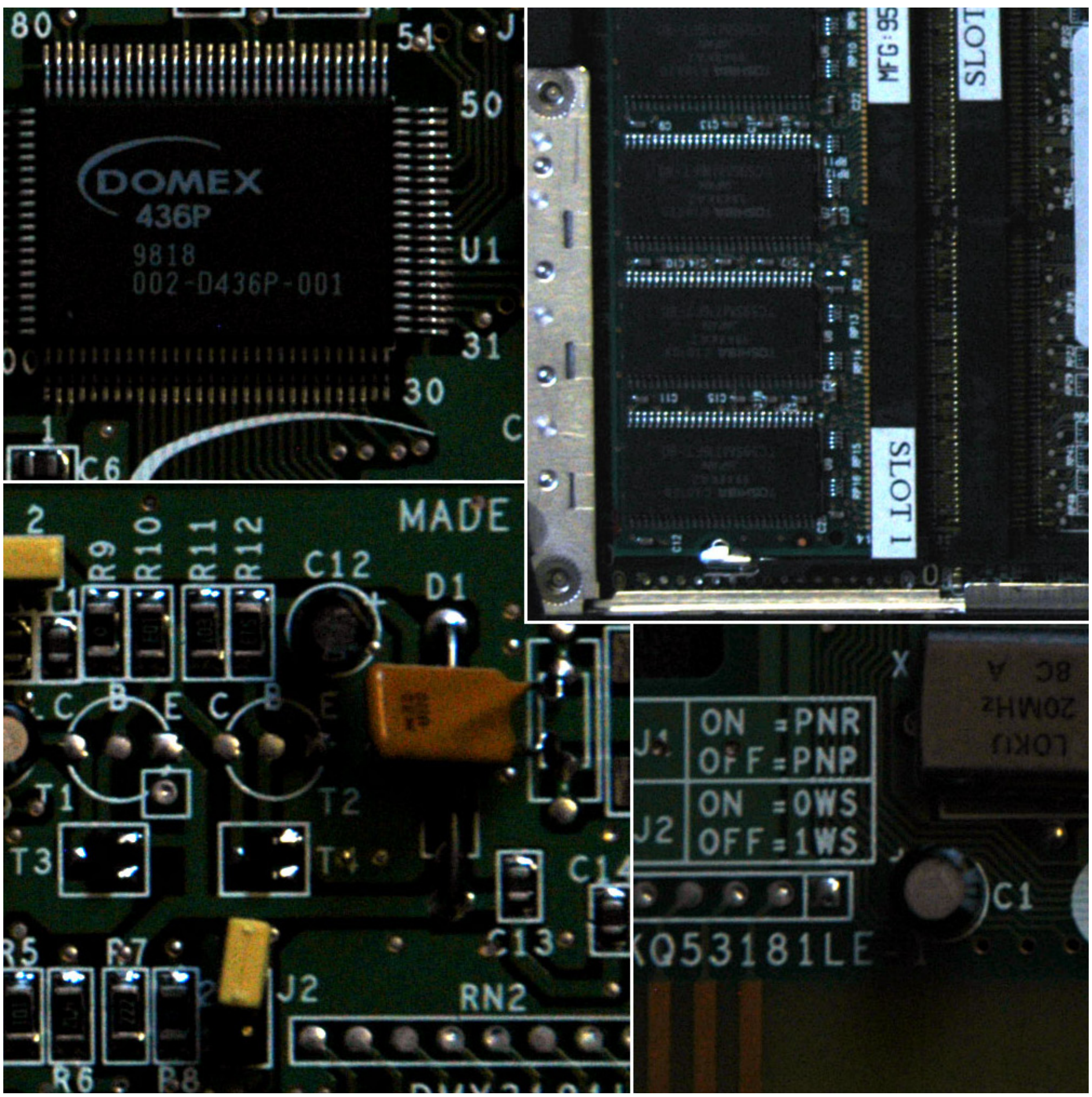
Since the three chapters on the core technologies will focus on their specific technologies, the examples may seem to include data that can only be managed by the technology under discussion. But that is rarely true. While it may be accurate to say that database management systems provide the most obvious and convenient mechanisms for storing attribute data about archaeological artifacts, those attributes may be recorded in files that are linked to a CAD or GIS data set. Similarly, a CAD model may seem to be the only way to present a complex structure, but much of the information may be retained in a database, and the maps created in a GIS may also be created in a CAD system.

In short, one of the points of studying in sequence the three technologies as they are used in archaeology is to try to understand what the benefits and limits of each seem to be – and to try to understand how they may be used together.

In particular, it is crucial to see how well-constructed databases are designed because they are central parts of GIS data sets and are often used with CAD systems. We will return to this issue of using the technologies together, but it must be clear from the beginning that a well-considered approach to computer technologies for any archaeological project begins with the specific needs of the project and should involve any and all computer technologies that seem appropriate. Those technologies should be fully integrated.

II

Computing and Computers



Glossary

Here is a general glossary to assist with this chapter. Many of the terms are ridiculously basic; some are less so, but all are important to the discussion. The basic ones may be unnecessary for many, but even experienced computer-users should examine the list to be sure that the familiar terms are used here as elsewhere in their experience.

Analog: information represented in forms that are or appear to be continuous, as opposed to digital information, which is or appears to be discrete. An analog watch represents the time precisely but with moving hands instead of displayed numbers; a digital watch represents time no more precisely but with numbers only. Digital displays seem more accurate/precise, but that is not necessarily so. (See digital.)

Application: a computer program intended to carry out real work for typical computer users, e.g., a word processor, a GIS program, or a Web browser. (See software; compare operating system.)

Archival copy: a copy of a file intended for long-term storage and retrieval.

ASCII: American Standard Code for Information Interchange – a standard for using eight computer bits as a unit (byte) to represent numerals and letters. This system is a standard that has long been used in personal computers. An eight-bit byte can create only 2^8 different characters (256), and only half of those – 128 characters – is actually defined by the system because, at the time the standard was defined, the last of the eight bits in each byte was used only as a transmission check. The last bit is no longer used as a check, making it possible to represent 128 more characters with the same 8 bits. Unfortunately, the standard having been defined already, the added 128 characters have been defined idiosyncratically by manufacturers such as Apple® and Microsoft®. Those definitions are not uniform. (See ISO standards.)

Attachment: a computer file sent along with – attached to – an email message.

Back-up: a secondary copy of any computer file, a copy intended to be used if the original is damaged or lost.

Binary: a number using two as its base (rather than ten, the base for the decimal system). Thus, 1111 in a binary system is equivalent to 15 in the decimal system.

Bit: a single electronic signal, binary in nature (on or off only, treated as zero or one).

Bus: the electronic conduits connecting various internal parts of computers so that information may be moved between/among them.

Byte: a group of bits (usually eight) treated together as a discrete entity.

CAD: computer-aided or computer-assisted design software and/or computer-aided drafting software.

CD (ROM): a removable storage medium for binary data that is encoded with optical signals rather than magnetic ones. The data may represent information (data) or instructions (software). CD-ROMs cannot be changed once the optical codes have been inscribed (ROM stands for read-only memory), though the CDs called CD-RW can be changed or re-written (RW for read-write). CD ROMs are superior back-up devices precisely because they cannot be changed.

CRT: cathode-ray tube. Old-fashioned TV-style display device, for displaying images on the nearly flat surface of a vacuum tube. (See LCD.)

DBMS: database management system.

Digital: based on digits (usually assumed to be binary digits only) and assumed to be electronic, magnetic, or optical in form, i.e., created by and used in computers.

Digitize: in more general usage, to translate information into digital form; in graphics applications to trace a drawing, map, or plan to create a digital version of the original.

Directory: a hierarchical grouping of files or of other (sub-)directories considered to belong together for any reason. Directories and sub-directories are defined by users. (In MAC® systems, folder.)

DOS (disk operating system): an operating system that assumes the presence of a disk, either a floppy disk or a hard disk to store the operating system software. (MS-DOS® is the name of the disk operating system originally made by Microsoft, an abbreviation for Microsoft disk operating system.)

DVD: similar to CDs, but DVDs can store several times as much information as CDs.

Extension: the portion of a file name in Windows (and now commonly in other systems as well) that follows a period and indicates the file type.

FTP (file transfer protocol): a system developed very early in the history of computer networks to enable the transfer of a digital file from one computer to another over a network. It is the file transfer system used on the web, though that is not made explicit to users.

File format: the specific encoding system used for a digital file and agreed upon by all who create or use files of that type. A file cannot be decoded by software unless the software designers know the decoding system.

Firewall: hardware and/or software lying between a computer and the Internet to protect the computer from unauthorized access via the Internet.

Flash memory: non-volatile electronic memory. Solid state memory devices that were very expensive, measured by the cost per unit of memory, but are now becoming less so and are being used to hold information such as digital images. CompactFlash®, SmartMedia®, Sony's memory stick®, and xD-Picture Card® are the current commercial versions available for digital cameras.

Floppy disk: a removable storage medium for storing binary codes using magnetic signals. The codes may represent data or software. The magnetic coding can be changed at will (intentionally via the computer operating system but also, intentionally or accidentally, by any magnetic field). Floppy disks were developed to make it possible to transfer data or programs easily from one computer to another by simply carrying a disk from one to another. (Original floppy disks were actually floppy, not rigid. The magnetic material was on a thin substrate that was not rigid, and the magnetic material and substrate were held within a paper package that added little rigidity. Current versions are rigid, since the magnetic material, though still not rigid itself, is held within a rigid plastic package.)

Folder: the Apple/MAC term for directory.

Freeware: software available without payment.

GIS: geographic information system.

GUI (graphical user interface): any system designed to permit users to direct a computer program with a pointing device (typically a mouse) and a variety of visual cues at which to point. Now more widely taken to indicate the system of visual cues and presentation formats created to aid users of a computer.

Hardware: the physical components of a computer, e.g., monitor, disk, keyboard.

Hard disk (hard drive): a stationary storage device for digital codes in magnetic form. The codes may represent data or software. The magnetic coding can be changed at will (intentionally via the computer operating system but also, intentionally or accidentally, by any magnetic field). The hard disk actually consists of a thin magnetic material on a substrate; the disk revolves at very high speeds while sensitive devices analogous to tone arms on old record players either measure the existing magnetic pulses on the magnetic material (to obtain data from the disk) or change the magnetic pulses (to put data onto the disk). What is colloquially called a hard disk usually consists of several disks rotating together on a common spindle as well as the mechanisms for rotating the disks, reading data, and writing data. Hard disks are normally not removed from a computer unless they have malfunctioned; they are sealed units. However, the entire unit may be removed, and hard disks in portable computers may readily be moved from one computer to another.

Internet: the cables (and now wireless connections) and routing boxes that permit computers connected thereto to send signals to one another. The Internet is a huge, world-wide network. Some computers on the Internet may do nothing more than supply files. Others may use more demanding communication protocols to carry out other cooperative tasks.

ISO standards: International Standards Organization standards, including those that specify computing codes. Similar to ASCII, the ISO standards numbered 8859-1 through 8859-15 specify characters for eight-bit byte systems (256 characters). The ASCII standard is used for the first 128 characters of all ISO 8859 standards, but the remaining 128 characters are different for different scripts, making it possible for people to use different versions of the ISO 8859 standard for different languages, e.g., modern Greek or Cyrillic. (See ASCII.)

JPEG (JPG): the file format developed by the Joint Photographic Experts Group for photographs. This is a compressed format, and there is some loss of image information when a file is compressed via a JPEG algorithm.

Java®: a computer language that has been implemented so as to permit a program written in that language to operate on any computer with the proper Java system. A Java application will therefore run on Windows, Linux, and MAC computers.

LCD (liquid crystal display): The flat-panel display types used in laptop computers and now on many desktops as well. Their primary advantages are size and weight, when compared to TV-like cathode-ray-tube monitors. (See CRT.)

Linux: an operating system based upon UNIX and maintained by an open standards committee. There are versions of Linux for PCs as well as Apple Macintosh® computers and larger, more powerful machines. Much software for Linux is free. The operating system itself is available for free. (See open source.)

Local Area Network (LAN): a network that is both relatively small and confined as to geographic area covered. All computers in a LAN are controlled by the same people/organization; so the level of cooperation can be extensive, although such levels of cooperation require some expertise on the part of the network administrators.

Macintosh: Apple's personal computer. The Macintosh operates with a different operating system than the PCs using Windows or MS-DOS (still used in some places, though more and more rarely). Users of Macintosh computers normally use software designed for the MAC operating system, the current version of which – OS X® – is based on UNIX®. The latest iterations of the MAC can also run Linux and Windows and their applications.

Malware: the general term for computer programs designed to damage computers or to gain control of them for an outsider. Viruses are one category; they are programs designed to damage the computer, usually to no purpose. Some viruses allow outsiders to use an infected computer for their purposes, not the owner's. Worms are designed to propagate themselves through the network, often doing no damage in the process. Trojan horses are the programs that carry surreptitious code within them for unexpected purposes, often to detect and pass on to others the keystrokes used, enabling passwords to be harvested.

Metadata: data about data. Some take the term to indicate the kinds of information that might appear in a library card catalog: information intended to help people locate relevant information (author, subject, date, etc.). Others take metadata to mean the information about data required to put it to good use (software required, file formats, vocabulary limits, etc.). Properly used, metadata should refer to both these kinds of information, which might separately be called indexing information (or resource discovery information) and data documentation.

Monitor/screen: the television-like, CRT viewing apparatus for a computer or an LCD serving the same function. (See CRT, LCD)

Network: a set of wiring, wireless transmission systems, and communication protocols allowing computers attached thereto to communicate with one another and possibly even work cooperatively. The extent of the communication/cooperation is determined by the protocols used and may be limited to such mundane things as file sharing or be as extensive as allowing individual computers to work on the same project simultaneously. The physical extent of the network is adjustable; a network may include only the computers in a small office, those in a large building, or something as large as the Internet.

OS X: the operating system, based on UNIX, for current Macintosh computers.
Open source: software that is supplied with its actual code accessible, permitting users to modify that code. Normally, any such modification must be shared, at no cost, with any other users. Open source software is usually available at no cost, though it may also be purchased with aids for installation or service assistance.
Operating system (OS): the basic program that must be loaded into a computer at start-up to prepare the computer for doing useful work. The operating system determines how the physical parts (hardware) of the computer communicate and interact. Applications such as word processors or web browsers operate on the computer via the operating system; so they need not operate directly on the hardware. Therefore, application software is usually written for a specific operating system.
Partition: a physically distinct portion of a hard disk that the operating system can treat as if it were a separate hard disk.
PC (personal computer): used generically to include the Macintosh and Linux computers here unless a clear distinction is stated; the term is often used to include only computers running Microsoft Windows.
Proprietary format: a digital file format controlled by a corporation. Such formats may be licensed to others, but the controlling corporation will not permit use of the format without permission and may change that format without notice.
RAM: random access memory. Electronic memory within a computer that may be found via a numeric address and therefore can be located directly, without looking through other memory locations. RAM is volatile; when electricity is turned off, the signals are gone.
ROM: read only memory. Non-volatile electronic memory that cannot be changed. (There are variations called EPROMs, for electronically programmable ROM, that can be changed, but changing an EPROM is intentionally rather difficult. EPROM memory is not volatile.)
Root: in Windows the broadest designation of the content of any given disk; the root directory contains all other directories on that disk. In UNIX and UNIX-derived systems, the root directory is not limited to a disk; it is the base directory in which ALL other directories exist. (A second or third hard disk in a UNIX system is considered to lie in a directory within the overall system.) In UNIX systems access to the root directory is limited to systems managers to prevent accidental tampering by users.
Server: a computer designed to hold data for other computers to access over a network.
Shareware: software that is available, without advance payment, for use/trial and for which a voluntary payment is expected if the software remains in use.
Software: program code that will cause the computer to perform requested functions. (See application, operating system.)
Spam: unwanted email, usually messages aimed at selling products but often scams of one sort or another.
TIFF (TIF - tagged image file format): a standard file format for images. Moving an image to this format should entail no loss of image information.
Unicode: a replacement for ASCII using 16 bits per character and consequently 2^{16} possible characters (65,536). It was once thought to be capable of representing all scripts and symbols, but there is already a Unicode standard based on 32 bits per character so that there really will be enough characters for all scripts, even hieroglyphics (4,294,967,296). (Unicode is equivalent to ISO-10646, which established both 16-bit and 32-bit standards.)
Vista®: the latest iteration of Windows, this operating system has been advertised as safer and more robust then prior versions of Windows. At this writing, the jury remains out as to its value; reception in the public has been slow.
Volatile: in the computer world, referring to a form of memory that requires electricity to function. When electricity is lost, anything stored in a volatile device is also lost.

Web: the portion of the Internet and computers connected thereto that supplies documents according to certain standards. Those standards permit the documents to be displayed by anyone with access to the Internet and appropriate software.

Wetware or grayware: a pejorative term for the human brain, often used when referring to the person operating a computer (who is considered less reliable than the computer, an assumption that may or may not be accurate).

Wide area network (WAN): a network covering a wide geographical area. WANs often use public networks such as the Internet to connect LANs to one another, forming a WAN. The Internet can be described as a WAN, but the term is normally taken to define a more centrally controlled network such as one set up by a specific company or institution.

Windows: the combined operating system and graphical user interface made by Microsoft and used on those personal computers that trace their ancestry to the original IBM personal computer. More generically, windows are the individual, bounded portions of a computer display in which a particular program or document may be used or seen.

The Basic Hardware

Computers are physical devices that use electrical states to represent numbers or letters and electrical currents to transfer those electrical states from place to place within the computer. Therefore, electricity is required. When computers are turned off, they lose all capacity to accomplish useful work. There must obviously be some way to store computer information in non-volatile forms that survive a loss of power. Originally, those non-volatile forms were punch cards and punched tape. Eventually, magnetic tape and magnetic disks came into use, and there are now optical and solid-state forms of data storage that are not volatile, with research on newer forms of non-volatile storage promising better products to come.

The basic parts of the computer are the central processing unit (CPU – the basic chip that does most of the actual computing, the best-known manufacturer of processors is Intel®), the bus (electronic conduits for transferring information among the various parts of the computer), the storage sub-system (disks or tapes), the memory (RAM), the keyboard (and mouse), and the display system (including both electronics within the computer and the display device or monitor). The CPU (and some other chips on the computer) can perform a limited number of mathematical functions (adding and subtracting being the most common), locate data in the memory, and direct traffic over the connecting cables (bus) of the system. The storage sub-system may include disks, tapes, or other forms of storage for program instructions and data to be used by the CPU (after transfer to the computer's volatile memory), and the keyboard provides a way to interact with the computer. The display system puts words, numbers, and images on the display device for the user. Of all these parts, the storage sub-system is most likely to be the logjam, because tapes and disks are physical devices that require a good deal of time to fetch information or to store new information. In archaeological use, for instance, large databases, CAD models, or GIS data sets will take a great deal of space on the computer hard disk; they will require time to fetch or save; saving and opening data files will often keep you waiting and twiddling your thumbs. Thus, since there are differences in speed of such devices, users may wish to spend the extra money required for faster devices under some circumstances; inexpensive computers rarely feature fast hard disks.

There are some interesting limits placed on computers that users should probably understand. For instance, computers can be purchased with various amounts of the volatile memory (RAM) used to store instructions and data while the computer is actually functioning. As operating systems, the software foundations that underlie everything else on a computer, have become more complicated – mostly because of the overhead imposed by the graphical user interface (GUI) – more memory has been required simply to hold the instructions of the operating system. Meanwhile, programs are capable of more complex operations; so they need more memory, as do the resulting larger and more complex data sets. Finally, modern operating systems permit the computer to work with multiple programs at the same time, adding to the need for memory. These forces have combined to increase the need for memory, and, generally speaking, computers need about as much memory as the buyer can afford. An estimate of the quantity of RAM needed might not remain current long enough to be useful here, but it is rare to find a widely advertised computer system offered with enough RAM for demanding users. For archaeologists in the field, who are likely to be working with word processors, database programs, CAD, and GIS (and possibly email or other Internet-related programs), as much RAM as can be afforded should be installed.

One particular problem commonly pops up when there is not enough RAM. If a program requires more RAM than is available, the computer will – without instruction – move data and/or instructions from RAM to the hard disk so that only the immediately necessary information is taking up the limited RAM. This slows down the work enormously because the computer is constantly moving information back and forth between the disk and RAM when it should be spending its time performing calculations – and because disks are relatively slow. Users can even hear the disk working as information is accessed and new information added (this is called disk thrashing); there is little as annoying as sitting and waiting for a file to be loaded while listening to the sounds of a hard drive constantly reading and writing data. When this happens, it is time to add RAM (or replace the computer).

Another potential limit for computers is the size of the hard drive (disk), the main data storage device. The hard drive has to be large enough to hold the programs and data regularly used – and to hold temporary files that are created constantly by the computer as it works. Those files can be quite large; so, when the computer is not being used, there should be a significant amount of hard disk space available. Hard disks have become so inexpensive that a buyer is now unlikely to find a computer with a hard drive that is too small for common uses, but data files for archaeological projects are often large, sometimes enormous; large disks are necessary.

The display system can also affect performance. For GIS and CAD programs in particular, the quality of the display system and the (separate) memory allocated to the display (by the internal electronics of the display/video card) will be very important, affecting both the quality of the display and the speed with which the display can be updated. In addition, the size of the monitor will have an impact on the utility of the programs. Larger monitors simply let more of an image – whether a map or a CAD drawing – appear on screen at one time and at a reasonable scale. For those using several programs at once – e.g., an email client, a database, and a word processor – having a large screen can save a great deal of time by permitting multiple windows to be open and visible at once. In general, a large monitor is more a convenience than a requirement, but CAD and GIS programs really do need larger monitors. Using CAD for either architecture or a site map and using GIS for complex maps make the use of large monitors (20-inch or larger diagonal measurement) very desirable. Otherwise, maps and drawings are too small to be seen clearly or show too little of an excavation or survey area at one time – or both.

LCD displays were too expensive for most academics when they were first introduced; so academics continued to use old-fashioned TV-like CRT monitors. Fortunately, prices have declined so that LCD monitors are now the norm. They have one great advantage: they take up much less space for the same size monitor. LCD displays also use the entire screen, whereas CRT monitors are often defined by the entire visible screen surface, only a portion of which is actually used for the image.

Laptop computers must be mentioned here, because they are so often used in the field by archaeologists. Modern laptops are the equal of their desktop competitors in most respects. RAM may be a bit more limited and should be checked; hard disk space may also be somewhat less generous, but hard disks are now small as well as inexpensive. Neither RAM nor hard disks should present a problem. Of course, the screens are smaller, but they are large enough for most work, and a separate monitor can be added when necessary for CAD or GIS work. One of the hidden advantages of using laptops lies in their resistance to power loss problems. That is, laptops will generally be used in fieldwork settings while plugged into the current; unlike desktop machines, however, they will keep on running if the power goes off. Their batteries will automatically take over. Thus, a laptop in the field is more than a convenience; it is a real necessity if electricity is not reliable. (If electricity is a rarity, the laptop will probably not keep running long enough to meet the needs of a field archaeologist. Batteries need to be recharged after, at best, a few hours of work.)

Along with the standard computers, hand-held devices may be used to record information in the field. Choices for such equipment will depend on many issues including function, price, ease of use, and the ease of communications with site computers. In addition, this seems to be an area where change can be expected to be especially rapid, given the increasing miniaturization and the combinations of devices now appearing in the marketplace, e.g., the iPod-cell-phone combination and cell-phone-Palm-pilot combinations.

Using small networks in the field (and larger ones in the office, most likely) can make it easier to hold down hardware costs. In a network environment, for instance, only one computer, the one used to store the data files, must have a very large hard disk drive. Similarly, only the computer(s) to be used for CAD or GIS require a large monitor, and computers used for less taxing work (word processing or database data entry, for instance) may not need extra RAM.

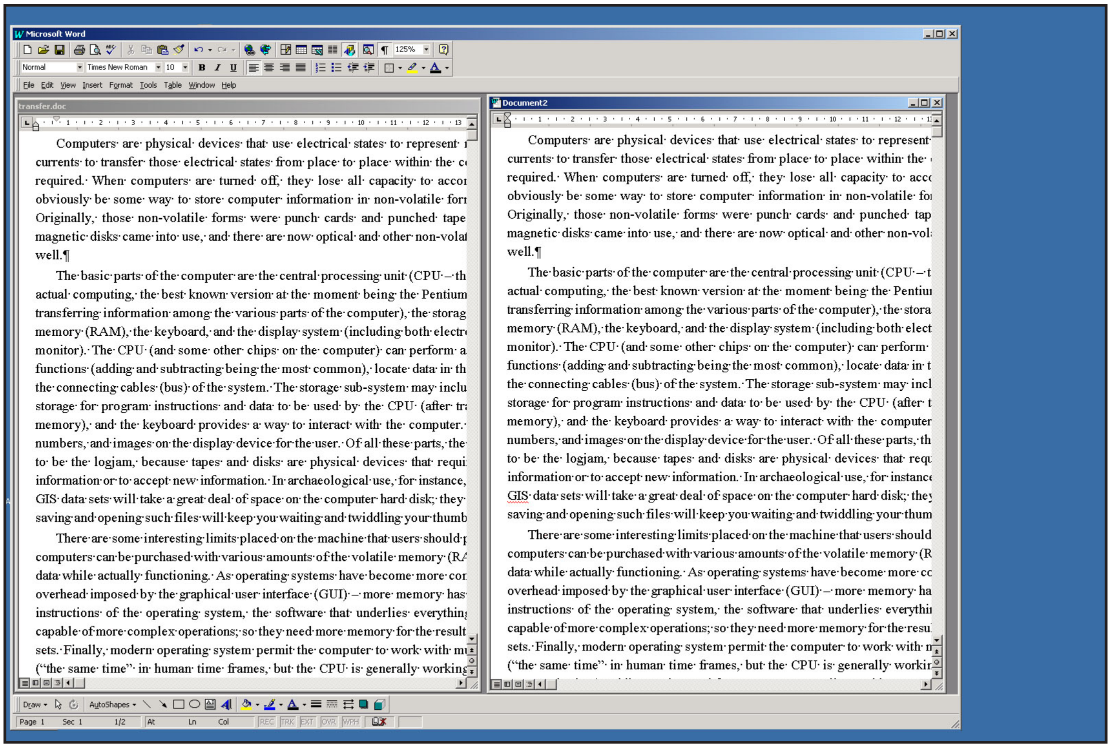
Start-Up Software: Operating Systems

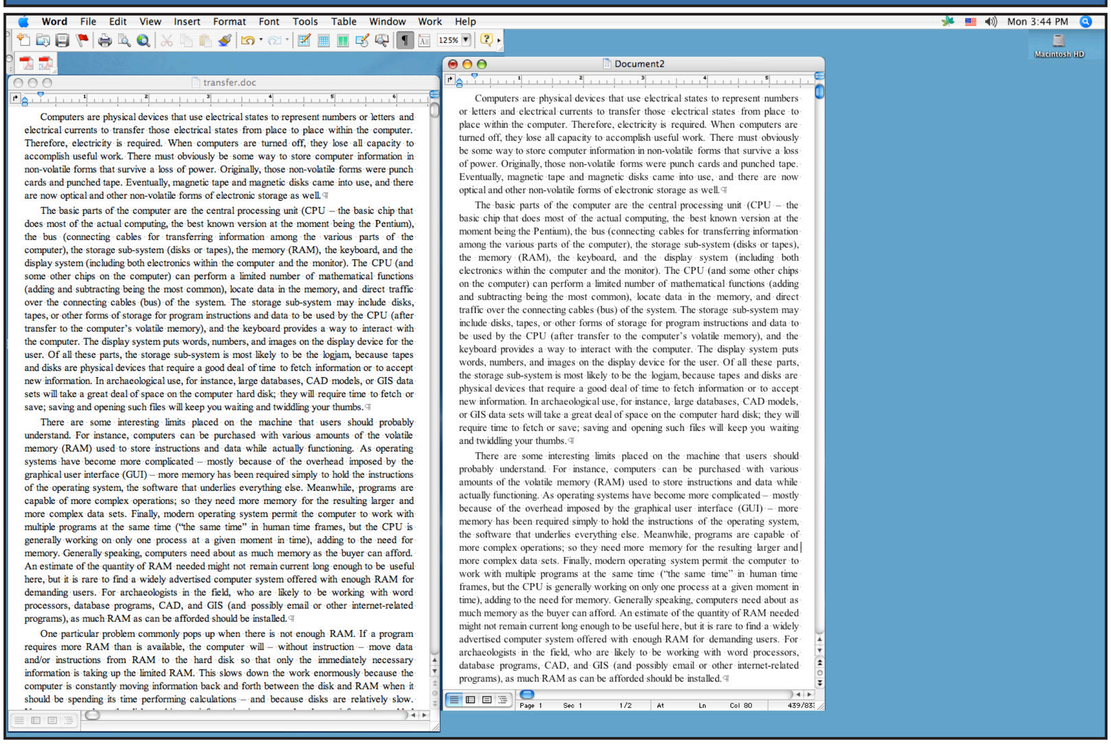
Computers start up with a basic program called an operating system (OS); for some time the operating system was called a disk operating system or DOS, hence Microsoft DOS as the early PC operating system. The OS seems to do little or nothing for the user, but it enables all the other programs to do their work because it provides the instructions for operations between and among the various parts of the computer. Modern operating systems also control many aspects of the interaction between user and computer.

A discussion of the advantages and disadvantages of specific operating systems can resemble a discussion of religions, with all participants taking fixed and inflexible positions about which OS is to be preferred. The OS contenders for users of desktop computers are Windows (from Microsoft, Vista being the latest iteration), the OS X (for the Apple Macintosh), and Linux (a free operating system based on the AT&T operating system called UNIX, which also underlies the current MAC OS). Many computer users despise Windows because it is a Microsoft product; others insist that Windows is the most popular OS for desktop computers for good reasons. Apple users have always had a certain quasi-religious zeal about the Macintosh, perhaps because they have for long been such a minority, perhaps to justify paying higher prices for the MAC hardware in the past (no longer necessarily the case). Linux use is growing because of the cost of the OS (nothing for more sophisticated users and inexpensive even for the novice) and the anti-Microsoft attitude among some PC owners.

There is no single OS that can stake a legitimate claim to being clearly superior in all respects. More programs run under Windows; that is, more programs can run on a computer with the Windows OS. Linux is the least expensive, and most of the software running on Linux machines is free. The MAC probably has the most polished user interface and, at least arguably, the best graphics. At the moment, the typical archaeologist should probably choose Windows, because there are more and/or better CAD and GIS programs for Windows computers, though it is said that at least one of the most highly regard GIS systems is preparing a version for Linux and the MAC. For database programs the situation is different, with the MAC and Linux machines having equally good choices. It is also possible to emulate Windows on a MAC in order to run Windows programs, and a user may run Windows and Linux on the same PC, switching between them as needed (without shutting down the computer). Finally, in the fast-changing computer world, the MAC's new use of Intel processors has made it possible to run Windows on a MAC at speeds comparable to those found on standard PCs.

This reluctant endorsement of Windows (I use all three OSs and consider Windows the least desirable of the three) requires a warning. Although Vista, the newest iteration of Windows has helped, problems with Windows seem remarkably persistent, often involving subsidiary programs because they are so closely tied into the operating system. Windows is also the target of choice for Internet virus writers. Virtually all the virus programs that are spread via email assume the use of the Windows operating system and are designed to attack Windows computers, not MACs or Linux machines.





Figures 1 and 2

A screen from a Windows system (above) and a MAC system (below), both running MS Word® with two documents open. The Windows system has one window containing both documents; the menu is part of the window. The MAC windows are independent, each containing a document; the menu is at the top of the screen. (Note: text appears smaller on the MAC because the monitor is physically larger, requiring more reduction of the image to fit here.)

The Windows system fits all document windows in the master program window. The MAC puts each document in an independent window. If multiple programs are running, the differences become more obvious, with a Windows system showing one window per program and the MAC showing one window per document. Other differences are minimal, since Windows and the MAC OS have become more and more similar over time.

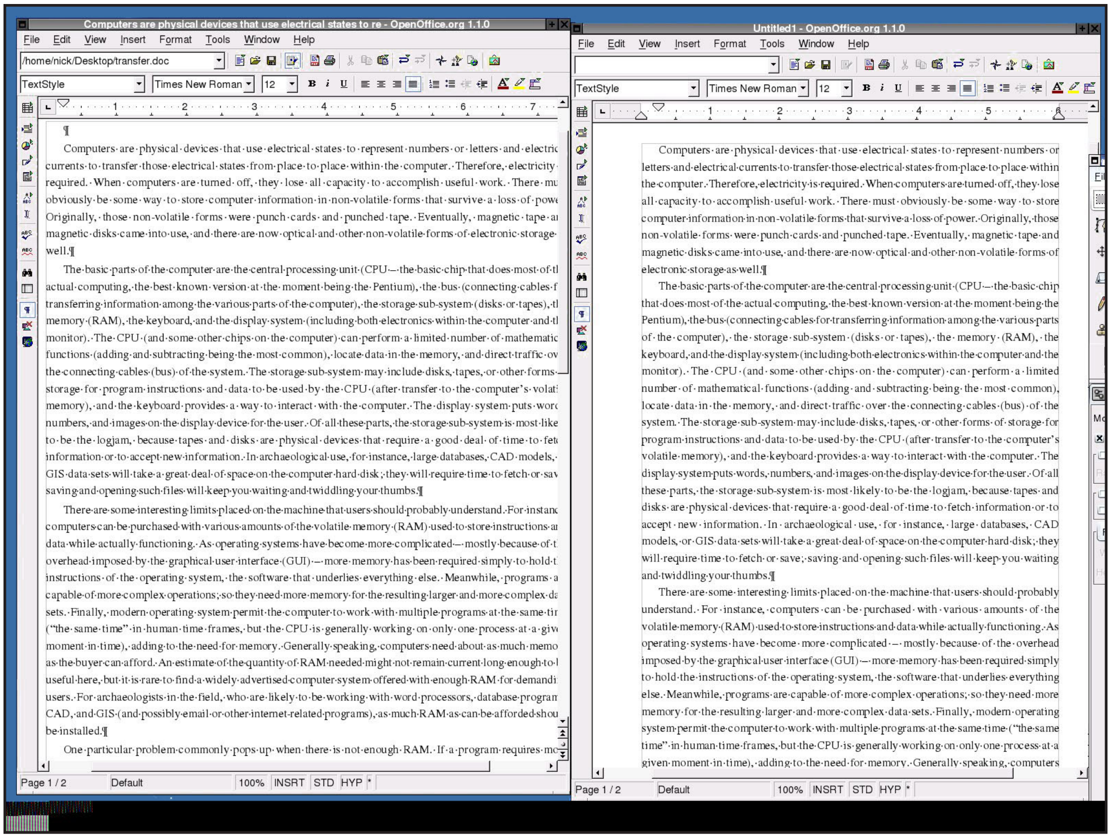


Figure 3

A screen from a Linux system with a word processor running and two documents open. Like the MAC, Linux usually places the documents in independent windows, but there is a separate menu for each window/document. (A portion of another window is visible to the extreme right. That window is related to the graphics program used to make the screen image.)

In recent years Windows has endured additional problems as its security has been repeatedly compromised. Because of the integration between the OS and many of the applications, especially Internet Explorer® and the email program Outlook®, Windows computers seem to be uniquely vulnerable to Internet-distributed viruses and worms. The problem is exacerbated because the OS itself is unusually complex and therefore hard for Microsoft to update. Recent problems with the security of Internet Explorer have added fuel to the fire of objections to Windows. Vista has been better, according to press reports, but it has not caught on well with users, many of whom (this writer included) have chosen not to upgrade and have stayed with Windows XP.

Archaeologists, like other scholars, will probably find that Windows is the OS to use for a variety of reasons. The needed programs will run on Windows; the college, university, contractor, or data repository may well support only Windows; more graduate students and other staff members will be familiar with Windows; and the computers themselves are ubiquitous. Because of the security problems mentioned above, two kinds of subsidiary programs are required: firewalls (to prevent direct access to your computer from outside) and virus protection programs. These should be installed along with the OS and configured according to current standards. The data collected by any archaeologist is too precious to remain unprotected. In addition, Windows users should be sure that they have a full Windows CD, not only the "restore" CD that may have come with the PC. In the field, a user should be prepared to reinstall all software and data on any given PC at any time. The foregoing is not meant to inspire fear; serious problems are truly rare, but a large project simply cannot withstand the loss that could accompany a major malfunction in the field.

As ever, changes in the computer world are so fast that, even as the foregoing is proofread, its accuracy may be changing. Running Windows on MACs greatly reduces the advantages of Windows-only approaches. Only time will tell whether using MACs to run both the MAC OS and Windows is practical, but it is certainly possible. (Since writing that I have been using Windows on my MAC to run AutoCAD without problems for well over a year. I have been pleased to be able to continue to use the MAC for all other programs and to use Windows and the MAC OS at the same time so that information can be moved back and forth between AutoCAD and the MAC.)

> **Choosing and Configuring a Computer**
>
> Buying computers is not very complex today, but those most widely advertised are often not properly configured for users like archaeologists who will have special needs. These are the issues to consider:
>
> CPU - Generally speaking, the fastest processors at any given moment do not offer much added speed when compared to the considerably less expensive versions of those processors operating just a bit slower. Intel and AMD® processors are equally acceptable, though some buyers are unnecessarily wary of AMD processors.
>
> RAM - As of this writing (mid-2008), at least 2 GB if database management systems, CAD programs, or GIS software will be used on the system.
>
> Hard Disk - Again as of this writing, a 250 GB hard disk is a minimum choice in terms of price and performance if the computer will be used for complex tasks. Speed must also be considered. (Adding a hard disk, especially an external, one is very simple and not very expensive.)
>
> Graphics Card - The most expensive graphics cards are often aimed at game-players, but good graphics cards are required for larger monitors (which have higher resolution) and are desirable for CAD and GIS users.
>
> Monitor - Large monitors (20-in. or more) are desirable for CAD and GIS users. LCD monitors are now so standard that there is no reason to consider an old-fashioned CRT. (Some users will want to check power consumption of the monitor since there are efficiency differences.)
>
> CD/DVD reader - Most computers will have disk drives for CDs and DVDs; however, some can read a disk but not write (create) one. Because archaeologists will virtually all be making CDs or DVDs for data storage, their computers should be able to read and write CDs and DVDs.

Applications and Macros

The operating system is intended mostly to prepare the computer for real work by controlling the internal communications that are required for that real work, although more and more features have been added to OS software. It is the applications or application programs that carry out specific work for the user with the aid of the OS. Typical applications are email clients, word processing programs, database management systems, Web browsers, and so on. While many may be operating at once, each has a relatively narrow range of duties, and the cross-over between and among them is generally rather limited. Any application must be tailored to a specific OS, though many programs are offered for multiple OSs by using a core of code that can be used anywhere and adding specific coding for individual OSs only as necessary. (The great appeal of the language called JAVA is that code can be written once and once only to run on a variety of OSs.)

The OS takes care of putting files onto disks, retrieving them, and accepting keystrokes from the keyboard. Applications, on the other hand, create the files to be stored, display the information from the files, and turn keystrokes into characters or requests for actions. Modern systems sometimes confuse the two a bit, with the OS, for example, helping to provide program functions through the user interface that is part of the OS. The division of labor is not so absolute, but one might compare the OS to an automobile engine and application programs to the transmission. Without the engine, the transmission is useless; without the transmission, the engine is simply standing in readiness.

Both operating systems and applications are programs, stored on disks when the system is off and retrieved into memory either automatically at startup (the OS) or on command from the user (applications). In addition, many applications permit the user to write simpler software using either a general computer language or a pseudo-language supplied with the application. The user-written software – usually called scripts or macros – allows a more sophisticated user to add features that make the program more efficient for that particular user. In some cases, the added programs, scripts, or macros are truly necessary; for instance, an archaeologist who wants the computer to check an artifact weight as it is entered into a database to be sure that it does not exceed pre-set limits must write some code. This may be done by using a set of menus and selecting pre-defined choices, but it generally results in something one would call computer code. More complex code may be used to automate data entry, as, for instance, when using survey data to generate a part of a CAD model directly from stored coordinates gathered by field survey.

Choosing Software

1. Start with a list of features you need.
2. Make sure the product does the job you want. Do not accept a program that cannot perform the functions you want and need unless (a) you have found no program that can or (b) you must give up one function to get a more important one.
3. Check the availability of training manuals beyond that supplied by the manufacturer. A widely-used program will have a large enough user base that such manuals will have been written; third-party manuals provide both more sources for learning to use the programs and proof that the programs are widely used.
4. If a program is not widely used but does seem to have serious advantages, be sure that its data are written in or can be exported to a common, preferably non-proprietary, format.
5. Check for price and possible academic discounts.
6. Check for support and comments from your internal computer experts.
7. Check with colleagues about their experience, especially those you consider knowledgeable. This is especially important if you are considering software that is not widely used.
8. Check the web or periodicals for reviews of software about which you have any doubts or questions.

The typical field archaeologist using a computer system may never need to write code, but anyone who is preparing a good computer system for an archaeological project will need to be able to write at least some code. Furthermore, anyone with oversight responsibility for the system must at least be able to review the code with the specialist to understand its function.

An archaeologist attempting to review the system must be able to examine and appraise the code as part of the review process. A reviewer will need to appraise all the code used, since that code may affect data entry or data retrieval. This means, of course, that an archaeologist who reviews a data system must be able to examine the specialized code created by the developer to be sure of its quality. This is vastly different from the kind of review process with which archaeologists are familiar. Field archaeologists did not need to check such arcane matters when dealing with site architects or photographers, for example; the results spoke for themselves.

Data users may need to be sure that they understand the data retrieval processes and any code affecting them, but users need not examine all the code used in a system. In actual practice, archaeologists using data files will tend to ignore the code and simply use the data. There will be occasions when that leads to misunderstandings, just as use of data from published reports in the past has occasionally caused problems when the full excavation report – complete with information about excavation or survey methods – has not been read with care.

Choosing Hardware and Software

Choices of hardware and software may depend on the project director(s) or on the preferences of the home institution of the project. Support at the home institution for specific hardware or software may require or encourage archaeologists to use specific computers – by type or by brand – and specific software.

Reliability is key for hardware. Cheap hardware that may break down in the field will suddenly seem very expensive if it fails. (A project in which I was involved was on hold for the better part of a week, leaving people in Pompeii with nothing to do, while a replacement part was sent from the U.S. While the cost of the equipment was not an issue in this instance, the cost to the project of the lost time was incalculable.)

More to be feared than inexpensive hardware is home-grown software, not the macros or scripts referred to above, but full-scale applications – a database management system, for instance, written in some computer language by a computer science grad student or an archaeologist with extensive computer training – instead of commercial software. While it may seem more expensive, commercial software or similar Linux software should be the standard. It is more likely to be robust; there will almost certainly be better training available; there will be manuals (perhaps not as good as in days past, but at least on-line assistance); the software will stand a much better chance of being upgraded to keep pace with other changes in the computer world; and keeping the author of the software on staff (and happy enough to stay) will not be an issue. In addition, when commercial software is used to create data files, those files will be in formats that are more likely to be understood by and accessible to other programs than the files generated by a made-to-order system (see discussion of data encoding below). Commercial software has its disadvantages, particularly frequent upgrades that seem too expensive and regular changes in file formats. Nevertheless, it is generally much less expensive and more efficient in the long run, not to mention more reliable. Scholars – no matter their field of study, but especially those like archaeologists whose data may be used by many colleagues over a very long time span – are especially well-advised to use commercial software precisely because colleagues will need access to the data. Commercial software makes access over time and distance more reliable.

The smaller the project, the more important it is to choose software that is widely used. Such software, generally speaking, will have more numerous commercially-produced manuals, more users from whom to get advice, and file formats that are stable and efficient.

Specific recommendations for software will not be made here, but choices will be discussed in the following chapters for database management, CAD, and GIS programs.

The Data Files

Regardless of the equipment or operating system used and regardless of the application programs in use, the most important task for any computer user anywhere is keeping track of the data files and protecting them. If an archaeologist loses the computer data, there may still be paper files to return to, but, at the least, the loss of time and effort will be enormous. There is much more involved than making an occasional – or even a regular – back-up copy of files. The more important the information being recorded, the more important it is to protect it. Therefore, a plan for regular backing up of files should be a part of any project. Backing up does not mean making a copy on the same disk; it means making a copy that will be safe from any accident that may befall the main file. In today's world, the easiest way to back-up data is probably to write the files to CDs or DVDs. CDs and DVDs are cheap today, and they are probably the safest medium for data files because they cannot be altered after being written, making the files less vulnerable. (CD-R not CD-RW disks should be used to prevent re-writing.) Note, however, that there are many manufacturers of CDs and DVDs; for critical use such as this, the disks considered best and least susceptible to damage should be used. Check the web to find out which manufacturer(s) can be trusted.

An alternative is to back up files via the Internet on a separate computer. For field work this presents a risk because access to the Internet may be intermittent;

CDs or DVDs made in the field are always there to be used. Archaeologists will doubtless be working on the Internet from the field in the future, but access must be better guaranteed before the typical archaeologist dares plan to have constant and dependable Internet access from a field project. (In addition, the constant battle between internet service providers – ISPs – and users over the bandwidth used makes it somewhat risky to plan on internet access for backing up files. The ISPs have too much power and too little supervision at the moment.)

There are potential problems with safety beyond the failure of the computer. People working on the site may accidentally expose files to loss or damage, especially if the working computer is connected to the Internet; so care must be exercised to prevent accidental or malicious damage to the files that are being maintained. More will be said about this in a later chapter.

File Organization

All computers store their files in batches of one kind or another. Files may be on different physical devices, on a physically distinct portion of a single disk (usually called a partition), or in hierarchical arrangements such as might be used in a standard paper filing system, with files grouped according to one or another rubric and stored according to group and sub-group.

The most common system for organizing files on modern personal computers involves the use of hierarchical storage groupings; these are generally called directories on Windows machines and folders on MACs. Directories and folders do not imply any particular physical location on a disk; rather they reflect the computer operating system's underlying procedures for storing, locating, and retrieving files. If a user declares there to be a directory (I will not bother to say "or folder" from here on) called MyFiles; the computer will store that information and be prepared to save, find, and retrieve files by asking the user to navigate to the MyFiles directory to find them. More important, the user can create a complex hierarchical system of directories inside directories such as a directory called MyFiles containing both files and another directory called DataBaseFiles that, in turn, contains both files and another directory called MyProjectA that contains, in turn, both files and another directory called OriginalFiles, and so on. Of course, each directory may contain many directories, sometimes called sub-directories, not just one. A rigidly hierarchical system might not permit individual files in any but the sub-directory at the lowest level.

A Windows computer will have one general directory per disk (root directory) in which all other directories on that disk exist; it will carry the name of the disk on which it exists plus the backslash, for instance "A:\," "B:\," or "C:\;"[1] so the letters indicate the disk, and the backslash indicates the root directory. (In Windows the disks A and B are normally reserved for removable floppy disks, and C is the first hard disk.) Each subsequent hard disk will normally carry the next letter in the alphabet, although naming the disks is also possible. The MAC OS treats disk drives in a somewhat less obvious fashion, but the system is, at root, the same, as is true for other operating systems. Our system above, then would have organized directories as follows: "C:\MyFiles\DataBaseFiles\MyProjectA\OriginalFiles," assuming the directories are on disk C. Unambiguous file names would then be in these forms, with the last name being that of the actual file:

C:\MyFiles\DataBaseFiles\MyProjectA\OriginalFiles\pottery.dbf
C:\MyFiles\DataBaseFiles\MyProjectA\geninfo.doc
C:\MyFiles\DataBaseFiles\geninfo.doc
C:\MyFiles\DataBaseFiles\MyProjectA\FinalFiles\pottery.dbf
C:\MyFiles\DataBaseFiles\MyProjectB\geninfo.doc
C:\MyFiles\DataBaseFiles\MyProjectB\OriginalFiles\pottery.dbf

[1] Users of the Web will be familiar with the use of the forward slash to separate directories and to separate file names. The backslash in Windows is functionally equivalent to using the forward slash in a URL.

As the foregoing discussion should imply, making the directory structure clear and unambiguous can be very valuable. A clear structure will lead any user to the right file without confusion or delay. Even if the project files require multiple disks, directories should be structured so that a user can determine which disk to use and which directory of that disk. Under some circumstances systems can now permit multiple "virtual directories" so that a file can be found via more than one hierarchy; you may have a similar capability in your email system, some of which now permit multiple mailboxes to seem to contain the same email, using rules to determine content of each mailbox but actually storing each email only once.

Data Encoding

There are three important issues about computer data that should be covered here, before moving on to specific software types. All are somewhat arcane but important for scholarly computing. In fact, these issues are precisely the kinds of issues that concern archaeologists and other scholars but few in the business world and few home users because the issues have to do with long-term availability of data and with access to data by people well beyond the office, the local institution, even the country from which the information comes.

Encoding Files

The first of the encoding issues has to do with the encoding scheme used by a program when data are put into a digital file. Every program must have at least one encoding scheme that permits it to store the data and then to retrieve the data correctly. For instance, this text was written with a word processor that can save the file in several different ways, each involving a different encoding scheme, and the program must be able to open any of those files again for more work. However, the final editing and layout were done with another program that uses a completely different set of encoding and decoding instructions. These processes of encoding and decoding require that the encoding and decoding schemes be explicitly and fully specified. Very sophisticated coding schemes must be used by database management systems, CAD programs, and GIS software. In each case, the importance of the archaeological data stored makes the coding scheme a crucial piece of the whole.

Some encoding schemes are public, fully specified for anyone to use at any time. For instance, a simple text file may consist of nothing much more complicated than a series of characters. Even then, however, software designed to use the file must be written to take data in eight-bit bytes (groups of eight on-or-off electrical states) and to apply the appropriate standard to deconstruct the numeric information, replace numbers with characters, and create a succession of characters for display on screen.

Most encoding schemes are much more complicated, often including headers to carry information that the user will never see. Most encoding schemes are also proprietary; they are owned by a software company and cannot be used by others without paying a royalty.

These issues of encoding are generally discussed under the rubric *file format*. The file format is the expression of the encoding scheme. It is, despite the seemingly arcane nature of the issue, extraordinarily important to all scholars because scholars – especially archaeologists – must be concerned about the longevity and continuing use of their data files. Data in proprietary formats may be harder to access, requiring specific programs to open the files, and any format may go out of use, meaning that data in that format will eventually be inaccessible. If the software to read and write data in a given format goes out of use, the data files in that format will, sooner or later, be useless.

The problem for archaeologists is thus a double one. Data in proprietary formats will be harder to share with colleagues and should thus be avoided whenever possible, and all data formats are likely to be temporary, sooner or later to be replaced by others. (Formats are often named by the three-letter extension added to files after the period in the DOS/Windows file system. Thus a file named text.txt is assumed to be using the TXT or TEXT format; one named picture.jpg is assumed to be in the JPG or JPEG format.)

Proprietary formats cannot be avoided completely, but they can be shunned far more often than most people realize. For instance, Microsoft Word is the most commonly used word processor today, and it is an excellent tool. Most academics use it. Those who do, however, often ignore the fact that Word can store files in non-proprietary formats as well as Word's proprietary format (DOC).[2] In particular, Word can read and write the Rich Text Format (RTF), a public format created by Microsoft, and any Word document can be transformed into a PDF file. Therefore, scholars should use either the RTF or PDF format for the final version of a document meant to be widely shared, whether they use Word or one of its competitors to create the original. Doing so will broaden the reach – and the life – of any document. (Note that the more a document is manipulated for appearance sake the harder it is to store that document in most non-proprietary formats; PDF files, though, should precisely mimic the appearance of the original file.)

On the other hand, there are proprietary formats that must be used because the best programs of a particular type require them. In that case, users must be very careful to use formats that are, at the least, very popular. Popular file formats may go out of use, but their popularity will mean that someone will make translators so that the data can be moved to new formats. (Some proprietary formats have particular features that are seductive but virtually impossible to translate into any other format. Such features should be avoided, though it can be very difficult for an inexperienced user to recognize them.)

Regardless of the format used – be it public or proprietary – there is a long-term problem. File formats are not static; new versions of standard software often use new file formats, requiring older files to be changed. Moreover, the formats of virtually all files will eventually become obsolete. As a result, all lasting data files, if they are to remain useful, must be held in an archive where moving the data into new formats as necessary can be accomplished (a process called data migration). We will return to this issue in a later chapter, but it is important that, from the very beginning, everyone involved in scholarly computing attend to these issues that affect long-term access to scholarly data. Archaeologists do not excavate so that their data can be accessed for a few years but so that their data can be accessed for decades; providing long-term access is a critical duty of the profession.

Encoding Characters

The second of these arcane encoding issues has to do with the encoding of simple script characters. U.S. computers and many others are configured to use the ASCII code (American Standard Code for Information Interchange) to define the relationship between the numbers stored by computers and the characters they represent in text files. For instance, ASCII defines *a* as the number 97 and *A* as the number 65. All the other Latin characters are similarly defined, along with the space, numerals, punctuation marks, and a few special characters (roughly those of the typical keyboard plus some special characters required for computing such as an end-of-paragraph indicator). Therefore, any program can parse a string of numbers known to represent letters by simply translating the numbers, eight bits at a time, into the appropriate letters. The only required information is the table of equivalents and knowing that the numbers are parsed in eight-bit groups called bytes. This string can be parsed readily: 01000001011100 10011000110110100001100001011001010110111101101100011011110110011101111001 to yield the following, with the aid of some binary math and a translation table:

2 The new versions of Word can be set to use the DOC format, but by default they use a new format. Microsoft has tried convince users that the new format is not proprietary, but it is, alas.

byte (binary number)	decimal equivalent	letter represented
01000001	65	A
01110010	114	r
01100011	99	c
01101000	104	h
01100001	97	a
01100101	101	e
01101111	111	o
01101100	108	l
01101111	111	o
01100111	103	g
01111001	121	y

Why on earth should such an arcane matter as the code table for characters concern any archaeologist? The answer lies in what the code table does and does not include. The Latin alphabet is included. No characters beyond the Latin alphabet are defined in the ASCII code – no complex characters that require accents, tildes, umlauts, or cedillas. As a result, non-English European languages may provide unexpected difficulties. U.S. operating systems use variants of ASCII to include some non-Latin characters in the undefined part of the code (128 positions). The additional 128 characters, unfortunately, have not been defined identically in the various manufacturers' code books. For instance, Apple uses ASCII code 240 to represent the Apple logo, but in Windows ASCII code 240 represents the lower case *eth*, a character used in Icelandic. Even had they been identically defined, there were not enough characters to represent all the characters in all European scripts, much less non-European ones.

Another standard for character encoding was developed by the International Standards Organization. It used the first 128 characters of ASCII but then provided several variations (named ISO 8859-1 through 8859-15) – each defined as a separate standard – for the remaining 128 characters. All the European scripts could be handled in at least one of the standards. People using one of those standards have the keyboards "mapped" so that the keys call the appropriate codes, just as a standard PC keyboard calls for 65 when the upper-case A is typed. (In ISO-8859-1 the number 240 represents that Icelandic character, *eth*; in ISO 8859-5 the Cyrillic number acronym; in ISO 8859-7 the Greek lower case *pi*, π.)

Neither ASCII nor the ISO standards can deal with the requirements of certain other scripts, for instance, ancient Greek, which uses both modern Greek characters and accent and breathing indicators that are not used in modern Greek. Therefore, another solution was created for scholars who needed to use ancient Greek. Special fonts were developed to substitute ancient Greek characters, with accents and breathers where needed, for the normal Latin ones. In these cases, the codes were not changed but substituting a special font effectively replaced the codes with Greek coding; thus, a user might get Δ on screen when typing D, with both Δ and D being represented by the number 68 (01000100); the choice of font determines whether one or the other glyph appears on the screen or printed page. This is an effective solution, but there is no defined standard, just a specific font vendor's particular, idiosyncratic substitution of Greek characters for Latin ones.

Consider now the situation for someone who receives a simple text file with a specific coding scheme but not a coding scheme specified to the recipient. If the text requires only Latin characters, it should present no problems at all. A computer set up with ASCII (any of the variations) or an ISO standard (any of them), will parse the file correctly. Only by mistakenly applying a font that substitutes some other characters for the Latin ones (e.g., ancient Greek) would the recipient find unrecognizable text. But what if the text is in modern Greek or a language that requires Cyrillic characters? Or a European language that uses accents or tildes?

The recipient needs to have the appropriate ISO character set and to know which one is appropriate – and how to make it the operative set for the moment. For instance, the ü is defined in ISO 8859-1 as code 252. ASCII does not define it, but it may be found at 159 in the undefined upper range of ASCII – if the user is operating a MAC but at 252 if the user has a Windows machine.

Unstated in the foregoing is the problem created with file names. Not only the file content, but the name may use characters not included in the character set used by any given computer. This is at least as serious a problem.

This problem has not gone unnoticed, and a new coding standard has been in preparation for some time. It is called Unicode, and it uses 16 bits for each character instead of 8. As a result, more than 65,000 different characters can be encoded, and that is enough for the glyphs in most scripts. (An additional standard is evolving as well, applying 32 bits to each character so that all glyphs can be represented.) Linux, the MAC operating system, and Windows can use Unicode, and it would seem to solve most scholarly problems, since all modern scripts can be accommodated. The problem, however, is that the inclusion of Unicode has not been either complete or pervasive, and there are few Unicode fonts available. As a result, documents are rarely saved in a form that utilizes Unicode, and those who must deal with documents in different languages and scripts have no simple way to be sure that they can communicate effectively. Archaeologists are uniquely subject to the complexities of this problem. Many of us work with colleagues in other countries whose scripts include tildes, umlauts, cedillas, and so on. Communicating can be complicated by this issue, and it is important to realize that. Imagine, for example, trying to create a coding scheme for a pottery database and trying to use only characters that are the same in all languages of all participants in the field project. Assume further that the project includes scholars from countries using the Cyrillic alphabet, the Greek alphabet, and the Turkish alphabet, and you have a surprisingly limited number of acceptable characters.

Archaeologists cannot change these encoding or file format problems. They are discussed here to emphasize the extent to which seemingly obscure computer issues can complicate the scholar's work and to show that someone involved in a project must understand these issues. There are relatively few other computer users who need both the longevity of computer files and the multitude of languages that archaeologists need. As our needs are unusual, so our attention to these problems must also be unusual. If we assume that someone else is keeping an eye on these matters for us, we will be sorely disappointed.

Data Compression

The final issue of encoding is data or file compression. It is common to try to shrink files for storage or for sending them over the Internet; doing so is called compressing the files. One method of compressing files has become so common that the files are no longer considered compressed files by many users – the JPEG image file (.JPG). Originally a way to compress files developed by the Joint Photographic Experts Group, the JPEG format has become accepted as a standard way to encode images. It is however, what is called a lossy format. That is, an image file, when compressed with the JPEG standard, loses some of the information contained in the original. A compression method that does not involve any loss of original information creates, not surprisingly, called a lossless format.

In general, important data files should not be compressed. The risk of information loss exceeds the benefits. In reality, however, file compression is so common – the JPG file in particular – that a blanket recommendation against file compression is pointless. It may, however, be reasonable to say that no data files should be backed up or stored as compressed unless they are normally used in that compressed form by standard software, that is, without being expanded into some other format before being used.

Graphics Issues: Two Different Approaches to Images

Although we will not be spending much time with graphics explicitly in this book, there are important graphics components to both CAD and GIS programs, and there is a very basic distinction between two approaches to graphics that must be clear in advance. Computer screens are designed to present images as individual points on the screen, each with a set color and brightness value. The screen must be approached in that way, since it does, in fact, display individual dots (called pixels, or, less frequently, rasters). The graphics we see on a computer monitor are composed of those pixels.

There is a problem with pixel-based images. They do not scale effectively. That is, an image with 300 pixels on a side (90,000 total pixels) cannot be enlarged effectively to have, say, 800 pixels on a side (640,000 total), because there is no sure way to determine what color to apply to the pixels that must be added to fill in between the ones actually specified in the original version. Even if the enlargement were a simple doubling or tripling of the number of pixels in each direction, the image must either become more and more blocky and crude or there must be some interpolation of pixel colors to permit enlargement. Similarly, it is difficult to reduce such an image, to remove pixels, via some explicit mathematical equation.

The other approach to images is to work with lines, points, and arcs instead of pixels. Each line, point, or arc can be given a color, a starting point, a length, a direction, and a thickness; such lines are called vectors. Depicting vectors on screen can be done easily – at any scale – via a translation process to get from lines, arcs, and points to pixels that must represent them at the scale required for the screen. There may be scales at which the drawing is displayed poorly; that is not

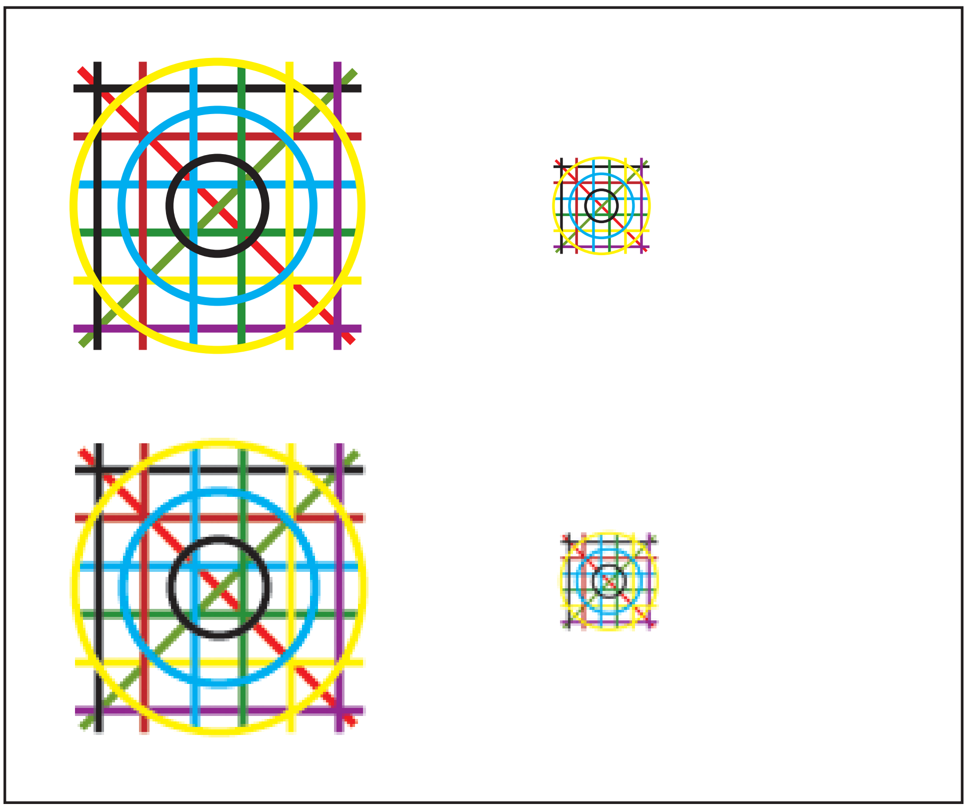


Figure 4

Two versions of the same design. The upper, left version was created with a vector program called Adobe® Illustrator® and reduced to one-third size (right) in that program. The original vector drawing was opened in a raster program called Adobe PhotoShop® (lower left) and, in that raster-based program, reduced by one-third. The differences are hard to see on screen, unless enlarged, but they are apparent on paper. The raster image, when reduced, loses sharpness and color crispness.

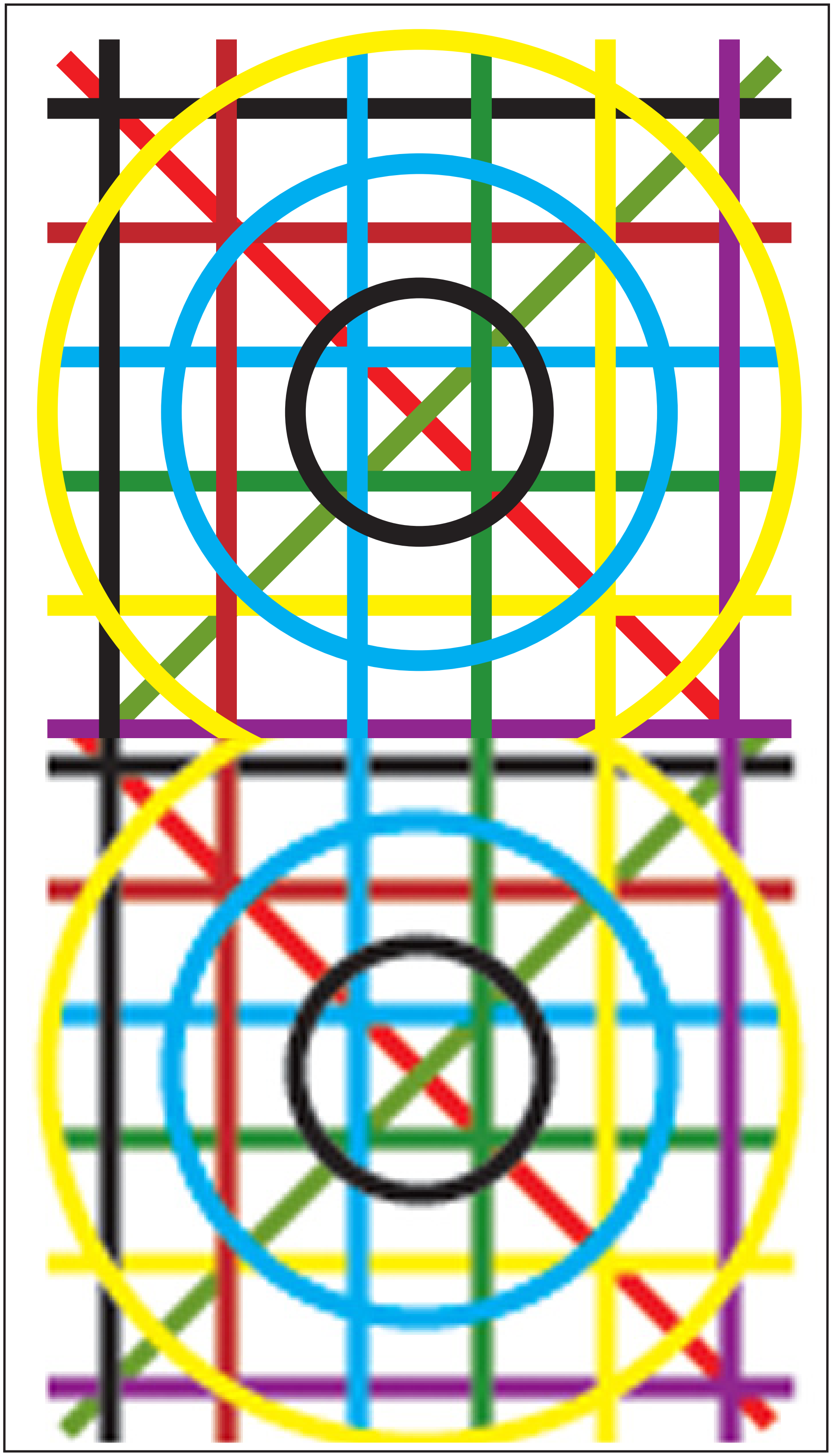


Figure 5

Three-times enlargements of the vector image (above) and the raster image (below), both slightly truncated due to space limitations. Enlargements done within the appropriate programs. The differences are apparent on screen but even more apparent when printed.

uncommon when very complex drawings are displayed at small scale, making the individual portions of the drawing virtually indistinguishable. In general, though, a vector graphic can be displayed at any level of magnification without the need for interpolation and the resulting degradation of the image.

Raster graphics are more common because they are much easier to create, but they cannot be scaled well. Vector graphics can be scaled very well, but creating an image entirely of mathematically defined vectors is not easy. Therefore, both kinds of graphics are widely used and may be expected to remain in common use in the computer world.

Field archaeologists will find themselves using vector graphics – whether they know it or not – with CAD and many GIS packages. They will use raster graphics for images from digital cameras or to display images on the Web (at least until a widely-used web standard for vector graphics emerges). Most vexing, archaeologists using GIS software will almost surely find themselves needing to use both raster images (satellite photos for instance) and vector images (many maps, depending on the sources) – and trying to be sure that they can deal with the information presented in ways that honor the original sources, regardless of the type, as will be discusses at some length in the GIS chapter.

Color, Gray-Scale, and Black-and-White

The vast majority of the computer images used every day are raster images, and they may be color or black-and-white images. In the world of photography, those two alternates were enough. They covered all the bases. In the digital world, though, black-and-white is truly back or white dots, no grays (often called bit-mapped images). So there is a third category of raster images, gray-scale images, which include not only black and white but shades of gray. It is even possible to specify the number of shades of gray.

Sharing Files

Networks

Networks can be set up to permit users to access files on another computer, and support for the necessary protocols to accomplish this is often available via university or corporate computing centers. Archaeologists will inevitably find themselves using some forms of networking in the future simply because the quantity of data available over the Internet will continue to grow, making use of such data an everyday matter. Local, project-oriented networks are also likely, and the expertise of computer-center personnel should be considered a critical resource for such local networks.

CDs, DVDs, and the "Sneaker Net"

It is a simple matter to share files by placing them on CDs or DVDs and mailing them. If users are physically closer, they can simply hand-carry them from place to place using the "sneaker net."

FTP

FTP of File Transfer Protocol is an old and relatively simple way to transfer files from one computer to another over a network; the basic protocol has been updated with a more secure version, predictably called Secure FTP. The computer on which the file is stored must be set up to permit others to reach it for obtaining the files. It can be set up to require a user name and password or to provide files to anonymous users. In either case the files are transferred directly from the host computer to the recipient. Even very large files can be transmitted quickly over good high-speed connections, but using FTP over a dial-up connection (yes, though rare, they still exist) can be quite slow. (Windows, MAC, and Linux computers include an FTP client that permits users to access FTP hosts directly and easily, but many users are not aware of the existence of those programs. Web browsers routinely include routines to permit FTP access as well; users download files on the web via FTP routines included in web browsing software.)

Attachments

Using email to send files from one person to another has become very common. The files to be sent may be "attached" to an email and sent along with the message. Despite the popularity of this method of transferring files, it is not to be recommended as a way to share files. There are several drawbacks. First, the sender, not the person in need of the file, controls the action. Second, many email systems place limits on the sizes of attachments, making it impossible to send large files. (The limits generally exist on both sending and receiving sides, and there are often different limits at each end.) Third, problems with attachments that carry viruses and other malware have made many reluctant to open attachments without checking for authenticity. Indeed, the problems with attachments as transmitters of malicious programs make the use of email attachments undesirable in general.

HAL and the Computer as God

One of the hidden problems of computer use is the common assumption that computers do not make mistakes either in processing data or in retrieving it. In fact, however, computers simply follow rules that have been created by humans. We humans surely do make mistakes. Therefore, the rule-based systems we create will contain errors. If that seems harsh, remember that the typical program is released with many "bugs," slight errors that may not normally cause problems – and those are programs written by highly-paid professional programmers. Over time bugs are usually repaired, but each repair carries with it the risk of causing a new and different problem. Archaeologists writing scripts and macros will make more errors; we can only hope that they will be relatively minor. Even when the computer is performing correctly and the software is working correctly, the processes being followed may not be the correct ones. If there is an erroneous instruction to add two numbers instead of subtracting one from the other, the computer will do as told, and the answer will be correct in the sense that the result of adding 3 and 2 will be five. Nevertheless, the user who wanted the result of subtracting 2 from 3 will not be well served.

If computers are not perfect but are deemed to be so, there is a true problem. Users may then accept incorrect, even silly answers from a computer without a thought – answers they would not accept from a human. As a result, anyone using computers for serious work must make some attempt to check the systems being developed to be sure that they do function as required. Similarly, it is important to try to prevent incorrect data entry, since users may credit an incorrect "fact" simply because it seems to come from the ever-accurate computer, and even a single mistake in data can ripple through a large collection. For instance, a decimal point misplaced by accident when entering a pot's diameter might never be noticed but could easily skew the average (and standard deviation) for the site corpus significantly. (Ten pots with a true average diameter of 12 cm. would show an average of 10.8 cm. if a single pot had been mistakenly entered as having a diameter of .12 instead of 12 – or an average of 22.8 cm. if one diameter were mistakenly entered as 120 instead of 12.) Since the problem would result from an error involving a single pot, it might never be found.

Network Collaboration

The term network today suggests the Internet to many people. However, there are many smaller networks, as small as a few computers connected to one another in an office. Such small networks, called local area networks (LANs), permit users to connect to multiple computers or printers from their own computers. This may require a network operating system, though many LANs operate with the simpler connection possibilities built into Windows, Linux, or the MAC OS.

Using a network makes it possible for many people to work on a single file, with the file stored in only one place on the network, not on each person's computer. Of course, working cooperatively on a single file requires special procedures to prevent two people from making changes simultaneously that either contradict one another or cancel one another out. Nevertheless, having a single file serve many users is a far more economical and efficient way to operate. In addition, as noted above, using a network computer to store the important large files from a project makes it unnecessary to have very large hard disk drives on many project computers.

Larger networks, such as those on a college campus or a large company, called wide area networks (WANs), connect multiple LANs to one another and permit sharing of information over a much larger number of computers and a larger geographic area. WANs may even use the Internet to connect LANs together. The typical WAN is controlled by an individual person, company, or institution so that all the parts work together properly.

In some ways, the Internet is simply a huge WAN. However, there is no single person, corporate entity or institution that controls all, or even most, of the pieces of the network. Instead, the Internet is actually just the wiring, wireless links, and switching boxes that allow those connected thereto to communicate – and the agreed procedures that permit interaction between and among them. Each connected computer operates according to a sub-set of the protocols that permit specific kinds of cooperation. The key is that all follow a set of rules and protocols that permit them to work together predictably.

As use of the Internet has grown, some archaeologists have become interested in collaborating over the Internet on large data sets. This seems likely to become more and more common as archaeological projects involve more people at more institutions and as data sets become larger and larger. This is not the place to discuss the technical aspects of such collaborative systems, but the use of the Internet for collaboration does place a very high premium on the use of standards for data storage formats, for access procedures, and for data protection – and common protocols for data transfer, security, and authentication. There will be many references throughout this book to the importance of standards and protocols; collaboration over the Internet requires attention to those issues. The need for common vocabularies is equally important here, but that is not, of course, a computer issue.

Copyright, File Integrity, and Version Control

Copyright often seems the enemy of scholarly information sharing, and the use of copyright protection has a bad reputation among the computer cognoscenti. Nevertheless, copyright protection can provide critical benefits for the maintenance of scholarly information. Asserting copyright over scholarly data provides a legal framework for controlling any changes to the data. While that can be viewed as elitist and controlling, someone must be responsible for making sure that archaeological data have been properly guarded, that additions have been done properly, changes have been made correctly, removal of data has not damaged referential integrity, and so on. The foregoing clearly implies that the holder of the copyright has the keys to the kingdom; the copyright holder can prevent additions or other changes, and that can freeze out a non-standard view of any given data set. However, that is preferable to letting any and all changes be made without any control at all. Someone who believes there to be errors, after all, can write freely about those beliefs and explain why his/her changes should be made. And anyone should be able to change the data set on his/her own computer for his/her own purposes. But there must be an "official" version of the data set, one that has been checked and that carries a kind of stamp of approval, one that provides the starting point for anyone studying the material. That should be the only data set available for circulation, and copyright permits that kind of control of the data set.

Copyrighting a data set makes explicit the copyright holder's responsibility for the integrity of that material. Asserting a copyright and explicitly providing free use of copyrighted material for non-commercial purposes also clarifies matters for colleagues and, at the same time, limits the commercial use of copyrighted materials. Whenever data files can be shared, in fact, they should ideally be shared via an archival repository with the copyright terms defined carefully so that anyone gaining access to the information always gets it from the source responsible for keeping it in good order (the archival repository). The copyright should be shared with (and eventually owned exclusively by) the repository, because the repository will have the personnel and the longevity to enforce the terms of the copyright statement over time – and the responsibility for keeping the files in good order. The archaeologists, on the other hand, will not want to spend their time with such issues, and they may survive the project by a relatively short time.

In order to get files into condition to be shared – and to be sure that the files being shared are the proper ones – there must be a commitment from the outset of a project to control effectively the files that are in use, in storage for back-up, off-site for disaster protection, and so on. A system for making clear which files are current, which may have been compromised in some way, which have been superseded, and so on is required. Otherwise, there is a considerable risk that edits will be made to the wrong file or that a current file will be over-written. This is often called version control, since the issue is to be sure which version of a file is in use at any moment.

Version control will be especially difficult in collaborative projects involving the Internet. Data entry and editing procedures over the Internet generally involve intermediary data storage and periodic updating; a systematic approach to file storage, editing, and copying is essential. In smaller projects, it is no less important to control the file versions, but it should be less difficult.

Conclusion

Adding computers to an archaeological project for data storage and retrieval adds more than new tools. As should be obvious from the foregoing, a number of basic issues, some of them truly critical to the way data are stored for the long term, must be understood to be sure that the data are properly recorded and properly preserved. Not every scholar needs a full understanding of the differences between raster and vector images or the problems of ASCII and Unicode. However, someone in a position of authority on every archaeological project must be aware of these kinds of arcane issues and attentive to them.

When an archaeological excavation or survey season ends or the project is concluded and the staff goes home, good practice requires that the finds and paper records be properly stored, the site properly back-filled or protected, and the scientific samples properly cared for. No less can be asked of the data about the finds, the site, and the samples. Those responsible for the computer files must be as diligent as those responsible for the objects and the site itself. This is true at the end of a season, at the end of a project, and when the data are handed off to an archive for storage in perpetuity. The issues discussed here are critical to this work.

III

Recording Data about Objects, Loci, Trenches, Features, . . . from Archaeological Projects: Databases and Database Management Systems

Glossary

4th Dimension®: a Windows and MAC database management system.
Access®: Microsoft's database management system for Windows; sold as a part of the MS Office® suite.
Attribute: in databases, a specific characteristic of an item; see column.
Atomize: break data into its smallest constituents, e.g., using the categories *street address*, *city*, *state*, and *zip*, not just one large category called *address* with all the pieces of the address included.
Cell: an individual field or column entry for a single row (a term more often used when discussing spreadsheets than databases).
Child table: a table related to another in a many-to-one (child-to-parent) relationship.
Codd, E. F.: an IBM scientist who first set out formal rules for so-called normalization of databases, rules intended to prevent errors from entering the data and to prohibit data duplication.
Column (field)**:** the particular attribute stored about each unique item in a table.
dBase®: the first popular database application for PCs.
Field: equivalent to column.
File: a single digital document; in database usage, a table may be stored in digital format as a single data file. Modern databases often store many tables in a single, complex computer file.
FileMaker®: A database management system made for Windows and MACs by a subsidiary of Apple.
Flat file (database): a database table that is complete on its own and needs no other tables to make it useful.
Foreign key: a column (field) in one table that is used to identify a related row in another table.
FoxPro®: a database management system based on some dBase standards and widely used in many areas because of its flexibility (and its longevity). This is a Microsoft product, and Microsoft has announced that it will not be updated in the future. User support will continue until 2015.
Join: the basic term used to describe combining the content of two related data tables. There are several kinds of table joins.
Key field: a column (field) that is indexed.
Lookup table: a table used to provide a limited selection of choices with which to fill a column (field) in another table.
Many-to-many relationship: the relationship of a column (field) in many rows (records) of one table to a column (field) in many rows (records) of another table. This complex relationship is the most difficult table-to-table relationship encountered in a relational database management system.
Many-to-one relationship: the relationship of a column (field) in many rows (records) of one table to a column (field) in a single row (record) of another table. A table with information about many children has a many-to-one relationship to a table containing information about the women who are the mothers of all those children. (Note that some of those women may have no children; others one. But the nature of the connection permits an undefined and unlimited number of children to be linked to any woman.)
MySQL: the most commonly used database management system with Linux and the most oft-used for web sites. MySQL, the company, has been bought by Sun Microsystems; the software remains free, well-regarded, and popular.
Normal(ize): organize data in tables to avoid duplication and to make sure that queries create correct responses. There are very specific requirements for various levels of normalization.
Object-oriented database: a form of database design that is informed by a hierarchical organizational scheme such that any item can be described, at least in part, by the characteristics of those above it in the hierarchy. This is not an approach recommended by the author, and it will not be discussed here.

One-to-many relationship: the relationship of a column (field) from any given row (record) in one table to a column (field) in many rows (records) of another table. The reverse of the many-to-one relationship, this relationship would relate a table with information about mothers to the table of all their children.

One-to-one relationship: the relationship of a column (field) from any given row (record) in one table to a column (field) in a single row (record) of another table. This rather simple relationship would relate two tables with information about the same subject or two tables with information about different subjects, but neither table having multiple entries about the object in question.

Oracle®: a database management system often used in very large corporate settings. It is not recommended for individuals without training in its use and ready access to good technical support for the life of the project.

Parent table: a table related to another in a one-to-many (parent-to-child) relationship.

PostGreSQL®: a widely used database management system for Linux.

Primary key: an indexed column (field) that is unambiguously the subject of the attributes in the row (record) and that signifies one and only one row (record) in the table; to be unambiguous the content of the column (field) must be unique. It is possible to construct a primary key from two or more columns (fields) in the table.

Record: equivalent to row.

Relational database: a set of related data tables, each of which depends on others in some way for completeness.

Repeating field: a column (field) that has multiple independent values as attributes for the item in the row (record).

Row (record)**:** the collection of attributes of a particular, unique item in a table.

SQL (Structured Query Language): the defined standard syntax for accessing data from tables.

Spreadsheet: a digital file very similar to a digital data table but with the possibility for including specialized formulae in the table as well as data. The term is also used to denote the kinds of programs that create spreadsheets.

Table: a collection of information organized into rows of data, each row concerning a unique item, and columns, each column containing a specific attribute of the items.

Unique column (field): any column (field) for which each data entry must be unique.

Introduction

Databases are most simply and directly defined as tables in computer form. Thus, the typical morning newspaper, with its list of weather conditions in various cities around the country or the world, is presenting database-style information. Indeed, that simple table is an excellent example to use for the beginning of this discussion. Consider this hypothetical example of such a table showing weather conditions and temperatures (Celsius) for cities around the globe:

City	weather for today			tomorrow		
	Hi	Lo		Hi	Lo	
Amsterdam	24	18	PC	26	20	PS
Cairo	40	34	S	41	33	S
Jerusalem	29	25	PS	30	26	PS
Damascus	38	30	S	39	30	S
Madrid	34	25	PC	32	23	SS
Athens	39	28	S	38	28	S
Paris	29	20	PC	28	23	PC
London	27	20	Sh	26	20	R
Berlin	26	20	PC	27	20	PC
Washington	36	30	S	37	30	PC
Ottawa	30	24	PS	29	23	PC
Ankara	35	28	PC	34	27	PC

Rows

Most important column, in the sense that the data in all the others are about it.

Columns

In this table the rows (horizontal lines) show the city to which the remaining information applies: high and low temperatures, and general conditions for that city on each of two days. The columns show the cities, temperatures, and so on. In each row the city is the key item; the other items are attributes of the cities.

Note that this table is not in any particular order. One of the simplest advantages of using a database to build such a table is that changing the order is trivial. Ordering the cities alphabetically requires only a simple request.

City	weather for today			tomorrow		
	Hi	Lo		Hi	Lo	
Amsterdam	24	18	PC	26	20	PS
Ankara	35	28	PC	34	27	PC
Athens	39	28	S	38	28	S
Berlin	26	20	PC	27	20	PC
Cairo	40	34	S	41	33	S
Damascus	38	30	S	39	30	S
Jerusalem	29	25	PS	30	26	PS
London	27	20	Sh	26	20	R
Madrid	34	25	PC	32	23	PC
Ottawa	30	24	PS	29	23	PC
Paris	29	20	PC	28	23	PC
Washington	36	30	S	37	30	PC

What's more, the information, here presented with temperature in Celsius, can be presented in Fahrenheit (calculated and rounded to the nearest degree) with another simple request:

City	weather for today				tomorrow		
	Hi	Lo			Hi	Lo	
Amsterdam	75	64	PC		79	68	PS
Ankara	95	82	PC		93	81	PC
Athens	102	82	S		100	82	S
Berlin	79	68	PC		81	68	PC
Cairo	104	93	S		106	91	S
Damascus	100	86	S		102	86	S
Jerusalem	84	77	PS		86	79	PS
London	81	68	Sh		79	68	R
Madrid	93	77	PC		90	73	SS
Ottawa	86	75	PS		84	73	PC
Paris	84	68	PC		82	73	PC
Washington	97	86	S		99	86	PC

The same could be done, just as easily with the abbreviations for weather conditions, which are in English in the example (S=sunny, PS=partly sunny, PC=partly cloudy; Sh=showers, R=rain) but might be wanted in appropriate abbreviations for any number of other languages.

These re-formulations of the weather information show only the simplest of the things possible with databases, but this is not a bad example to illustrate some broader uses. For instance, consider the long-term value of weather information. It is desirable to maintain records of daily highs and lows, and the same database that churns out the information for a newspaper can keep that information indefinitely to permit comparisons, examination of long-term trends, and the like.

A database such as the weather database we have been using as an example is likely to be maintained by some agency for distribution to many users. Thus, the selection of information shown above might be used in an international paper, but a paper intended for a local audience might want a very different selection of cities for its readers, and the database would be constructed to permit many different sub-selections, a subject to which we will return.

Many readers will be familiar with business spreadsheets and the extremely popular spreadsheet program, Excel®. This weather database is not significantly different from a spreadsheet with the same information. Indeed, as we have presented the information thus far, a spreadsheet could match the presentations of the database. In each case, database and spreadsheet, a city is connected to attributes – high and low temperatures and general weather conditions – on specific dates. Each city has the same attributes for every date. Such a simple table – one that can stand on its own and needs no external information to assist – is often called a flat file or flat file database; a spreadsheet can hold the same information as a flat-file database.

Database Organization

This simple database – and any other database – can be compiled using a number of database management systems (DBMS). The most widely used ones in the Windows world are Access (from Microsoft), FoxPro (also a Microsoft product now), and FileMaker (from FileMaker, a subsidiary of Apple). For MAC users FileMaker and 4th Dimension (also available for Windows) are popular choices. Users of Linux are most likely to work with MySQL or PostGreSQL. For each program the processes of creating a data table are different; so there will be no attempt to describe those processes here; nor will there be recommendations of software. Instead, the intent here is to discuss the ways all databases should be structured and organized so that the complexity of the data can be properly reflected in the organization of the database. Any modern DBMS can be used to accomplish this level of organization.

Before continuing we need to settle on some database terms. There are two sets of terms regularly used to refer to the components of databases. One set includes *file*, *record*, and *field*; the other *table*, *row*, and *column*. The table is, of course, the individual collection of columns and rows that contain tabular information. Since databases initially stored any table as a single file, *file* was equivalent to *table* in early usage. Similarly, *row* and *record* are equivalent, as are *column* and *field*. The terms were used together in specific ways, though; a record is a discrete part of a computer file in many storage formats, and a field was defined as a discrete portion of a record, defined in various ways by different software. Thus, *file*, *record*, and *field* all depend on specific computer-storage terms. *Table*, *row*, and *column* are all related to the appearance of a table instead. As databases have become more complex – in ways to be discussed below – the equivalence between file and table has disappeared, with files becoming more complex and capable of containing many tables; therefore, we will use the *table*, *row*, *column* terminology from here on. A table is normally named by the user as any computer file might be, using a name that is meaningful to the user. The row is not named but it can be identified by one term that is its subject, the city in our weather table. Columns are normally given attribute names that are often cryptic, e.g., hitoday, lotoday, condtoday.

The term *key* will also be used; a key is simply a column that has been indexed. More important conceptually, one key must unambiguously label the item that the row describes, in this case, the city. The other information in the row is, after all, about the city. This is the primary key. The primary key entry must be unique to function, to identify the subject of the row unambiguously. (But imagine a table about U.S. weather that must provide weather conditions for Springfield, IL, and Springfield, MA – and perhaps other Springfields. The primary key would need to be at least a combination of city and state to be unique. In fact, one might need the name of the county as well, since some states have more than one Springfield.)

Indexing

A table may be ordered according to the content of any column; it is then said to be indexed *on* the column. The rows are not actually put into order in the table; rather, the DBMS software creates multiple indices (transparently) so that access to any given row, using any of the indices, can be speedy. An index may be a separate file maintained by the DBMS, but its function is simply to help speed access by providing quick ways to find rows in sequence according to any of the available indices.

One more term is often used – especially with databases used in connection with GIS software – *attribute*. An attribute is, as in standard English usage, simply a characteristic of an object; it is equivalent to a field or column. Because the term has its standard English usage, it will be avoided in this chapter where *column* could be used, but it will be used regularly in the GIS chapter.

In the database above – the weather table – the city, the primary key, would be the only key, since indexing the other columns would be pointless. The other columns provide attributes of the cities: high and low temperatures and weather conditions (for each of two days). It is often the case that there are no keys save the primary key.

Structured Query Language (SQL) – The Database Standard

The most widely used system to access information from data tables is a common approach that many different DBMSs permit; indeed, many use it exclusively. This is the Structured Query Language or SQL. Statements composed using the Structured Query Language follow proscribed syntactic rules that have database organization at their cores. Sooner or later, anyone using a database should become familiar with the syntax of SQL, not only because it can be used to access tables in many different formats but also because it is the *lingua franca* of databases. Manuals will often explain processes by reference to SQL syntax, and SQL statements will be used here to show how data might be extracted from a table or tables. Indeed, some DBMSs generate SQL statements to carry out user-created commands, and it is helpful to be able to parse and edit such statements. (In the examples here columns are often named with abbreviations and more than one word for clarity. However, because of SQL syntactic requirements, real column names should have neither spaces nor punctuation marks. All sample SQL statements here will use column names without spaces or punctuation marks.)

The most common way to extract data from a table using SQL is to use a *Select* statement such as

SELECT hitoday, lotoday, condtoday FROM weather WHERE city = 'Amsterdam'

This would return the high and low temperatures as well as the weather conditions for today only for Amsterdam. The syntax is straight-forward: SELECT (choose) hitoday, lotoday, condtoday (columns) FROM (from) weather (the specified table) WHERE (if) city = 'Amsterdam' (these conditions are met). The result is a set of data from the table. The data may be treated as a separate table and either stored with its own name or accessed directly without storing it first. (Upper-case commands are for clarity only.)

A Database for a Museum Exhibit

Now let us start to construct a data table to contain information about objects in a museum exhibit. This will be both a more archaeological example and one that leads to mounting complexity. The exhibit is to be in a small museum with a good but very small collection of ceramics of the Moche culture. In addition to their own objects, the museum will borrow enough from other institutions to have an exhibit larger than possible without the loans.

A database is to be constructed for information about the individual pots. The intent is to prepare a base from which information (text and/or dimensions) for an illustrated catalog as well as other exhibit-related materials can be extracted. The database will also be used to track the security/location of the objects as they are moved from their normal positions or home institutions to our museum and then to the display cases. For each pot the following information must be recorded (not necessarily a complete list of what a museum might want but a representative sample of information): home institution, inventory number, catalog number, shape name, period, beginning date for the period, ending date for the period, dimensions (two attributes: height and diameter), a short description of the pot, bibliographic sources, comparanda, original storage location, current location, case number, and position in the case. An object number has also been added to serve as the primary key. It might seem that the catalog number would serve as the primary key, but, should the arrangement of the catalog change, that number could also change. The object number, on the other hand, exists only to provide an unchanging and unambiguous identification number for every object; with that as the primary key, the catalog number is just another attribute of the object as it should be. In addition, the date of last change to the data in each row is indicated, along with the person responsible for the last update. Such information is too often omitted from data tables, though it can be very important for users.

These 19 attributes (counting the two dimensions as two attributes) are the columns of the data table, and there will be one row per pot. Certain of the data items will be used for display labels and/or the publication to be prepared for the exhibit; for instance, the description will become the label/catalog description. Therefore, entries should be in prose appropriate for all anticipated purposes.

All of our examples came from the same institution, and none has arrived yet;

Moche Pottery Data Table - Left

Obj. No.	Cat. No.	Home institution	Inventory No.	Period	Shape	Beg. Date	End Date	Ht. (cm.)	Dia. (cm.)	Description
1	77	Mus. Arq.	XST-004-001	Moche IV	man with tunic	450	550	26	16.5	A man, either wearing a cap or bare-headed (in that case, with head shaved on top) holds a tunic in front of himself. His face is decorated. . .
2	69	Mus. Arq.	XXC-000-001	Moche IV	Portrait head with disfigured face	450	550	28.3	18.8	Portrait head with disfiguration likely representing the result of a tropical disease; the appearance resembles that of a mummy. A male . . .
3	78	Mus. Arq.	071-002-003	Moche IV	Man with feline	450	550	23.8	14.5	A naked man with his left eye closed (possibly blind in that eye) is seated cross-legged with his hands clasped in front of himself, possibly . . .
4	33	Mus. Arq.	089-004-001	Moche II	Jaguar	100	200	18	18.3	This crouching or lying jaguar looks straight, ahead with open mouth and tongue protruding. This appears to be a cub, but since jaguars are . . .
5	31	Mus. Arq.	091-003-005	Moche I	Dog	50	100	18	9	Sitting erect, tail curling up behind him, this spotted dog shows carefully-observed features on the head and snout. Canine teeth are clearly . .
6	76	Mus. Arq.	XSC-007-004	Moche III	Man with lime gourd	200	450	17.5	14.7	The man is seated, with toes sticking out beneath his tunic, holding a lime gourd in both hands. His face is inlaid with turquoise or chrysacolla for his . . .
7	79	Mus. Arq.	084-004-005	Moche IV	Fineline coca rite	450	550	27.3	15.5	There are two scenes on this pot. On one side a double-headed serpent is shown creating a kind of cave for two men seated within. Each has a lime gourd, and . . .

Figure 1 - Left

The Left half of the Moche *Pots* Table. Many of the terms are abbreviated or cut off in this example of the data table. Abbreviations would not be used if this were a table prepared in a museum because the data might be used in various settings requiring full terms. A database technician could easily add routines to make the output look as desired, but requiring a technician to deal with output is counter-productive. The abbreviations are needed here so the information will fit on two printed pages, but they should not be used in a good database. By the same token, the use of “curator” as an entry term for the person who updated the information is a convenience here. The entry should name a person unambiguously to be sure the information will be useful years later.

Moche Pottery Data Table - Right

Bibliog. Sources	Comparanda	Orig. Loc.	Curr. Loc.	Case No.	Case Pos.	Last update	By
Lavalle 1989	Lavalle 1989: Pls. 159, 160 Cat. Nos. 76, 78, 79	Home inst.	transit	3	middle shelf, center-right	1/1/01	curator
Hoyle 1939: fig. 214; Wasserman-San Blas 1938; Ubbelohde-Doering 1952	Wasserman-San Blas 1938: Pl. 162; Ubbelohde-Doering 1952: Pl. 213	Home inst.	transit	3	lower shelf, center	1/1/01	curator
Benson 1974	Cat. Nos. 76, 77, 79	Home inst.	transit	3	upper shelf, far right	1/1/01	curator
	Cat. Nos. 78, 82, 84	Home inst.	transit	2	bottom shelf, far right	1/1/01	curator
Holye 1948; Alva &Donnan 1993: 123, 159-161; Donnan & Mackey 1978: 144	Donnan 1978: figs. 239, 240, 270; Cat. Nos. 84, 106, 107	Home inst.	transit	1	middle shelf, center left	1/1/01	curator
	Cat. Nos. 77, 78, 79, 96, 97	Home inst.	transit	3	bottom shelf, center left	1/1/01	curator
	Alva & Donnan 1993: 60-67; Cat. Nos. 76, 77, 78	Home inst.	transit	2	middle shelf, far left	1/1/01	curator

Figure 1 - Right

The Right half of the Moche *Pots* Table. Many of the terms are abbreviated or cut off in this example of the data table. Abbreviations would not be used if this were a table prepared in a museum because the data might be used in various settings requiring full terms. A database technician could easily add routines to make the output look as desired, but requiring a technician to deal with output is counter-productive. The abbreviations are needed here so the information will fit on two printed pages, but they should not be used in a good database. By the same token, the use of “curator” as an entry term for the person who updated the information is a convenience here. The entry should name a person unambiguously to be sure the information will be useful years later.

so the original location is marked "Home inst." The current location is marked "transit."

In this formulation the object number is the primary key, the unique identification number for each pot. (It is at least theoretically possible that the inventory numbers from two different institutions might have been the same for two different pots; so inventory number cannot be safely used. Inventory number and institution together could have been used since it is possible to combine multiple columns as a primary key and require only that the combination of columns be unique. While that is possible, it adds complexity in most instances without providing many practical benefits. In fact, adding a simple sequence number to each row in a table, as has been done with these pots to make an object number, is often the best and easiest way to be sure there is a primary key, though many database purists would object, as will be discussed below.) For this table, the sequence number is the object number, as shown in the schema, figures 2 and 3. Note that having an object number separate from the catalog number makes it possible to change the position of the object in the catalog without changing the primary key.

Each pot has the same categories of information. At first glance, this does not look much more complicated than the temperature table shown already. There are many more columns, but the notion is the same. For each item there are attributes. This table could also be ordered according to the content of any of the columns, and specific items could easily be selected as well, for instance, all pots of a given period, from the same institution, of the same shape, or even with a diameter between 16 and 18 cm. In fact, this table is equivalent to a complete so-called flat-file database, and the same data could be stored in a spreadsheet.

There is one serious complication in the Moche *Pots* Table. Both bibliographic sources and comparanda present this common problem – multiple entries of attributes for many individual pots. That is, there may be a single comparandum or two, three, four, or . . . In fact, there is no way to know in advance how many comparanda an individual pot may have. Similarly, there may be a single bibliographic source, but there may also be more than one for a given pot, and it is not possible to know in advance how many bibliographic sources there will be.

How does a database designer deal with the possibility of an unknown number of entries for a given column? There are four possibilities:

1. Construct the table as illus-

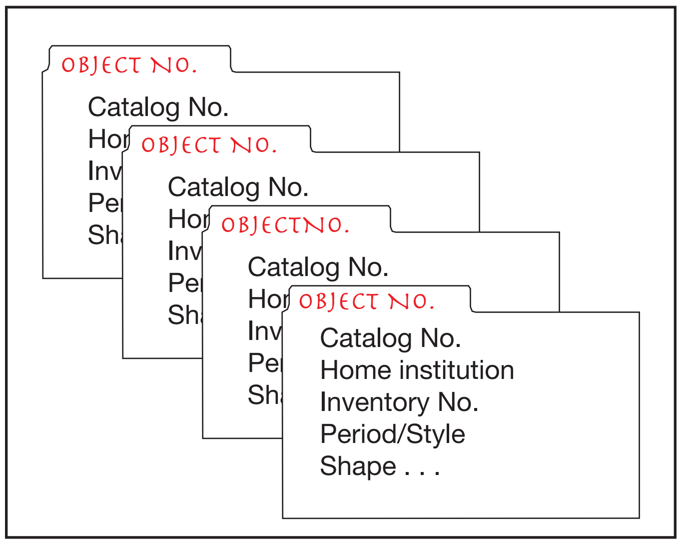


Figure 2
Another way to think of a data table – as a card catalog, with the primary key in the tab and other information on the body of the card; one item (row) per card. The illustration shows four cards (with data categories, not actual data). This is an effective way to conceive of the information in a data table because it permits tables to be shown with links to one another, as will be seen.

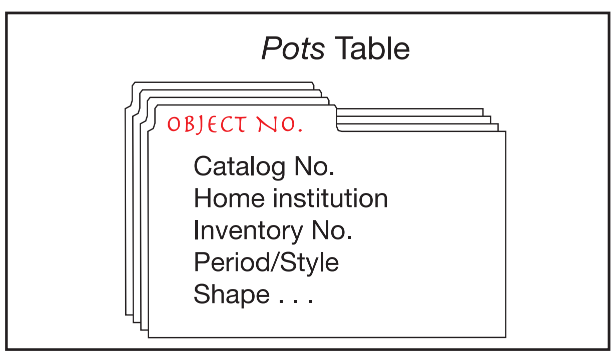


Figure 3
The card catalog view of the *Pots* data table compressed for simplification. This view will be used in coming illustrations; so please be sure you are comfortable with this presentation before moving on.

trated, with multiple entries crammed together into a single column.

2. Use a best guess as to the maximum number of comparanda and bibliographic sources and put that many columns into the base record – that is, in this example, three or four comparandum columns and two or three bibliographic sources columns.

3. Use a feature of some database programs called repeating columns to enter each item separately but, for all intents and purposes, in the same column.

4. Create a separate table for comparanda and another separate table for bibliographic sources, linking each to the base table.

Neither of the first two alternatives is a good solution. If the bibliographic sources column is not limited to a single entry (alternative 1), how does one create a simple list of the sources? The references in any given row are not separate; so they can't readily be shown individually. The same is true for the comparanda. This procedure would yield a partial list of bibliographic sources like this:

Lavalle 1989
Hoyle 1939: fig. 214; Wasserman-San Blas 1938; Ubbelohde-Doering 1952
Benson 1974
Holye 1948; Alva & Donnan: 1993: 123, 159-161; Donnan & Mackey 1978: 144

Similarly, a partial list of comparanda would appear as:

Lavalle 1989: Pls. 159, 160; Cat. Nos. 76, 78, 79
Wasserman-San Blas 1938: Pl. 162; Ubbelohde-Doering 1952: Pl. 213
Cat. Nos. 76, 77, 79
Cat. Nos. 78, 82, 84
Donnan 1978: figs. 239, 240, 270; Cat. Nos. 84, 106, 107
Cat. Nos. 77, 78, 79, 96, 97
Alva & Donnan 1993: 60-67; Cat. Nos. 76, 77, 78

Putting all the entries in a single column may thus be acceptable for the catalog and the exhibit labels (where some re-formatting might help) but not for long-term, scholarly use of the database. If the database were used by a scholar, lists of bibliographic resources or comparanda would be required – in readable formats showing each comparandum or resource individually.

The second alternative, using many columns, one for each comparandum and bibliographic source, would seem an obvious solution; however, it complicates any searching. Each of the comparanda columns would have to be searched independently to find a specific comparandum, as would each bibliographic source column for a given reference. Displaying the table would also be difficult, and there would be many empty columns because each row would need the maximum predicted number of columns for sources and comparanda, regardless of the number actually filled with data. This is also inefficient; much of the storage space would be empty. (In fairness, it must be said that modern DBMS systems compress their data files to eliminate that problem. In addition, storage space has become a very minor issue due to plummeting prices and expanding capacity of hard disks.) Of course, this method is also poor because it requires an advance determination of the maximum number of comparanda and bibliographic sources. If, in the course of data entry, a single pot happens to have more comparanda or bibliographic sources than planned, the table design will have to be altered or the unexpected information omitted.

The other two methods for dealing with this problem are more efficient; both provide for displaying comparanda and scholarly sources individually – as many as required in both cases. Both comparanda and sources can be displayed individually in lists related to the individual pots, and all can be listed for inspection. Both these alternatives also permit searching for a specified comparandum or source with a single SQL SELECT statement or a single process within the database itself. Most important is the fact that both methods permit as many entries as required without knowing in advance how many there will be at the end of the day.

The first of these methods is the use of repeating columns. As the name implies, a single column can have multiple entries in such a case, and each is treated as a single, separate entry. Not all database programs make this possible, but some do provide for repeating columns to deal with just these kinds of situations. (FileMaker is the only widely used database management system with this feature known to the author; with each new release, I check to see if the feature has been removed.) An individual column is simply defined as a repeating column when the database is constructed, and multiple entries are then permitted – with no limit on the number.

For the vast majority of database programs that do not permit repeating columns, a similar solution is to create a separate table for any column that needs multiple entries. That second table can be constructed with just two columns. For comparanda, one column would contain the object number (the primary key in the *Pots* Table), and the other would contain an individual comparandum. In this *Comparanda* Table, the primary key from the *Pots* Table would be used as a foreign key, since it refers to a unique column in another table. The foreign key must refer to a unique key in the other table to make the linkage unambiguous.[1] (There should also be a third column, a sequence number that would provide a simple primary key for this table.) This solution is very similar to the use of repeating columns; printouts and screen displays can be designed to show the information appropriately, and there can be as many entries as needed – without any advance limit or prediction of the number of entries. One SQL SELECT statement can be used to find any specified comparandum. A similar solution works for reference sources. Again three columns are required, one for the object number (the foreign key here), one for the actual source, and a third column for a sequence number to be used as the primary key. The two extra tables – a *Sources* Table and a *Comparanda* Table – would contain these data items.

***Sources* Table:**

Object No.	Source No.	Source
1	1	Lavalle 1989
2	2	Hoyle 1939: fig. 214
2	3	Wasserman-San Blas 1938
2	4	Ubbelohde-Doering 1952
3	5	Benson 1974
5	6	Holye 1948
5	7	Alva & Donnan 1993: 123
5	8	Alva & Donnan 1993: 159-161
5	9	Donnan & Mackey 1978: 144

[1] The term *unambiguous* will be found often in this chapter. Ambiguity is the enemy of good database design and the literal nature of computers referred to earlier makes it critical that ambiguity be eliminated at the database design level. Unfortunately, any computer or database program will happily tolerate ambiguity – and simply provide inaccurate search results as a consequence. It is the database designer's job to prevent inaccurate search results by, among other things, preventing ambiguity.

Comparanda **Table:**

Object No.	Comp. No.	Comparandum
1	1	Lavalle 1989: Pl. 159
1	2	Lavalle 1989: Pl. 160
1	3	Cat. No. 76
1	4	Cat. No. 78
1	5	Cat. No. 79
2	6	Wasserman-San Blas 1938: Pl. 162
2	7	Ubbelohde-Doering 1952: Pl. 213
3	8	Cat. No. 76
3	9	Cat. No. 77
3	10	Cat. No. 79
4	11	Cat. No. 78
4	12	Cat. No. 82
4	13	Cat. No. 84
5	14	Donnan 1978: fig. 239
5	15	Donnan 1978: fig. 240
5	16	Donnan 1978: fig. 270
5	17	Cat. No. 84
5	18	Cat. No. 106
5	19	Cat. No. 107
6	20	Cat. No. 77
6	21	Cat. No. 78
6	22	Cat. No. 79
6	23	Cat. No. 96
7	24	Cat. No. 97
7	25	Alva & Donnan 1993: 60-67
7	26	Cat. No. 76
7	27	Cat. No. 77
7	28	Cat. No. 78

Each source is separate, which may seem to be unnecessary, but it is a better approach to the data since each item is treated individually and can be accessed individually. For instance, there are two different references in Alva & Donnan 1993 for object number 5 (each referring to different page numbers). They are treated separately in the *Sources* Table, as they should be.

Each table has a column for the object number so that the sources and comparanda can be related to the proper pot. (The foreign key does not have to refer to the primary key in the related table, though it normally does. The foreign key does need to refer to a unique column, as noted above.) The result of this process is related tables – one, the *Pots* Table with various attributes of each pot, and the other two, the *Comparanda* Table and the *Sources* Table, related to the *Pots* Table so as to supply, when required, multiple items for certain attributes of individual pots. Such relationships are called one-to-many relationships because the *Pots* Table has a single item or row to which multiple items or rows in the secondary tables are linked. The *Pots* Table (one) in this arrangement is referred to as the parent table; it has the primary key. The secondary tables (many) are called child tables; they contain the foreign keys. Figure 4 shows the new arrangement schematically; note that there is a sequence number for the *Comparanda* Table (Comp. No.) and another for the *Sources* Table (Source No.) to serve as the primary keys for these two new tables.

Note how this arrangement of parent-child tables makes the use of the sequence number as the primary key in the parent table even more sensible. The information in the child tables is, after all, about the object, not a catalog number.

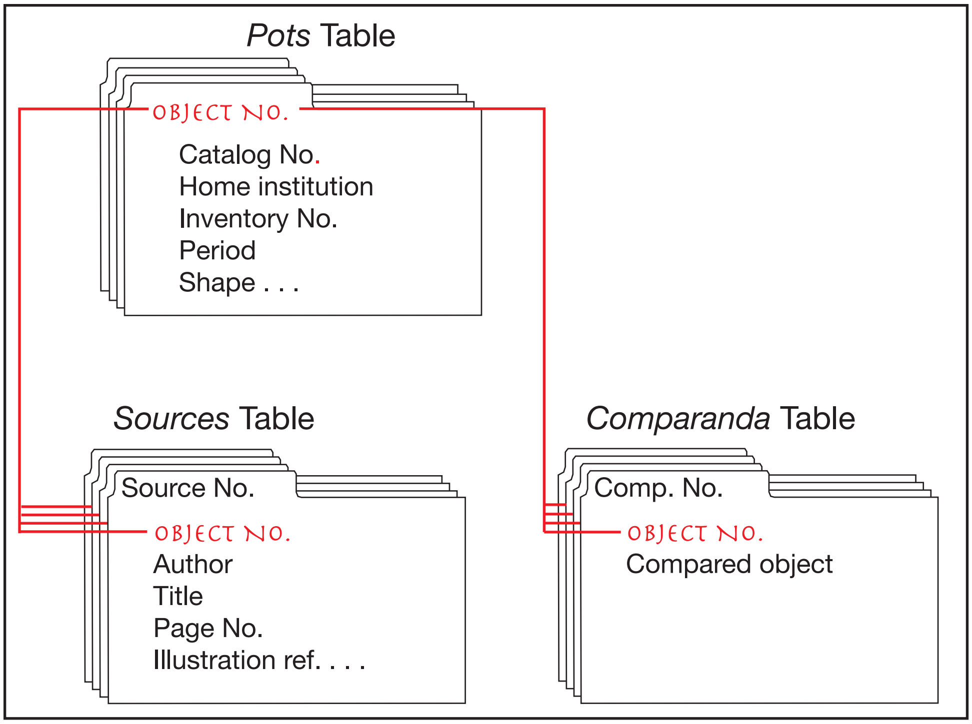


Figure 4

New configuration of Moche pottery database with parent table (*Pots* Table) and child tables (*Comparanda* and Sources), connected via the use of the primary key from the *Pots* Table, Object No., in each child table. The one-to-many relationship is indicated by a single connection from the parent table's primary key leading to potentially multiple connections to the foreign key in the child tables.

There are two crucial differences between the use of repeating columns and the use of a child table. First, repeating columns are possible in few database programs (partly because they violate some theoretical rules of database design), but child tables can be used in all relational database management systems, including any that permit repeating columns. Second, tables with repeating columns cannot be exported into a common, non-proprietary format; for export to another database system, any table with a repeating column must be broken into two separate tables, one with everything but the repeating column data and the other with the repeating column data only (and a foreign key); in other words, separate tables must be created. Separate tables, on the other hand, can be exported without problems. Because of these problems, repeating columns should NEVER be used in a data table intended for more than personal use; the universally applicable approach of parent-child tables should be used exclusively.

Parent-child or one-to-many table relationships depend on the primary-key-to-foreign key link. Data in the primary key or another unique column in the parent table and also used in the foreign key in the child table links that row in the child table to the only row in the parent table with the same data. The foreign key need not be unique in the child table, allowing any number of rows in that table to link to the parent.

Note the possibilities for selecting data from related tables. It is possible even to locate information in one table according to criteria in another. For instance, we could select all pots with "Donnan 1978: fig. 239" as a comparandum or locate all comparanda for Moche II pots. It would also be easy to select specific authors for a partial bibliography, working only in the *Sources* Table, or to gather those bibliographic sources that relate to any of the Moche II pots, again using two tables.

SQL SELECT statements will show how these queries can be stated:

To find all pots that use "Donnan 1978: fig. 239" as a comparandum:

```
SELECT Pots.CatalogNo. (and, by naming them, any other columns desired)
FROM Pots WHERE Pots.ObjectNo. = Comparanda.ObjectNo. AND
Comparanda.ComparedObject = 'Donnan 1978: fig. 239'
```

The use of "Pots.CatalogNo." to specify the column makes clear the need to pick the column from a specific table since multiple tables are involved here. In addition, the conditions in this example include both the primary-key-to-foreign-key link and the specific item used as a comparandum.

To find all comparanda for Moche II pots:

```
SELECT Pots.CatalogNo., Comparanda.ComparedObject FROM Pots,
Comparanda WHERE Pots.ObjectNo. = Comparanda.ObjectNo. AND Pots.
Period = 'Moche II'
```

In this case a column from each table has been requested as output, with the requirement of the primary-key-to-foreign key link and the added requirement that only pots dated to the Moche II period be included.

As you might imagine, it is also possible to make a similar request with less precise requirements. Thus, the following statement would return the same results if "Moche II" were always in the period column. If, however, there could be extra characters either before or after "Moche II" in the period column, this new version of the statement, utilizing the wildcard character, %, would be better:

```
SELECT Pots.CatalogNo., Comparanda.ComparedObject FROM Pots,
    Comparanda WHERE Pots.ObjectNo. = Comparanda.ObjectNo. AND Pots.
    Period LIKE '%Moche II%'
```

Here the LIKE condition lets the wildcard (%) stand for <u>any</u> character(s) prior to or following the specified ones. (There are many web sites with more information about forming SQL statements; one such site is the W3 Schools site, www.w3schools.com/default.asp.)

Using Numbers, Not Described Date Ranges, for Dates

There are several reasons for using explicit numeric dates rather than verbal descriptions thereof in a data table. First, of course, the use of numbers means that the information is much more precise. That is a double-edged sword. Archaeological information is rarely that precise. How many artifacts have specific years for beginning and ending dates attached?

But what does it mean to say that an artifact can be dated to the span from "mid 8th century to early sixth"? Does that mean 750 B.C.E. to 590 B.C.E. or 780 to 560 or some other choice? The use of numbers removes ambiguity while implying a level of specificity that is not real.

The level of specificity can be properly conveyed by adding a column to explain the beginning date and another to explain the ending date. For instance, a beginning date might be recorded as -760 (equivalent to 760 B.C.E.) and the associated explanation might be "mid 8th century B.C.E." Similarly, the ending date might be -575 and its explanation "early sixth century B.C.E." Why go to the trouble of having both absolute years and text explanations?

Numbers permit ordering, and use of separate beginning and ending dates permits ordering by either criterion. (As indicated above, we can use a minus sign to indicate dates B.C.E.) Numbers also enable users to search in ways they cannot otherwise, for instance, looking for all fibulae thought to have been "in style" between 600 B.C.E. and 500 B.C.E.

This issue is typical of the complexity required by the switch from paper records to computerized ones. The database design must take account of the needs for accuracy and explicitness **and** permit useful searches.

Artifacts from an Excavation

Now we turn to another rather simple archaeological example. This table contains information about the fibulae found in the great tumulus called the Midas Mound at Gordion, Turkey. (A fibula is a rather elaborate safety pin used in antiquity to hold clothing in place, as a dress pin.) The fibulae were found in various places in the tomb, and information about them was presented in the excavation report as a group of printed tables. For each fibula style there was a separate section of text with multiple tables, each table showing information about the fibulae of the style in question that were found in a particular location. (A group of extraordinary examples was treated separately.) The tables, taken together, contained all the information about all the fibulae from the tomb excepting the ten extraordinary ones treated separately.

A few years ago I experimented with these Gordion fibulae by combining the published tables into one large computer table (with the inventory number as the primary key) to make some new views and analyses of

Sample Entries from the New Table of Gordion Fibulae

Inventory No.	Mate	Findspot	Style Group	Style Sub-group	Other Typological Trait(s)	Pin Side	Height (m.)	Length (m.)	Description	Plate No.
MM 242 (B 911)	MM 241	Floor, at head of bed	XII	9	3 studs on each block	R	0.03	0.04	Rounded cushion between blocks. Mushroom-shaped spring button. Spring-buttons plain. Cushion between blocks small. Two studs lost.	
MM 243 (B 906)	MM 244	Floor, at head of bed	XII	9	3 studs on each block	L	0.04	0.05	Rounded cushion between blocks. Mushroom-shaped spring-button. Spring-buttons plain. Cushions small.	78A
MM 244 (B 823)	MM 243	Floor, at head of bed	XII	9	3 studs on each block	R	0.04	0.05	Rounded cushion between blocks. Mushroom-shaped spring-button. Spring-buttons plain. Cushions small.	78B
MM 245 (B 908)	MM 246	Floor, at head of bed	XII	9	3 studs on each block	L	0.04	0.05	Similar to MM 244	

Figure 5

Sample data entries for fibulae from the Gordion Midas Mound. (Note that the utility and value of the database are seriously diminished by using the phrase "Similar to MM 244" in place of a true description because a search for fibulae with any characteristics of interest would omit this example.

the data possible. In the process, some data items had to be added. Information about the fibulae had been presented in tables for groups and sub-groups (as determined with the aid of the standard classification work) as well as findspot. The new database table contained all the fibulae; so each entry had to have a column for group, another for sub-group, and one for findspot so all the information in the publication would be preserved. Eventually the available information about mates was also added. (Many of the pins were made as pairs, a right- and a left-sided version of the same design.) The completed data table had all the entries for Midas Mound fibulae from the various tables in the publication. A selection of entries from the new table is shown in figure 5. This data includes stylistic information as well as findspot information, both of which were unnecessary in the published tables since each table contained only examples of a specific style from a given findspot.

The computer version of the table is very similar to the paper tables in the publication. All the same information exists in the computer table, but it is one large table instead of many smaller ones. Individual tables could easily be created to match the original paper tables by selecting only examples that meet certain criteria: group, sub-group, and findspot. However, there are several things the computer version of the table makes possible that the paper tables do not. First, the arrangement can be changed from moment to moment, depending on the needs of the user. For instance, one might like to select all the fibulae of a given style to see where they were found or all those from a given location to see how many styles were represented there and/or in what proportions. One might also want to organize them by one of the measurements to see what the range was and to see how closely variable the dimensions are (perhaps preparatory to applying a statistical test).

Of course, the way the tables are organized limits the possible uses of the data. For that reason it is not too early in this discussion to point out an issue that may already be obvious. When organizing a database, it is always best to break the data into the smallest possible segments (called atomizing the data). Those segments can always be combined when desired for presentation, but the segments provide the

Sample Entries from Table of Fibulae Styles

Style	Sub-style	Date	Description
XII	9	mid 8th century to early sixth	near semi-circular bow, hook catch usu. with spur or volute at base & grooves of some sort on outer face, moldings at bow end & often middle – flat x-sec of bow with bosses
XII	7	8th century	near semi-circular bow, hook catch usu. with spur or volute at base & grooves of some sort on outer face, moldings at bow end & often middle – cf. 13, but flatter bead, more prominent reels, no mid-bow version
XII	5	late 8th into 7th century	near semi-circular bow, hook catch usu. with spur or volute at base & grooves of some sort on outer face, moldings at bow end & often middle – small, blocks at bow ends

Figure 6

Sample data entries for fibulae styles found in the Gordion Midas Mound.

search, ordering, or discrimination criteria that will be crucial for inspecting the data in actual use. Had the presentation of the data about the fibulae not included the careful description of findspots, for instance, it would not be possible to search or group the examples effectively according to that information; were stylistic group and sub-group not separately entered, accessing them separately would have been very difficult. On the other hand, the absence of a simple indicator of completeness prevents us from examining these fibulae with the aid of that criterion. Would that have been desirable when recording these fibulae in a computer data table? If so, the data should originally have been collected with that in mind. In other words, it is at least as important to know what questions you may want to ask when you start recording information for a database as when you start recording for a card file. When in doubt, it is better to split the data into too many categories and to record too much information than to use too few categories or to omit information. As with most recording systems, changing the system after some data have been entered can be difficult. (Note that, in the case of the Moche pottery, the dimensions were split into two data items, height and diameter, rather than combined in a form such as "26 x 16.5.")

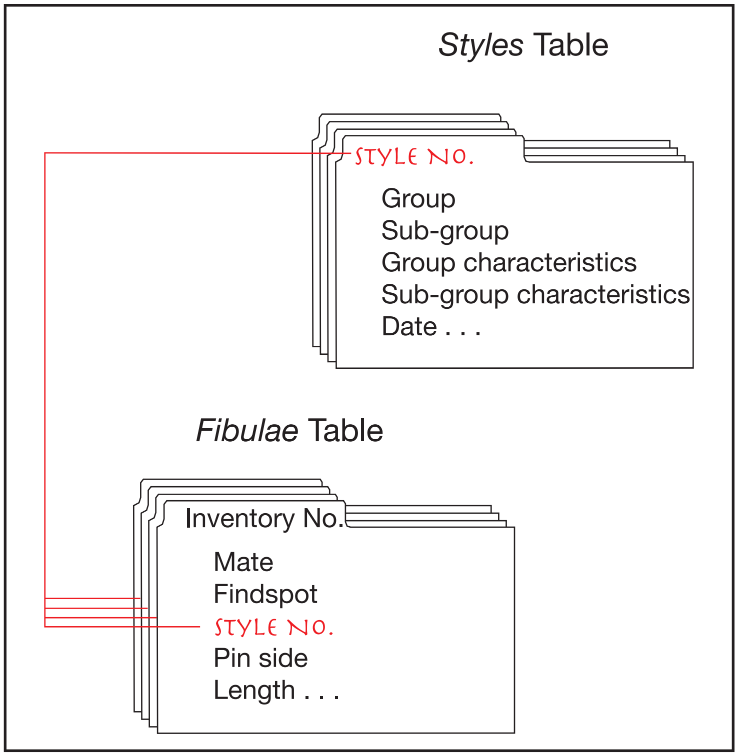


Figure 7
The Gordion fibulae database schema with *Fibulae* Table as the child of a Styles Table. (Ideally Group and Sub-group should have been treated in separate tables, but the data available when making the tables did not permit that.)

To return to our Gordion fibulae, we might want to order them by date, but we only have stylistic notations to indicate the group and sub-group for each fibula. We could add the date range provided by the scholar who generated the stylistic system, but think how inefficient that would be. Every entry would need a date, and any change in the dates applied to a given fibula group or sub-group would require changing the date for every fibula in the group or sub-group. Not only is this inefficient, it violates one of the cardinal rules of database design (to which we will return for a fuller discussion): No column in any row of a table should have its content implied by any other column save the primary key; thus, the date is implied by the style and should not be separately entered. (A simpler example: the fibulae have lengths specified in m.; entering length in inches as well would be redundant.)

The better solution for dating the fibulae is to create a second data table that lists the groups and sub-groups and the dates attached to them. In this table, which we will call the *Styles* Table, there may also be a drawing of a typical example of the style, and there might be page references to the appropriate portions of the classification study of fibulae. A text definition of the group and sub-group might also be included. In short, anything considered relevant to the particular group plus sub-group could be included – but nothing about individual fibulae.

For the purposes of this exercise, there are only a few things in this second table. First, there are stylistic designations – group and sub-group. Then there are defining characteristics of the group and sub-group. Finally, there is a date. Other matters, such as a specific bibliographic reference, might be included.

The new table here is related to the *Fibulae* Table in the reverse of the way the *Comparanda* Table and *Sources* Table were related to the Moche *Pots* Table. In the previous case, the Moche *Pots* Table had a one-to-many (parent-child) relationship to the others; a single pot could have many comparanda or sources. Here the *Fibulae* Table has a many-to-one relationship (child-parent) to the *Styles* Table; many fibulae are of the same style. The *Fibulae* Table is the child table, and the *Styles* Table is the parent table. Therefore, the foreign key must be in the *Fibulae* Table; each foreign key must match a unique entry in the *Styles* Table, the primary key there. Furthermore, in this case the primary key in the *Styles* Table and the matching foreign key in the *Fibulae* Table must consist of two columns together, the group and sub-group columns. The sub-group is a crucial element of the *Styles* Table; so the organization must use both group and sub-group. Here again, I might actually ignore the database theorists and use a sequence number in the *Styles* Table as the primary key rather than making the primary key a combination of group and sub-group. That is the schema shown in figure 7. Note that the Style Group and Style Sub-Group are no longer in the *Fibulae* Table. Style groups and sub-groups are supplied by the *Styles* Table and the primary-key-to-foreign-key link. (Note also, however, that this table design is far from perfect since style groups are repeated.)

This arrangement for the fibulae is clearly hierarchical; the *Styles* Table – a partial list of the table entries is shown in figure 6 – lists groups and sub-groups. The *Fibulae* Table lists examples from those groups and sub-groups.

The Gordion example has one interesting feature that deserves mention. In the *Styles* Table, there is a description of the date, rather than a number, because that was what could be extracted from the readily available materials. There should have been a beginning date and an ending date, as were used in the Moche pottery catalog to permit more natural searches.

The museum database used one-to-many relationships; details in the child tables augmented the base file with multiple sources and comparanda. The Gordion database used the many-to-one relationship to the *Styles* Table to group the fibulae rather than adding further detail; a parent table was used to add stylistic groupings to the information about individual fibulae.

Complexity Need Not Interfere

Both the Moche database and the Gordion example involve more than one data table, but users of either set of data need not see the complexity of the presence of multiple tables if that might be a problem; indeed, neither data entry personnel nor those simply looking at the data need to know that there are multiple tables involved. Data entry can be done as if there were only a single table, and the data in the system can be presented to users in a way that ignores or even actively suppresses the fact that the data reside in multiple tables.

For instance, two simple but different forms for the Moche pottery information illustrate what someone might see while entering data into the system or viewing the data at a later date (figures 8 and 9). The sources and comparanda can be identified individually, but the form shows all the information as a straightforward, seamless whole. The fact that three tables are involved is not apparent at all in the version shown in figure 8; the boxes surrounding comparanda and bibliographic sources in figure 9, on the other hand, suggest clearly the ways the data have been stored without making that seem to matter. In both examples all data shows on one form, with no clear statement that the data have been stored in two separate tables. Neither a data-entry person nor a scholar using the data would have any reason even to wonder about the underlying data structure. (Needless to say, the underlying structure can be made very obvious if that serves some purpose.)

Your mind may now be churning with the possibilities offered by one-to-many and many-to-one relationships. Any time we have relationships between individuals and their groups, we can use file relationships to permit a single entry of information about a group to be used for all members of that group. Similarly, when there are multiple entries for a specific column – e.g., sources – file relationships can make those multiple entries as easy to use and understand as a single one –and far more useful than the same data crammed into a single column.

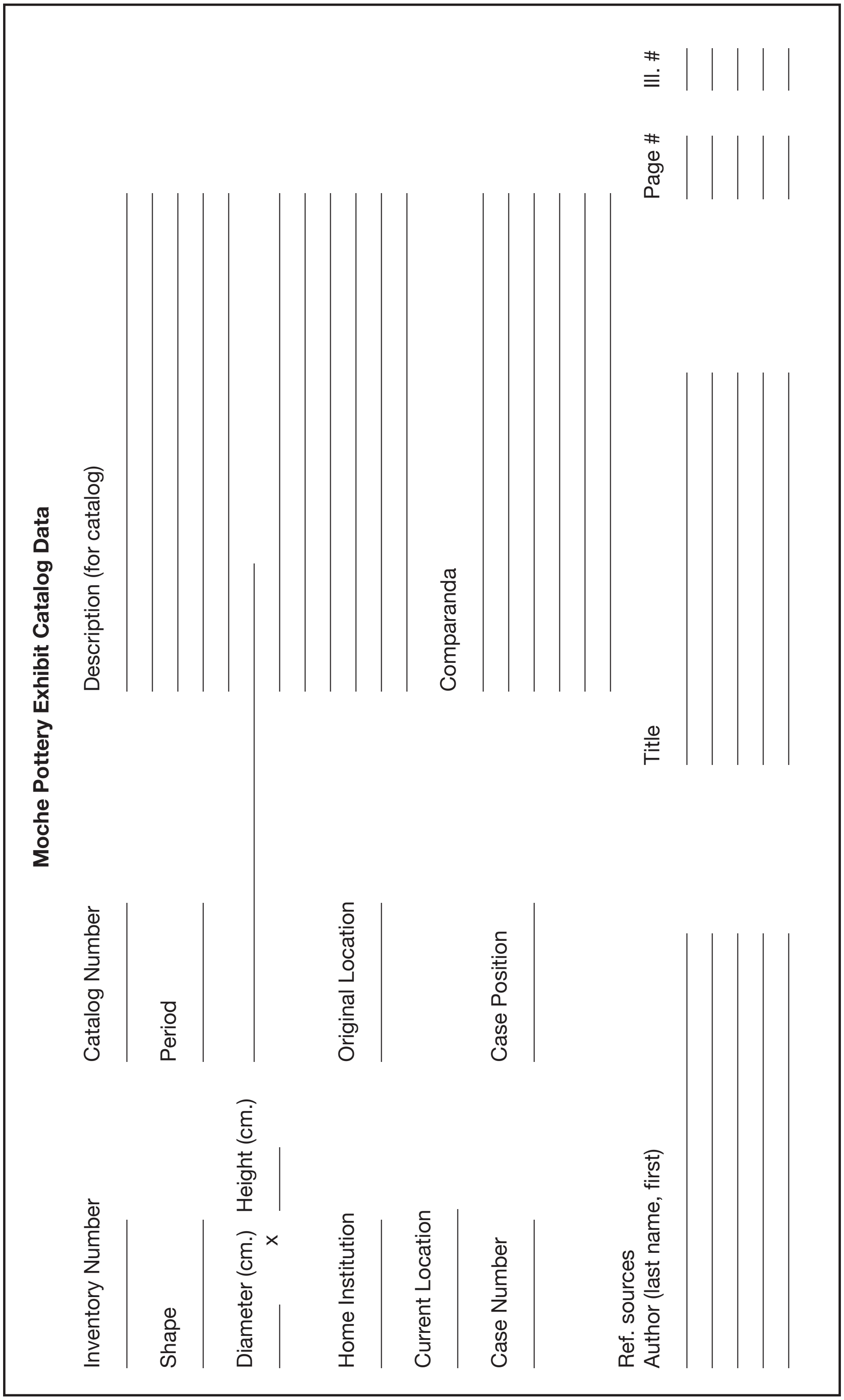
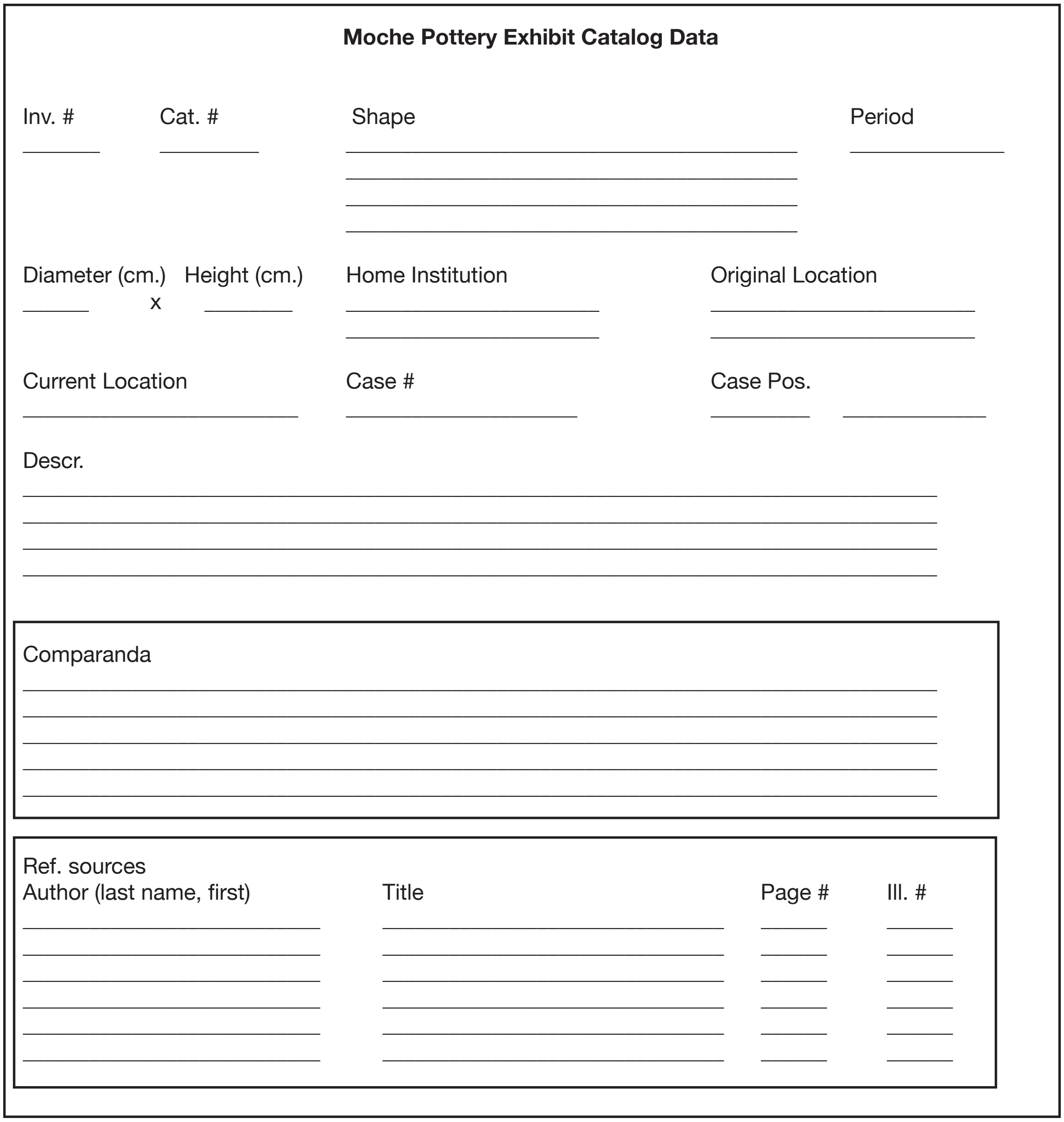


Figure 8

A data form as set up for Moche pottery data on a computer. Note that the data for comparanda and sources seem to be part of the pottery data, in no way separate. (New lines for references would be added automatically as needed in a data-entry form.)



Moche Pottery Exhibit Catalog Data

Inv. # | Cat. # | Shape | Period

Diameter (cm.) x Height (cm.) | Home Institution | Original Location

Current Location | Case # | Case Pos.

Descr.

Comparanda

Ref. sources
Author (last name, first) | Title | Page # | Ill. #

Figure 9

A different data screen for Moche pottery data. Here both comparanda and sources are in boxes to make clear their different subject matter, something that can easily be done or not, as the database designer prefers.

An Excavation Database

It is now time to use the information we have amassed to organize an excavation database – an example based on an actual excavation database. The following are some of the categories about which information needs to be recorded: operations (excavation trenches) and loci (contexts); ceramics lots, catalogued objects, photographs, drawings, and personnel. In addition, there are some housekeeping details that need to be tracked; for instance, objects in need of photography or drawing. Other records are required for an excavation database, but this list provides ample complexity to start. (In this scheme, the locus is the smallest volumetric excavation unit that is measured and surveyed. Lots are considered to have come from anywhere within the locus and do not have physical boundaries separate from those of the locus. Note that it is critical to understand the way the excavator uses such terms in order to construct a database that is true to the excavation system.)

For each of these categories there should be a data table. A hierarchical model suggests that there would be a parent table for trench information, a child table of the *Trench* Table for loci, and a child table of the *Loci* Table for ceramic lots. (Of course, there would be other child tables for metal lots, bone lots, etc.) They would exist in a hierarchical relationship, with trenches being the top level, loci next, and lots at the lowest level. Trenches contain loci, and loci contain (multiple different kinds of) lots. Loci also contain catalog items.

There may be – in this case there is – another level to the hierarchy. Sherds in each lot, in this excavation, are recorded by style. That is, a lot may contain several wares, and the number of rim, body, foot, and handle sherds will be recorded for each. Since there can be many different wares in a given lot, another table is added to the hierarchy, this one with the wares and specific numbers of sherds for each ware found in each lot. (Note that only sherds would be included here; whole vessels would be in the catalog, not the *Ceramic Lots* Table.) Figure 10 shows the

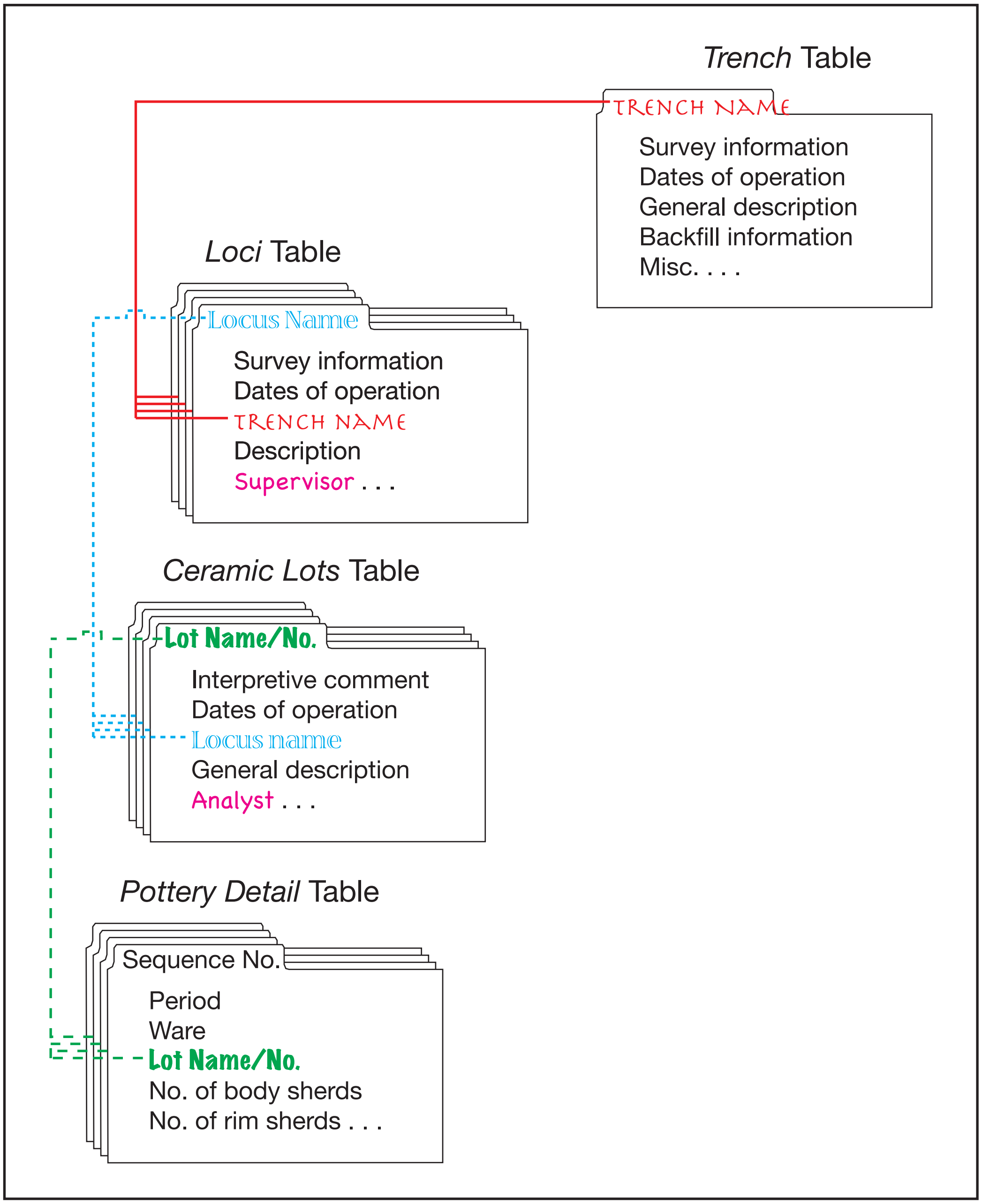


Figure 10

Excavation database schema with trenches containing loci. Loci, in turn, contain ceramic lots. Ceramic lots require pottery details. (Other types of lots – and the detail tables required for them – are not shown.)

Interrupted Data Entry or Edit Procedures

Protecting the integrity of the data is a job of prime importance for the database system designer; corrupted data may be completely worthless. Data are most easily corrupted when data entry or edit processes are interrupted by loss of power, e.g., when data are entered or edited in the field where the electrical supply is uncertain. If data entry is interrupted by a power outage, a program or OS crash, or for any other reason, the database must not be contaminated. The problem is particularly difficult when multiple tables are adjusted in the same process; the database tables should not be altered until everything can be adjusted at one time.

Some DBMSs have excellent built-in ways to prevent contamination of the data, but not FileMaker or Access. So a database designer will often have to provide some mechanism via data entry procedures. The best such approach is to use temporary tables that hold data outside the actual data tables. Sometimes the information in the temporary files will be stored for a time and then integrated into the overall files in a discrete procedure that only the system administrator is empowered to initiate. Other systems will add new data to the database at the completion of the data entry process for a given entity, emptying all temporary data at the same time. Both approaches are good, but one or the other will probably have to be implemented with most databases intended for use in the field.

Editing requires the same care. The edit process should leave the original data untouched until final approval by the editor, when all changes are made simultaneously.

schema. There are four tables in a single hierarchy arising from three parent-child (one-to-many) relationships.

Any table in the hierarchy below another includes a column with a name/number to serve as its foreign key and to connect it to the table above in the hierarchy. As a result, each group of pottery sherds from the *Pottery Detail* Table can be related directly to the proper lot and, via the lot-locus relationship, to the proper locus and, via the locus-trench relationship, to the proper trench.

This schema deals only with ceramics; there would be another hierarchy descending from the *Loci* Table for metal objects, another for stone objects, and so on. The schema in figure 10 would be repeated in some form, depending on the way data are recorded for other finds, for each category.

You will note that the *Loci* Table includes a column for supervisor, and the *Ceramic Lots* Table includes a column for analyst. It is unlikely that the locus supervisor is the person who fills out the pottery analysis forms; so it is important to record the names of the people doing both jobs. If the excavation system requires that the locus supervisor fill out the lot forms, though, it would be unnecessary to list an analyst for the lot. (Identifying people responsible for data entry is often thought to be unnecessary. This is not a database design question; it can be done – automatically and without the data entry person wasting any time on the task – in any DBMS. The question is one of what knowledge matters. I am firmly convinced that data entry personnel should be specified; you may not be.)

A unique name or number used to identify the supervisor, implies the existence of other information about that person in the system. At a practical level mailing address, phone number, and the like should be recorded, and at a more theoretical level differences from supervisor to supervisor should be explicit when one tries to evaluate information. A supervisor may even be partially color-blind and not know it at the time he or she identifies pottery colors. Indeed, a host of issues about the experience and qualifications of staff members will be of interest to future users of the data; so there should be a table about personnel that includes relevant information about each person who works on the project, including information about the seasons worked, experience gained elsewhere, and other factors that might relate to the quality and reliability of project data. As noted of course, housekeeping information such as address, phone number, email address, and the like should also be recorded. (Most personal information should be eliminated from the *Personnel* Table when the database is archived for privacy reasons.)

With a *Personnel* Table added, it would be possible to select only lots recorded by an individual project member or loci supervised by a particular team member, but note what happens to the hierarchy with the addition of the *Personnel* Table. That table is not in a hierarchical relationship to the others, only to the *Lots* Table and the *Loci* Table.

That is not the end of the complexity. When a *Catalog* Table is added for individual artifacts, it stands in a hierarchical relationship to the *Loci* Table and to the *Personnel* Table; the *Catalog* Table is a child table for both, yet the *Loci* Table is also a child table for the *Personnel* Table. (See figure 12 for the expanded schema.) The database as a whole can no longer be viewed as even a group of hierarchical relationships; it is much more complex than that already, with relationships that are clearly hierarchical and others that are not.

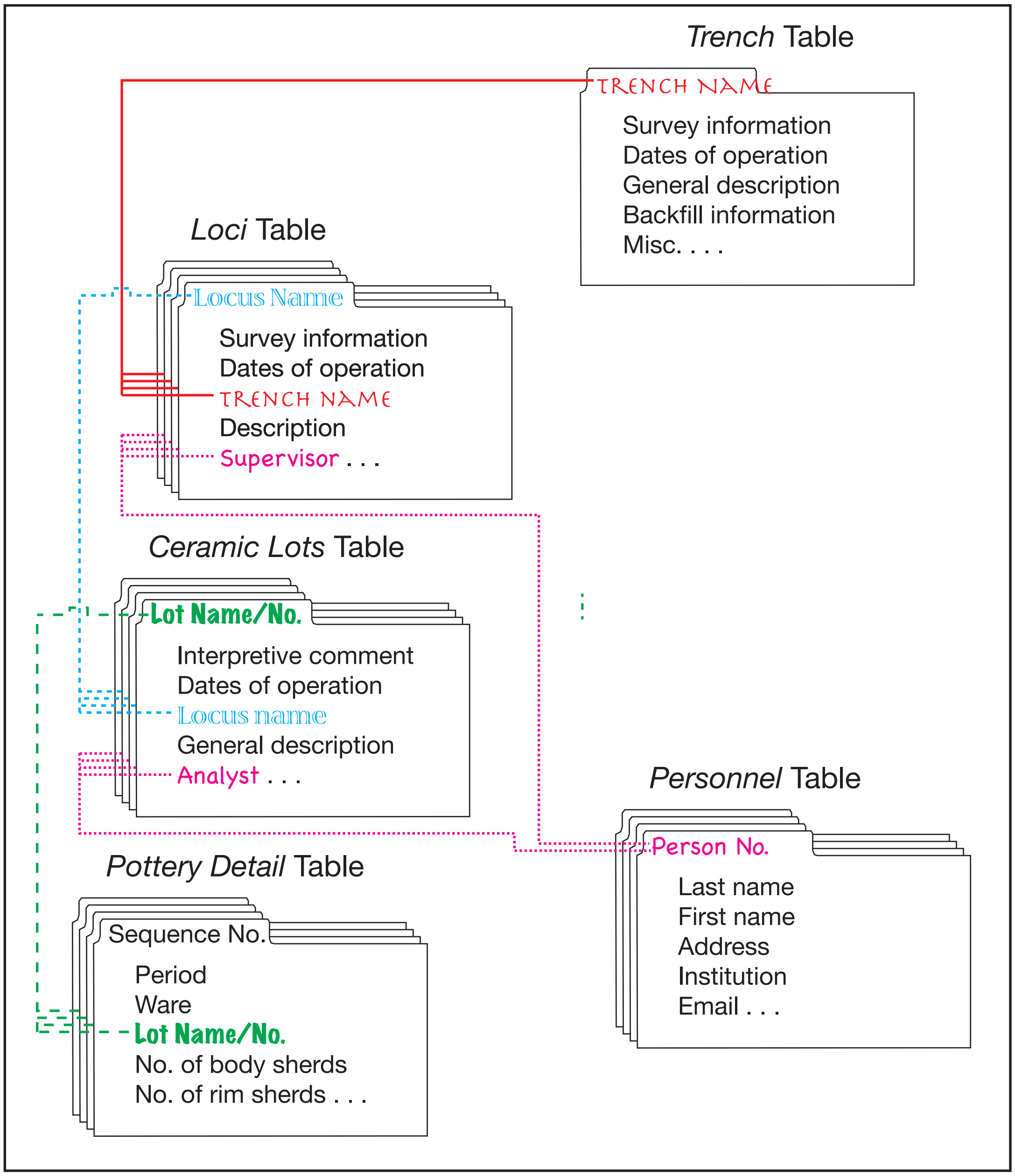


Figure 11
Excavation database schema with added *Personnel* Table.

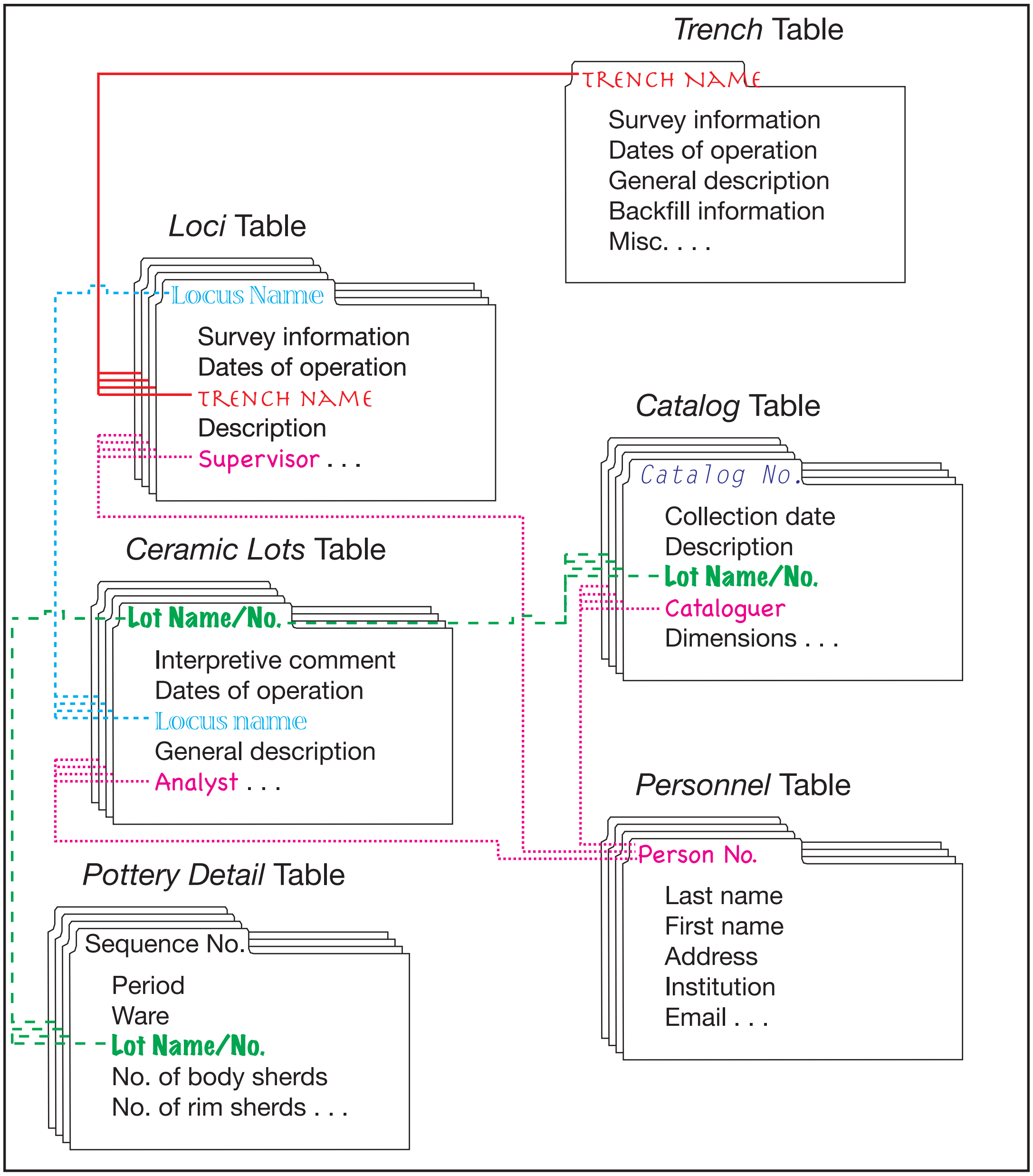


Figure 12

Excavation database schema with added *Personnel* Table and *Catalog* Table.

Note the selection and searching opportunities that are now available. A user might find all the lots analyzed by a given person and from that selection locate all pottery detail rows descending from that analyst. One might also find all pottery in the *Pottery Detail* Table of a particular style and work back, through the *Ceramic Lots* Table, to find the loci containing pottery of that style. Of course, it would also be a simple matter to obtain all the pottery from a given locus. The point is that one may use tables in combinations from anywhere within the system if the organization of the tables has been carefully planned.

Constructing SQL statements may again help to show how the data may be queried. As is so often the case, there are multiple ways to satisfy our needs, and I will choose those that keep the new SQL syntax to a minimum. These may not always be the quickest ways to get the results desired.

To find all loci with sherds of a specific period, for this example “EBA 1,” the SQL statement remains simple, starting with the *Pottery Detail* Table:

SELECT PotteryDetail.LotNameNo., CeramicLots.LocusName FROM PotteryDetail, CeramicLots WHERE PotteryDetail.LotNameNo. = CeramicLots.LotNameNo. AND PotteryDetail.Period ='EBA 1' (Yes, we could ask for only unique locus names rather then repeating those with many examples of EBA 1 pottery.)

To find all the pottery in the Pottery Detail Table that was recorded by a specific member of the team (George Smith) requires more complexity: SELECT Personnel.PersonNo, Personnel.LastName, Personnel.FirstName, CeramicLots.LotNameNo., CeramicLots.LocusName FROM Personnel, CeramicLots WHERE Personnel.PersonNo = CeramicLots.Analyst AND Personnel.LastName.='Smith' AND Personnel.FirstName = 'George' INTO TempTable -- and then this:

SELECT TempTable.LastName, TempTable.FirstName, TempTable.LotNameNo., TempTable.LocusName, PotteryDetail.Period, PotteryDetail.Ware, PotteryDetail.BodySherds, PotteryDetail.RimSherds FROM TempTable, PotteryDetail WHERE PotteryDetail.LotNameNo. = TempTable.LotNameNo. INTO GeorgeSmithTable

This process first created a new table (*Temp* Table) with the INTO syntax putting into each row in *Temp* Table the name and Person No. for the team member whose recorded pottery was wanted, George Smith; a ceramic lot he analyzed; and the locus for that lot. This new table automatically used the column names from the beginning tables. The second statement used *Temp* Table with the *Pottery Detail* Table (and the critical primary-key-to-foreign-key link planned in the prior

Objects and Lots

Some excavators want objects to have lot numbers, since lot numbers can be assigned by the trench supervisor at the time of excavation and provide an important housekeeping check. The catalog number must wait for the recorder to assign it. A whole pot, however, may not be treated as a lot in the same way that a group of sherds can, since it consists of a single item. So how does one treat the catalogued objects?

One common approach is to leave catalogued ceramics completely out of the *Ceramic Lots* Table. Catalogued objects are not analyzed in the same way, recorded in the same way or at the same time, or accessed in the same way. On the other hand, if lot numbers are seen as the housekeeping sequence for each type of object, an excavator might want to assign a lot number to anything removed from the site, either individually or as a group. Catalogued objects might be recorded in the *Ceramic Lots* Table with only the lot number and a catalog number – something that could be done automatically during catalog data entry. That would allow the lot number to retain its primacy without stressing the *Ceramic Lots* Table.

How best to treat catalogued objects is typical of database design questions. The issue is not the requirements of the computer; the issue is the needs of the excavator/excavation. Where should objects be in the hierarchy? How should lots be construed? When the excavation questions have been answered, the database must follow suit, and even then the practical questions about the way the excavation recording system will be used are key. The technology is remarkably flexible. (I have been quoted as saying that a database designer who cannot do what the excavator wants should be fired. Though my feet have been held to the fire, with those words ringing in my ears, I continue to believe in the statement's essential accuracy. The database must do what the excavator needs, and it is the job of the database designer to make sure of that result.)

The typical excavation makes these kinds of questions more complex. What does one do, for instance, with a catalogued object that is made up of pieces from multiple lots? The answer to that question must be the excavator's. With that answer, it should be possible to structure the database "correctly" – satisfying both the director and the database designer.

SELECT statement, the Lot Name/No.) to find the analyses performed by George Smith. Note that the columns chosen permit a new table to be fully self-explanatory, showing George Smith's name, the locus number, and the lot number for each group of sherds analyzed by him.

Among the information to be recorded about the excavation is information about objects in need of photography or drawing. That should be very simple, and it is. One need only add a small table, to be called the *Needs* Table, with

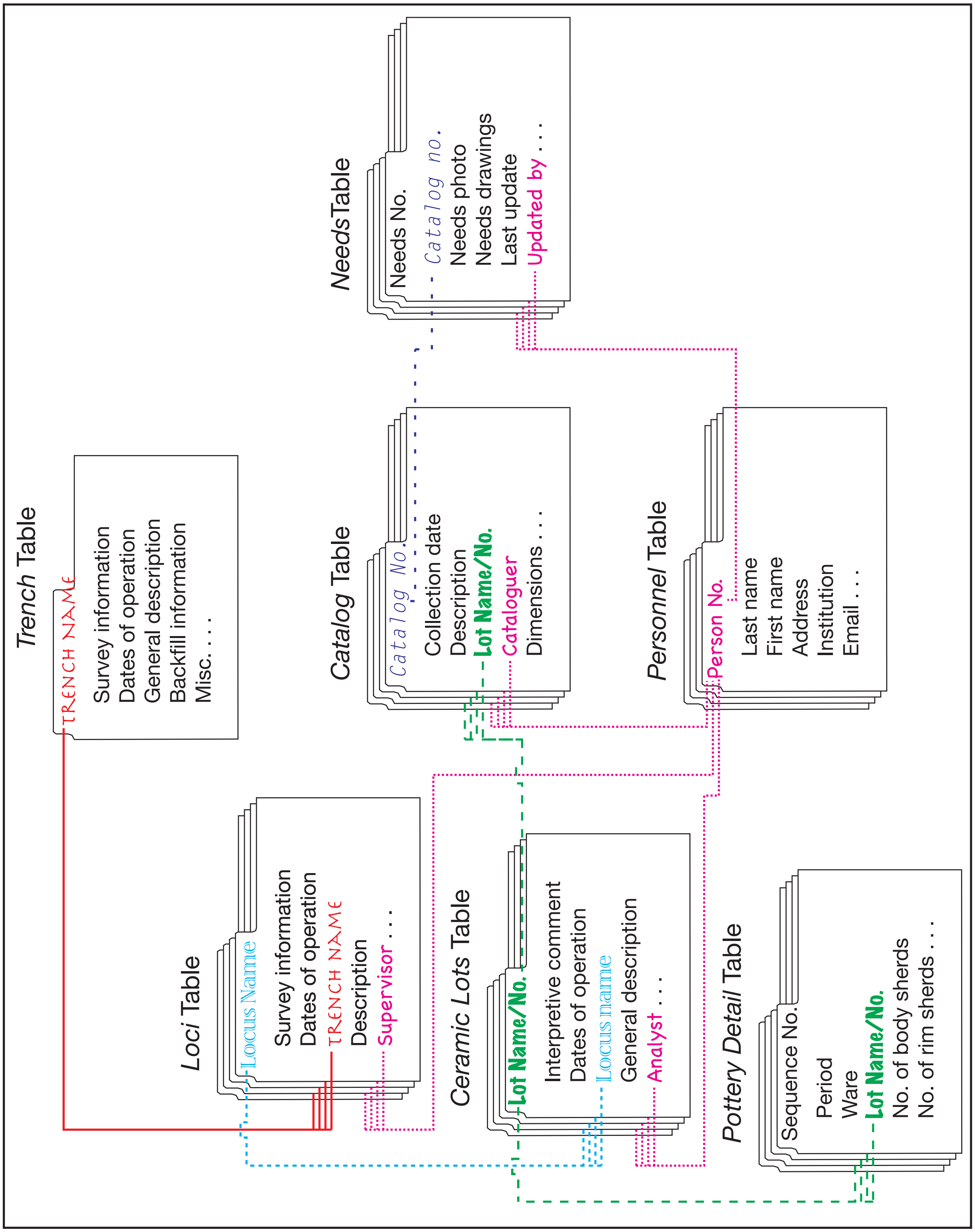


Figure 13
Excavation database schema with added *Needs* Table.

three columns: catalog number, "needs photograph?", and "needs drawing?" The "needs photograph?" and "needs drawing?" columns can be simple yes/no columns. Another table may seem to imply another data entry process, but, as with the Moche pottery data entry screen, the information can be added to the catalog entry screen during normal data entry processes, even though it is to be stored in a separate table. Note that the *Needs* Table is related to the *Catalog* Table in a one-to-one relationship. For each catalog object there may be an entry, but only one. For each entry in the *Needs* Table, there must be a catalog entry – and only one catalog entry. See figure 13 for the schematic view at this level, and see "True to the Original and to Today's User," p. 227 for another good example of the one-to-one relationship. (Note that the table could apply to lots, loci, or operations as well as catalog items, assuming the excavator might want a locus, an operation, or the content of a lot photographed. In that case, the *Needs* Table should be structured differently. Each row could contain a sequence number, the "needs photograph?" and "needs drawing?" columns and two more columns: one for the type of subject – catalog item, lot, locus, etc. – and the other for the identifying name/number of the subject. This approach, though, requires foreign keys from multiple tables in the *Needs* Table. Database purists would reject this design.)

Why not simply add the two columns – "needs photograph?", and "needs drawing?" – to each table? Simply stated, this is not information that really belongs in the finished database. Indeed, the *Needs* Table will be removed at some future date, when it no longer has entries because the work is complete.

To this point we have seen three kinds of table relationships – one-to-many, many-to-one, and one-to-one (the *Needs* Table). You are probably wondering about the obvious missing relationship: many-to-many. A many-to-many relationship conceptually resembles two one-to-many relationships operating in opposite directions at the same time, but it is more complex still.

There were in the Moche pottery database two examples of many-to-many relationships, but only a one-to-many relationship was noted and discussed in each case. Some pots in the basic table had more than one comparandum, and some had more than one bibliographic source; so we constructed child tables for comparanda and sources. However, the situation was actually more complex than acknowledged. Many comparanda and sources applied to more than one pot. Therefore, the *Comparanda* Table and the *Sources* Table could have been perceived as parent tables for the *Pots* Table rather than child tables. Of course, there must be a way to express this dual relationship where a given file functions simultaneously as a parent and a child table for another. This is the many-to-many relationship.

The many-to-many issue is clearly problematic when one examines just the first few examples from the Moche data, as shown in figure 14, which includes only the catalogued objects as comparanda, not objects from publications that would further complicate matters. Which direction of the lines – right-to-left or left-to-right – indicates the relationships? Is object number 1 related to catalog numbers 76, 78, and 79? Or is catalog number 76 related to objects numbers 1, 3, and 6? Of course the relationships go in both directions, but the parent-child relationship only makes explicit

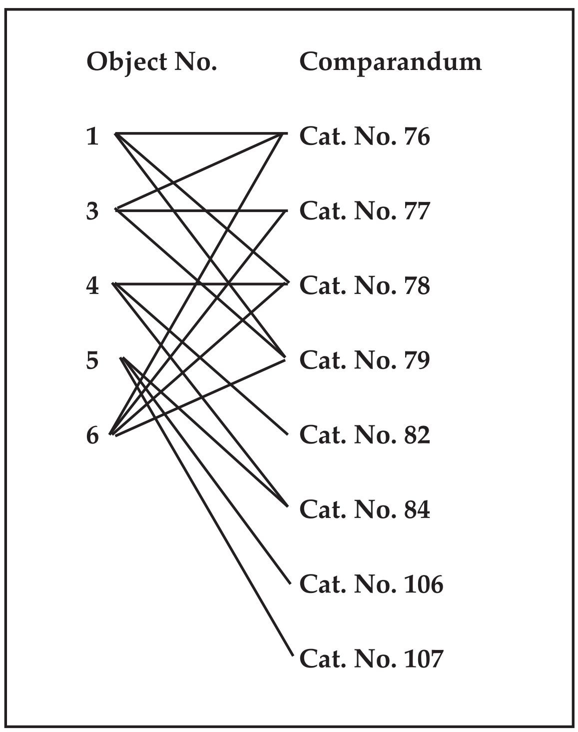


Figure 14
Objects with comparanda – a many-to-many relationship.

the link from object number to comparandum. That is, having the object number and the defined primary-key-to-foreign-key link is sufficient to find all comparanda for any object. The reverse is not true; all foreign keys for all occurrences of any given comparandum would be needed to find the objects referenced. That link from comparandum to object seems to be implicit with the catalog numbers, but it breaks down when we go beyond catalog objects to references from publications. There can be no simple primary-key-to-foreign-key link there. The relationship here is certainly a many-to-many relationship: many pots and many comparanda related to one another.

The best way to deal with this complex set of relationships is to treat the object used as a comparandum separately from the comparison. Then the *Comparanda* Table can contain just the objects used for comparison purposes, one object per row and with no objects repeated. Another table can lie between the *Pots* Table and the *Comparanda* Table, linking them by listing – and explaining – the relationships. Thus, the *Comparanda* Table lists all the objects used for comparison purposes, one object per row, whether the objects are from the exhibition or other museums or publications. A new table, let us call it the *Comparisons* Table makes the connections between objects in the *Pots* Table and objects in the *Comparanda* Table. To be effective, the *Comparisons* Table should do more than list the two items being compared; it should also state the nature of the relationship. For example, one row would contain the primary key for Catalog No. 77 in the *Pots* Table (the object number from the *Pots* Table as a foreign key in this table, identifying the object being compared); it would also have the primary key for the row containing the

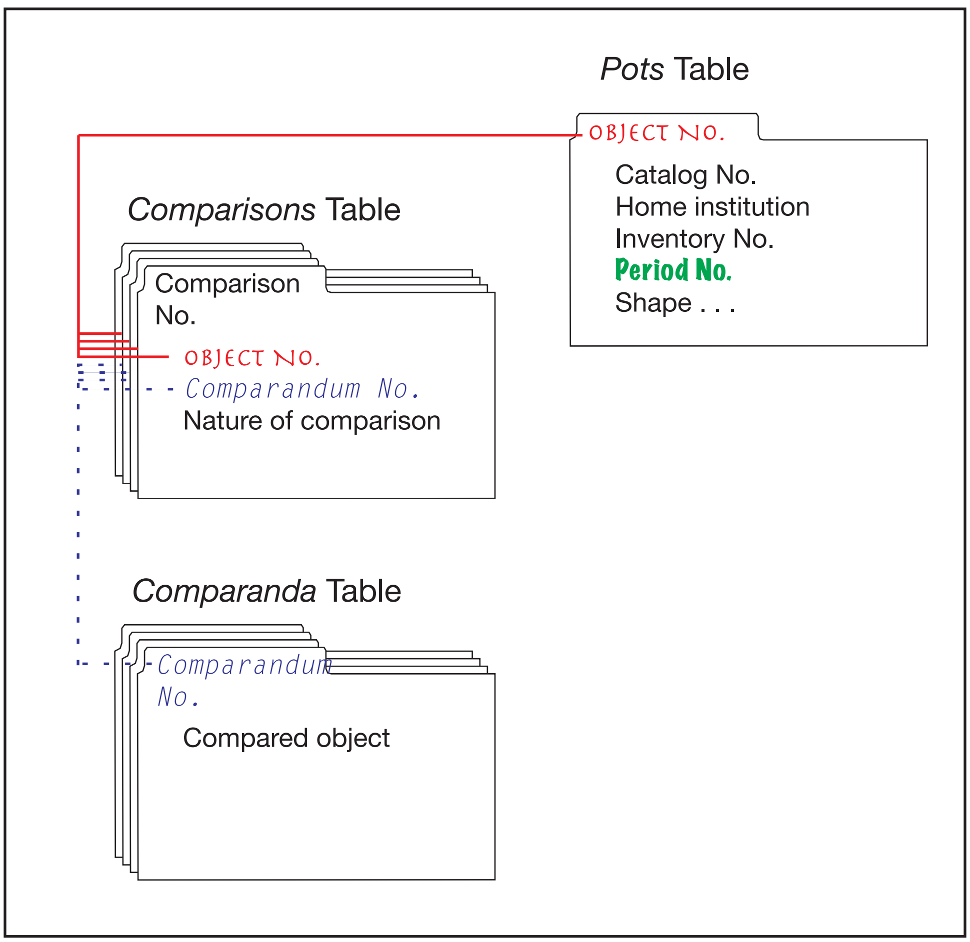


Figure 15
The Moche schema with only the *Pots*, *Comparanda* and *Comparisons* tables to show the many-to-many relationship more simply

comparandum "Lavelle, 1989: Pl. 159," in the *Comparanda* Table. The comparandum number from the *Comparanda* Table is also a foreign key in this table, identifying the object used as a comparison). Finally the *Comparisons* Table also contains the nature of the comparison: "face decoration for coca-chewing rites."

Using this approach, any number of references from an object in the *Pots* Table to a compared object in the *Comparanda* Table can be stated. At the same time, any number of reverse relationships – from an object listed for comparison purposes to one of the objects in the *Pots* Table – can also be stated. Each relationship, whether from exhibit pot to comparison or from comparison to exhibit pot – is referenced and explained as a relationship, but no object is listed more than once in its base table. Note that this approach has another advantage. Since each relationship is expressed individually, each has its own explanation as well as referenced objects;

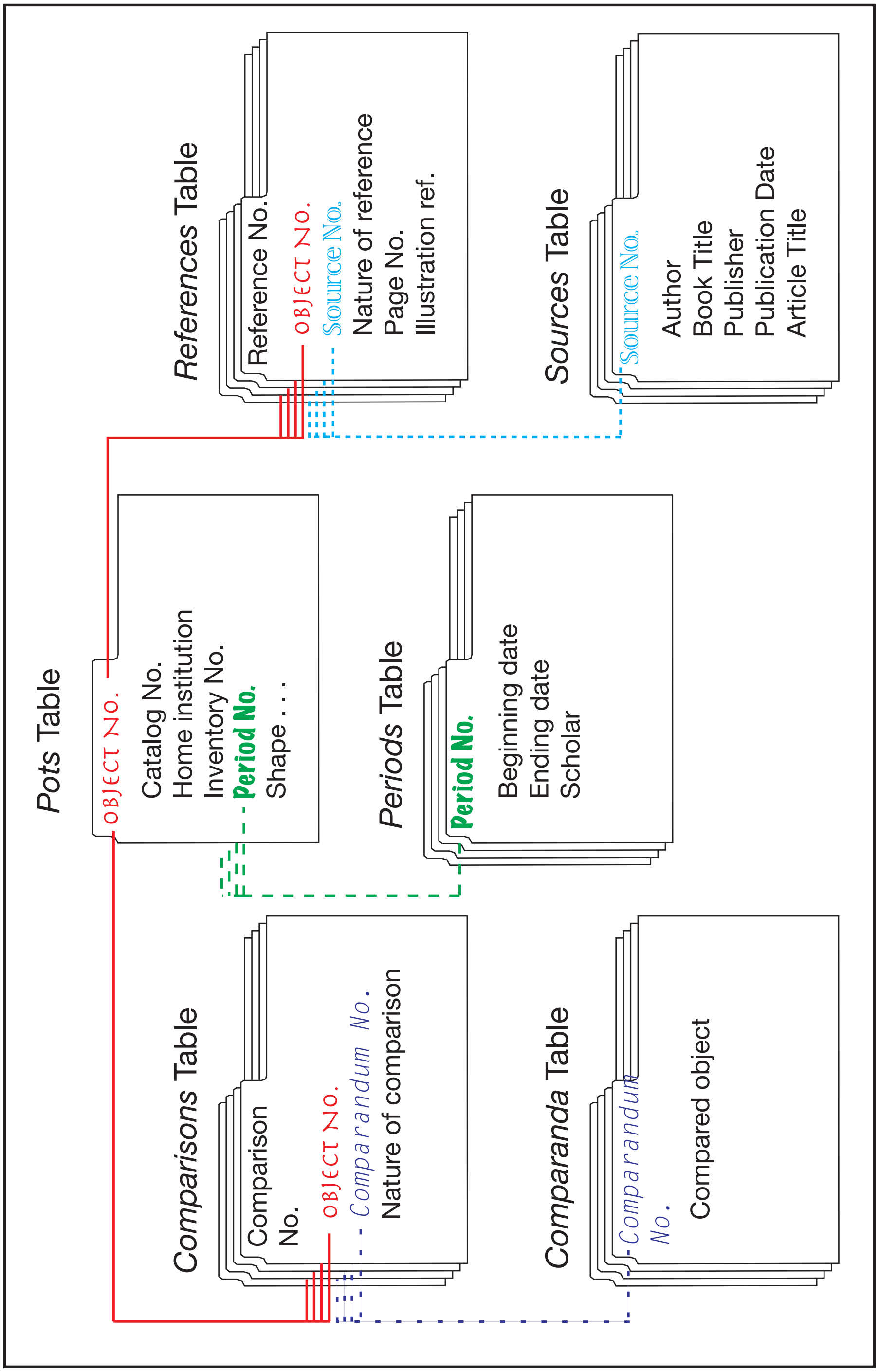


Figure 16

Moche pottery database schema with two many-to-many relationships – one for comparanda and another for sources.

in fact, the same relationship of object-to-object could be listed more than once, each with its own explanation. While that may seem inefficient, it would permit very full and explicit comparisons as well as searches for them. So, for example, it would be a simple matter to select all the comparisons based upon "face decoration for coca-chewing rites" and then gather all the referenced pots in the exhibition **and/or** the comparanda so as to list all objects related to any other object by virtue of this decoration. The database has not only become a far more useful research tool at this level, such searches of the data make it far easier to find and correct errors in data entry because the searches can be so carefully specified.

Using many-to-many relationships is much more complicated than using one-to-many or many-to-one relationships. Most DBMS software does not so easily deal with such relationships, and SQL statements will require building intermediate tables similar to those in the last example of SQL statements to select the information needed from tables related in this way. Nevertheless, this is an excellent way to construct tables because it more accurately and unambiguously reflects the nature of the data.

Similar many-to-many relationships exist for the bibliographic *Sources* Table; many different pots are related to many different sources. Here again, the relationships should be expressed as a many-to-many relationship, requiring another new table. The *Pots* Table must be related to the *Sources* Table via an interme-

Tracking Information

It is crucial that data tables be kept up-to-date at all times. Incorrect or outmoded information is worse than none – and still worse if the user does not know it is incorrect. Good records should at least make available some information about the currency of the data, and there are ways the computer system can be designed to assist with this. Any entry process can include names of data entry personnel and dates of entries to help. Good data entry procedures will require some form of log-in at the beginning of a data session; so the data entry person will identify himself/herself at the beginning of a session. Of course, the computer will "know" the date unless there has been a malfunction. Therefore, adding the person who entered the data and the date of the data entry is trivial – and automatic.

Adding names and dates automatically to a row should not be confused with attaching names and dates surreptitiously. Data entry personnel should not be treated as untrustworthy. If adding this kind of information to the tables were seen as spying, it would suggest a lack of trust. It is therefore much better to make sure everyone knows that such information will be collected as a way of helping staff members to understand the way data are recorded and updated, not as a way to spy on anyone.

Tracking changes to data presents a more difficult problem. If a row in a given table contains the name of the data entry person and the date of the data entry, what does one do when a data item is changed? Change the data entry person and date? Add more columns for editors and edit dates? There is no simple or universal answer to this question, but there is a need to keep track of changes to important data. One solution is – prior to making any changes – to make a copy of the current table row for placement in a "shadow" table that contains only old, subsequently altered data. Then the original entry is updated – with a new editor's name and date of edit (keeping the original data entry person and date always). One could then examine all the entries for a given row in any table to see the history of alterations in that row. Another approach is to require that a memo column have explanations added any time there is a change. Yet another is to limit access for editing to a very small number of people and then not track changes at all – an approach that seems to me to be doomed to fail on projects of even moderate size. Sooner or later the access limitations will break down, and notes about changes are unlikely to be made from the very beginning.

Excavators are unlikely to worry about data tracking. It can seem overkill. However, archaeology is a human endeavor, and we humans tend both to change our minds and to make mistakes with remarkable regularity; therefore, tracking of data entry is important.

diary *References* Table. As with the comparanda, each source in the *Sources* Table will occupy a row in that table, and each source will be connected to the *References* Table by a primary-key-to-foreign-key link. The *References* Table will, in turn, be connected to the *Pots* Table by another primary-key-to-foreign-key link. Also like the comparanda, the information in the Sources Table will be minimal because the details of the reference (as opposed to the source) will be in the *References* Table. Thus, the page or illustration references will be in the *References* Table, not the *Sources* Table, though author, publisher, and other information about the source itself will be in the *Sources* Table. Each relationship specified in the *References* Table will indicate the nature of the specific reference to the related pot; e.g., a comment about the pot in question; a photo of it; or a relevant comment about a pot of the same shape, material, design, etc. The resulting design of the database with two many-to-many relationships is shown in figure 16. As should be clear, the logic of this set of tables is compelling. Each has its own subject matter, and each is carefully circumscribed to omit extraneous information that better fits in another table.

The schema with parent-child relationships between the *Pots* Table and both the *Comparanda* Table and the *Sources* Table required repeating individual comparanda and sources many times. Many pots, after all, referred to the same comparanda and many to the same sources. Therefore, the original tables for sources and comparanda would have had many rows that could hardly have been distinguished from one another, each row containing information about the same source or object but linking that source or object to a different pot from the exhibition. Separating the sources into a better-designed table makes it possible not only to eliminate multiple entries of the same resource but to construct each *Sources* Table row as a proper bibliographic reference, without the page numbers or illustration numbers, which are placed in the *References* Table. Similarly, putting all comparanda into a table – without repeats – permits each

Data Types

When dates in a table are entered as years, how does the computer system deal with the date entries? Using a negative sign to indicate B.C.E. should enable the computer to order any numbers from largest to smallest or smallest to largest with the aid of the sign. However, the computer can only do that if the entries in the date column are taken to be numbers, not alphabetic characters. That is, putting numbers in alphabetic order is not the same as putting them in numeric order. These are in alphabetic order: 10, 100, 11, 110, 12, 120, 13 . . . but not numeric order.

For this problem, database management systems permit users to specify the contents of a column as numeric or alphabetic. The computer can then order items correctly.

Since numbers may be integers (as when years are entered) or more precise numbers requiring decimals, most systems will also permit users to specify the level of precision with which numbers in a column are stored. In any case, it is necessary to specify the nature of a column when constructing a database.

Modern database management systems also require users to specify a column as a date (day, month, year, not the year alone) if its contents are to be treated as dates. Only by specifying the use of a date in a column can the DBMS correctly order the items in that column.

In some cases, database management systems also require the user to specify the maximum length of a text column. That helps the system manage disk/storage space more efficiently; it also separates short, simple, category-based entries from longer discursive ones.

Different database systems have different approaches to the definitions of column content, but virtually all systems will require an explicit distinction between and among text, numbers, and dates at a minimum. Most also supply a yes/no data type, which is the simplest data type since it requires only a single bit to define the data. Memos – very long text columns – may also be defined in database systems.

In addition, most database management systems will permit a column to be defined by a formula – either one using mathematical calculations or one that concatenates text. When formulae are used, the results may be stored or the formula itself may be stored. If the formula is stored rather than the result of the calculation, transferring data to another system may be somewhat problematic, and care must be taken to monitor that transfer.

comparandum to be complete in and of itself as a referenced object. An intermediate table contains the linkage and any explanation needed to make individual comparisons explicit. This schema yields more information about each source and comparandum with less duplication.

The intermediate tables are similar, though not identical. The *Comparisons* Table, which links pots to comparanda, shows only four columns, one for a sequence number so that there can be a primary key, one for the pot number, one for the comparandum number, and one to explain the nature of the comparison. It might seem that the sequence number is unnecessary as a primary key, since the pot number and comparandum number could be combined to serve as the primary key. However, that assumes that only one relationship could exist between two

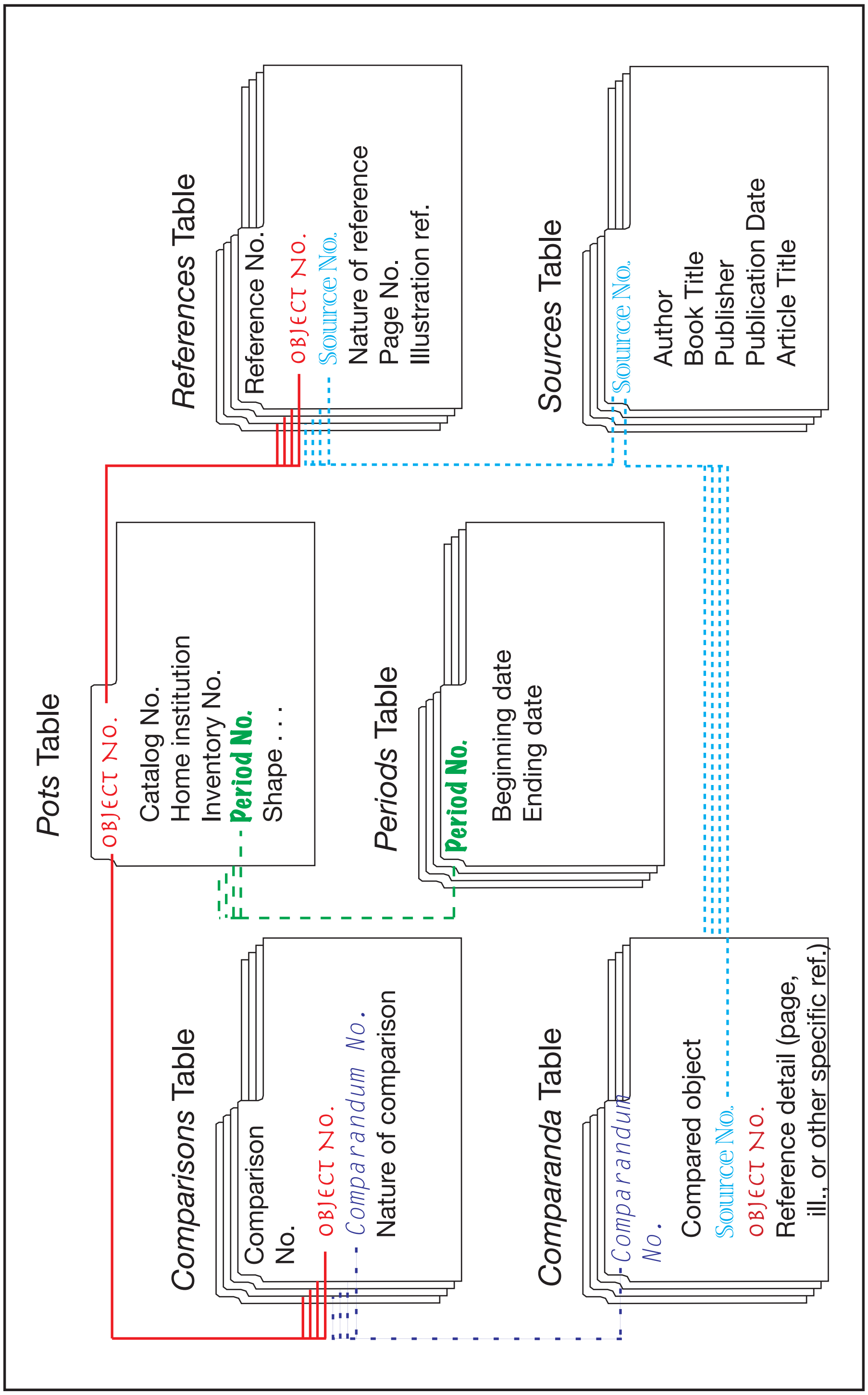


Figure 17
Moche pottery database schema with two many-to-many relationships and comparanda linked to sources.

compared objects. In reality, they might be compared multiple times – as to motif, clay body, shape, size, or findspot. Therefore, the use of a sequence number as a primary key is necessary here.

The *References* Table is similar, with a sequence number for the primary key (reference number), an object number (the primary key from the *Pots* Table used as a foreign key here), the source number (the primary key from the *Sources* Table used as a foreign key), the nature of the reference, and two columns to define the reference more fully: one for page number and one for illustration number. Again, the use of a sequence number as primary key may seem improper, but the same source might refer to the same object in several ways, providing information in one place about a motif and in another about manufacturing techniques.

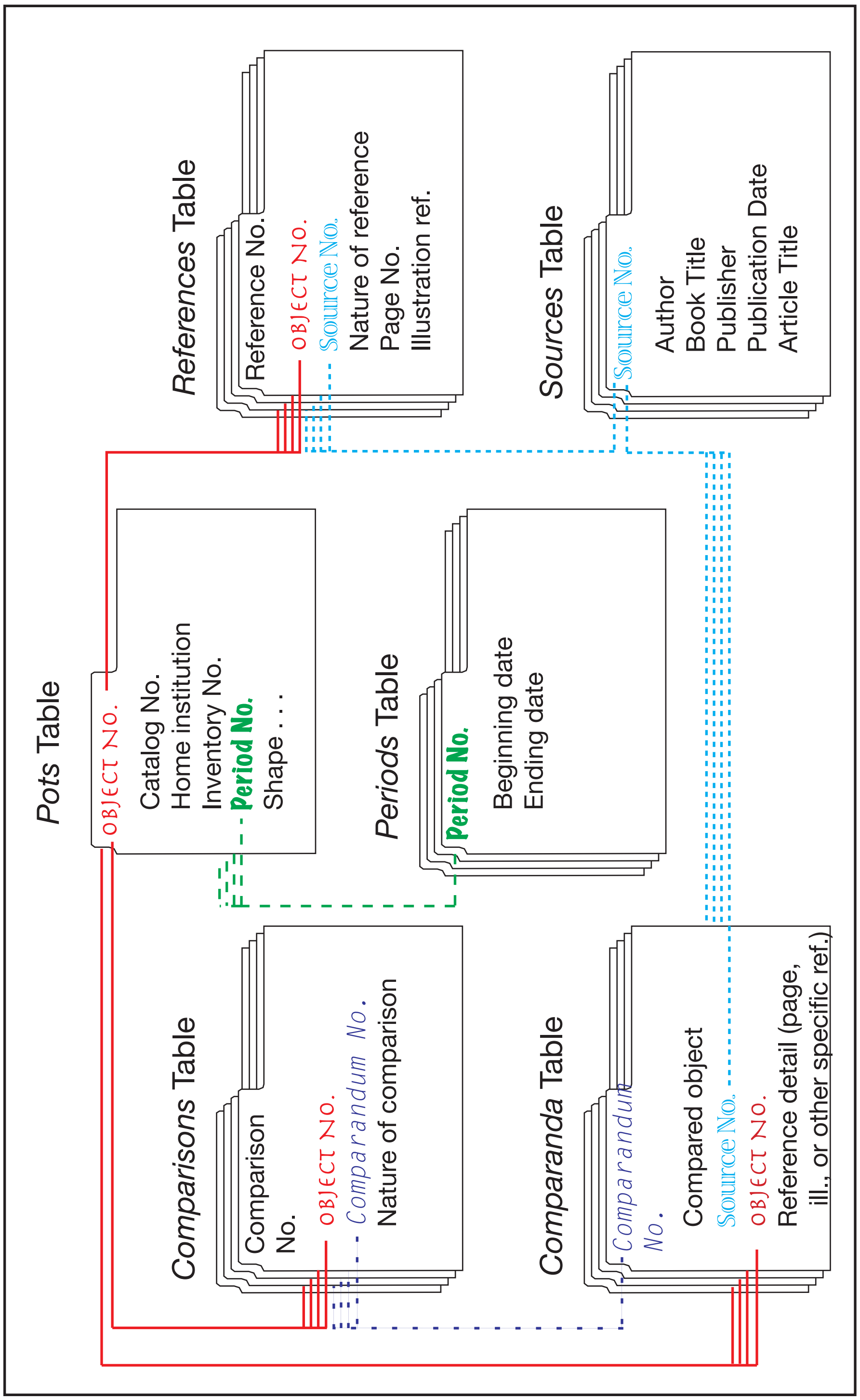


Figure 18
Moche pottery database schema with comparanda linked to exhibition objects as well as sources.

The schema shown in figure 15 is now much more complicated than the first one with child tables shown in figure 4, As the schema has become more complex, however, the individual tables have become simpler, more confined. Each table has more limited functions and more limited data types. For the many-to-many relationships we have tables that contain little more than linkage information, and the *Sources* Table now contains only bibliographic information. All the tables are now easier to manage, and some tables, e.g., the *Sources* Table, can be created separately from the others.

Placing all bibliographic resources into the *Sources* Table provides another benefit via yet another relationship. The *Comparanda* Table will include objects from some of the bibliographic resources listed in the *Sources* Table, for instance the pot identified as "Lavelle, 1989: Pl. 159." This reference is to an object for comparison that is not part of the exhibit but is found in a bibliographic resource. Such a bibliographic resource should be more specific as to the reference, and it could be if the data referred to the *Sources* Table via a link to that table. The link can be made, of course, via the primary key, source number, in the *Sources* Table. The source number can be used as a foreign key in the *Comparanda* Table to link the object for comparison to the bibliographic resource in which it can be found. The page or illustration number would remain in the *Comparanda* Table, but the full bibliographic citation would be in the *Sources* Table. The new relationship between the *Comparanda* Table and the *Sources* Table is shown in figure 17, along with the required changes to the *Comparanda* Table. This is little different from the use of a bibliography in a publication except that the access system here is electronic; in both cases the bibliographic resource is fully defined only once but can be referenced via an unambiguous link – in a book the short form of the reference name, in the data schema the source number.

Many readers will have anticipated the next complication. As the comparanda coming from bibliographic resources may be more fully specified by relating the *Comparanda* Table to the *Sources* Table, so objects used as comparanda that are from the exhibit itself may be more fully specified by a relationship from the *Comparanda* Table to the *Pots* Table. Using object number from the *Pots* Table as the foreign key in the *Comparanda* Table makes that possible. The new design is shown in figure 18. (Note that the column called compared object remains. Not all comparanda will be from bibliographic sources or the exhibit itself, and there must be a way to reference them.)

This design is far more complex than the design with which we began, but it is far better. There are no longer duplicate entries of any sort, only duplicate pointers to a single entry. There are no longer multiple entries for a single column in a single row; there are no longer multiple entries for the actual dates implied by period names.

It may seem that a great deal of complexity has been introduced into this discussion of database design with little apparent purpose. However, there are most definitely reasons for adding the complexity. Many questions can be answered precisely because of care taken in the design of the database. For instance, we could quickly find those pots without comparanda or bibliographic references to know that more information about those examples should be added, and we could determine which pots refer to other pots in the exhibit to consider re-arranging the pots. We could order the pots by height, by diameter, or by date. (Note that none of those possibilities would exist if the columns had not been defined as numbers.) We could find out which examples are actually in the museum collection, which are to be in a specific case, which are referred to in basic sources, ... Virtually any question one can imagine could be quickly and easily answered, including questions that combine information from multiple tables, questions such as "What are the pots in the exhibit that fall in a specific date range and are in the shape of a dog but lack a comparandum?" That's the whole point of building such a database and making it so complex!

The SQL statement to find all pots thought to be current between 400 and 600, in the shape of a dog, and missing a comparandum:

SELECT Pots.ObjectNo., Pots.CatalogNo., Periods.BeginningDate., Periods.EndingDate FROM Pots, Periods WHERE Pots.PeriodNo.=Periods.PeriodNo. AND Shape.Pots LIKE '%dog%' AND Periods.BeginningDate <600 AND Periods.EndingDate >400 INTO Temp -- and then this:

LEFT SELECT ObjectNo., CatalogNo. FROM Temp, ComparisonNo. FROM Comparisons WHERE Temp.ObjectNo. = Comparisons.ObjectNo.

The first SQL statement finds all the pots with the proper period of currency and the correct shape, putting them into rows in the new table called *Temp* Table, whether there are comparanda or not. A second SQL command was needed to finish: In the last command the term *LEFT* instructed the system to show all entries in the first-named (left) table, whether there are related comparisons or not. Therefore, the second process will yield all the examples from *Temp* Table, with nothing in the column for ComparisonNo. for those examples lacking a comparandum. The user could either use another *SELECT* command to get only the examples with no entry in that column or simply look at the new selection set and identify those examples without a comparandum. The latter would be acceptable for a small data set, but the former would probably be required for a larger one. (Note that a pot might have many comparanda but the command sequence used here would only retrieve the first one.)

Looking at the Gordion fibulae example again, we have many similar opportunities for answering questions, e.g., fibulae of a given date, without a mate, over 4 cm. long, and found in the bed near the right elbow. In addition, we could easily determine where in the tomb specific styles were most common, which style (and date) was most common overall, and so on.

In the case of the excavation system we prepared, the examples are less numerous because we only scratched the surface of the design, omitting all non-ceramic lots for instance. As you can predict, however, a fully operational system would permit all objects from a lot or a locus to be seen together – or learning which lots and loci contain pottery of a given period. Such views into the data would assist greatly with excavators' analyses.

Regardless of whether one is starting with objects in a museum or objects from an excavation or survey, the design is intended to lead to answers of questions posed by excavators and subsequent users of the data. One cannot easily know all questions that will ultimately be asked of a database; so it may not be possible to anticipate all and to design the system to answer all questions. Many of those questions should be obvious, though, and the potential to answer them should be designed into the system. Even when the questions are not obvious, making sure that data categories are inclusive and thoroughly broken down into their smallest components will provide for the maximum in ultimate utility of the data. When problems arise and change is required, change need not be feared. If the job has been done right from the beginning, there should be far less need to worry about loss of data in the process of changing file structures.

As a colleague once said, "You never get it right the first time." Experience suggests that such a pessimistic appraisal is correct. Therefore, do not hesitate to make change when necessary. Getting it right the last time is what matters. But be aware that sloppy design at the beginning can be fatal. If data are lumped together in ways that do not permit later separation or if some observations are not recorded, a full recovery may be impossible. Less critical errors may only start the process badly and make improving it more difficult.

You should now have enough background to understand a database constructed by someone else, to see how the tables are related, what columns have been defined, and what information is recorded in what ways. You should be able to see what questions the database has been designed to answer (though you may not know how to ask those questions using the particular DBMS), and you should be able to spot some flaws in the design such as using multiple columns in a table for the same information instead of a separate child table.

You should also be attuned to the issue of tracking data entry. Although methods for tracking data entry and editing processes have not been fully explained here, the principles have been discussed. You may not know exactly how to record tracking information, but you should be able to look at an existing database to see the tracking methods – and to evaluate them.

Documenting Databases

While you may have the tools to look at a database and understand it, you will rarely be able to do that with only the database management software at your disposal. Unless you happen to be very familiar with the particular DBMS, you will not be able find the necessary information without the help of the people who created the database. That help should exist in the form of documentation created in the process of designing the system. The documentation is necessary for the system creators so that they can make necessary changes or additions over the life of the project and so that they can be sure that the implications of their design are made explicit. The documentation is also necessary in case there are changes in computer personnel requiring that new people work on the system. Of course, the documentation is also necessary if anyone is to try to understand the system either

Lookup Tables & Controlling Input Options

There is another kind of table that has not been discussed, the lookup table. It is exactly what the name would suggest, a table containing data that may be needed – and looked up when necessary. Such tables are often used in data entry routines to provide all possible entries for a specific column. For instance, one might provide a list of pottery styles when the style column must be filled in. The list would appear on screen when needed so that a selection could be made. Not only does that mean that the permitted data entries are shown, it eliminates typing errors when the chosen entry goes directly into the data table. Such a list of potential entries can be in its own table, a lookup table.

With a lookup table, the data entry procedure can be designed to force the use of only the listed choices or to permit free entry as an added option.

A lookup table is not part of the relational data set; its contents are simply used as prepared entries to be fitted into a specific column at the appropriate time. One could use a parent table instead, entering only a foreign key in the child table to refer to the primary key in the parent table. Many would argue that the use of a foreign key and a parent table is better than using a lookup table. That permits later adjustment of the term – in only one place – to change it everywhere.

Lookup tables and relational tables with term lists can serve very similar purposes, and the choice between the two often comes down to intangibles. Does the designer want a code or a term? Will the term change, or is it likely to be stable? Might the table often be used in isolation, separate from the entire database, so that the term rather than the code is really needed? These considerations will determine whether one uses a foreign key and a related table or a lookup table; users will see no differences.

Both lookup tables and related tables can provide excellent ways to be sure that all entries in a given column meet the demands of the excavator; however, both share some risks. The data entry personnel may become so tied to specific terms that, faced with an anomalous find, correct entry is impossible. In such cases a lookup table can be made more complex to solve the problem; I once created a complex lookup table that gave users prepared entries but allowed them to add new entries on their own and even to alter the prepared entries. That required some complex behind-the-scenes manipulation of the lookup table, but it provided both the standard list of pottery styles and the option of adding to them or altering them. I do not believe that could have been accomplished with a related table.

to evaluate it or to replicate it with new software. Finally, as will be discussed more fully in the archiving chapter, the documentation must accompany the data set when it is deposited in an archival repository.

You will already have in mind some of the needs for documentation, but a list of requirements is provided here to make sure that no requirements are missed.

1. The entire database: The scope of the entire data set must be explicitly defined so that a user will know clearly what was intended – and what was not. The software used should also be defined (including all versions and the respective dates of use), as should the file format(s) in which the database has been saved. Though the file format is implied by the software, it should be made explicit, and any other file formats used should also be stated, as should changes in structure caused by updating the software. (New versions of software often use new file formats.)

2. Tables: Every table must be named and its general contents described.

3. Columns: Each column in each table must have a name provided (preferably a name that is meaningful but not too long) and the content specified – including data type (number, text, date, etc.), rules applied to data entry (e.g., specific requirements for capitalization or spelling, use of terms for unknown or unavailable information), limits on data entered (greater-than and/or less-than limits, look-up tables, lists of acceptable terms, fields from related tables used for verification, etc.), whether the column is a key (indexed) one, whether it is a primary key, whether it is unique, whether it is a foreign key. In addition, links to other files via any column must be made explicit. Thus, a primary or foreign key column should also be defined as providing a one-to-one, many-to-one, or many-to-many link to another table, complete with the name of the other table(s), the name(s) of the column(s) in the other table(s), and the purpose(s) of the link(s). Finally, the documentation should include

Limiting Data Entry Choices

Using a lookup table or a related table to limit data entry choices gives the excavation director considerable control; allowing unconstrained data entry may provide the potential for more nuanced entries – and more errors.

There is another consideration. Using a table to circumscribe entry choices may limit data entry before the project director can be sure what entries ought to be permitted and what entries prohibited. One approach to that problem is to leave data entry open for the first season of a project and then to examine the data entries for each field that might have a limited number of acceptable entries, cull the unacceptable, select the appropriate term or phrase where more than one has been used to carry the same meaning, adjust all the entries to match the new list, and then add a lookup table for the future. Even in this careful process, however, data entry personnel in subsequent seasons should be permitted to add new terms (that will be examined at the end of the season in the same careful process) when the list does not suffice. Note that this process assumes considerable work with the data tables in the off season and an on-going commitment to database refinement.

There is another hidden problem with lookup or related tables that supply prescribed terms – and with data entry generally. Data users always want the most precise determination possible, but the data entry person may not be able to be so precise. If the subject matter is Late Bronze Age pottery from the Aegean and there are fine-ware sherds, then Late Helladic III C: 2 is a better designation for a user than LH III C, which is better than LH III, which is better than LH. That being said, the data entry personnel may not have the training, the confidence, or the time to make the fine discriminations necessary to label a sherd more precisely than LH or LH III. This is one of those areas that cries out for open, honest descriptions of data entry, personnel, and control processes. No excavator should expect precise specifications on the spot from trench masters who cannot be experts in all materials, and no user should expect such fine distinctions either. However, someone will ultimately make those distinctions at a later date. The project directors must plan to integrate the later, more precise recording by specialists in such a way that it will become a part of the excavation database. Then the more precise terms will not only be available but reliable.

information about the ways in which the requirements for individual columns have been enforced. For instance, were data entry procedures designed to limit entries so that unwanted entries could be denied? Any checks to be sure that data entry was properly carried out should also be specified. (It is fairly easy to sort a table by a given column or to conduct well-considered searches to check for some incorrect data entries.)

Any column that contains a measurement – linear, volumetric, weight, or any other measurement – should be a number only, and the unit of measurement should be made clear in the documentation. That unit should also be clear on all forms for data input or for reporting. Even units of measurement that seem absolutely obvious must be specified.

4. Relationships: Every relationship between tables should be shown (in addition to explanations in column definitions). This is most easily done with diagrams such as the ones used for the Moche, Gordion, and sample excavation tables in various figures above, with text to explain the natures of the relationships.

5. Prepared entry procedures: All data entry procedures should be described in reasonable detail. The aim here is to describe the way data were entered, the screens used, the paper forms from which the data came (when appropriate), the ways limits on data entry worked in practice, and the ways data entry personnel were known to have erred, if any. For example, it is important to know whether tables or formulaic routines used to limit data entry choices permitted the data entry personnel to over-ride the limits and whether free text entries were checked or edited.

6. Prepared searches: Virtually all databases will include certain prepared searches and data selections. In general, those will be the searches and the selections of data thought to be valuable by the project personnel – all the pottery from a destruction layer, for instance. Information defining such searches and data selections must be provided.

7. Added tables: If there are added data tables from experts who have studied particular object types or from commercial sources, those tables must be completely defined as all others have been. In addition, information about the construction of those external tables should be provided, if possible by the experts who constructed them. Finally, the ways they have been related to the other tables in the data set should be fully and carefully defined. (Of course, if they have not been integrated into the remaining data tables, that should be specified as well.)

8. Personnel: All those involved in organizing the data, constructing the tables, making the forms, and so on should be identified so that questions regarding the preparations of the data tables may be directed to the proper individuals. While data entry personnel will have been identified individually in the data tables, the general entry procedures should also be specified in the documentation so that a user can get a good sense of the flow of information within the project.

While it may have seemed earlier that understanding a data set would be an extremely difficult job without substantial experience with the particular DBMS software used and a good deal of time to study the particular database, the documentation just described should make the job much easier. With an overview, lists of all

Changing Database Organization

Changing the structure of a database is so easy that doing so can seem to be safe and simple when it may be neither. In fact, any change to a database may have quite unexpected results and should be undertaken with considerable care. Therefore, changes should be tried on copies of the real data sets and, to the extent possible, fully tested there.

Even changes that seem truly benign and have been well tested may carry eventual surprises. Therefore, any change to a database already in use should be accompanied by a back-up process that creates a copy of all data – and the full system – as it exists before the change. That back-up copy of the database should then be kept until there is no remaining doubt about the stability of the altered version of the system.

tables and all columns therein, understanding of the data entry systems, and an understanding of the relationships between and among files, it should be possible to understand a database and to evaluate it. Absent the documentation, on the contrary, it would be virtually impossible. Thus, the documentation is essential.

Building a Database

It is beyond the scope of this book to describe the process of building a database because the DBMS software chosen will determine the processes. Nevertheless, there are many important guideline that apply to any DBMS.

First and foremost, the database system should be designed in cooperation with the project director **before** the project has begun. It is crucial that the database designer and the project director work together at the beginning so that all the conceptual issues that matter – hierarchies, naming systems, control of unit names,

Adding a Column to a Table – Avoiding Empty Columns

Adding a new data column to a table has hidden difficulties. If the column represents a new observation, for instance, something to be recorded about lots not yet excavated that has not been recorded about lots already excavated, all lots already recorded will have no entry in the new column. What will someone looking at the table make of the empty column?

This is a very important matter because of the potential implications of an empty column. A truly empty column is called a null column. A null column is different from one in which there is a space (a text column) or a zero (numeric column). Databases ignore nulls in many calculations, but they will not ignore zero or space entries. For instance, a null column will be ignored in a calculation of the average value for all entries in the column; a zero, however, will be included and will affect the calculation. (The average of fifteen numbers that total 1500 is 100. If there were five zero entries that should have been null entries, the number of samples should have been be ten instead of fifteen and the average 150 instead of 100.)

Although the null entry may be ignored in calculations, it is inherently ambiguous. What does it mean when you inspect a data row and see nothing in a column? Do you think it means zero or uncertain, for instance? Because of that ambiguity many database designers reject the use of null entries altogether, arguing that all columns should be filled and that there are effective ways to mark individual entries so that the meaning is clear – and so that they will be ignored in calculations. For example, negative numbers may be used in a column to indicate an unknown quantity or impossibly large or small numbers may be used to the same purpose. Calculations can then be performed only on a sub-set of the whole that excludes the unwanted entries.

In text fields an entry like "N.A.," for "not applicable" is greatly to be preferred to a null entry. With either text or numbers, one must plan in advance what entries will be used for all circumstances when actual data are not available. (Of course, this implies that a yes/no column must be used judiciously in any database, since there are no possibilities for an unknown entry.)

To return to the original point, what is to be done when a new column is added to a table and the existing rows have no entries in that column? The existing rows should be filled with an appropriate entry such as unknown or not recorded in the new column. (A copy of the data files should also be made before adding the column, and that copy should become a permanent part of the data set, not to be discarded until everyone is satisfied that the new file is unambiguous.) In addition, the documentation about the database should thoroughly explain the fact that all entries made before a certain date did not include the column in question and were therefore filled with "unknown" or "not recorded" to indicate missing data. This is a good example of the importance of recording dates of entries. With dates of recording included, the user of a data set is much better prepared to understand issues such as this one. This is also a good example of the importance of documentation. A user who knows a great deal about a data set – its preparation, history, and so on – is far better able to make effective use of the information.

Password Protection:

Many computer specialists are nearly paranoid about the use of passwords to guard access to data. Their experience suggests that their concerns are not only legitimate but that great care about this is absolutely crucial. My experience has not been the same. Password protection is indeed important because it allows the database designer to protect data effectively. For example, a certain password may be required to do any data editing. But overuse of password protection so as to make it seem that staff members are not trusted is a cure that may kill the patient.

Nobody on an excavation is there to make mistakes or intentionally to cause problems. To indicate distrust with zealous and burdensome password schemes thus seems counterproductive to me. On the other hand, using passwords to make users more comfortable – to let them know that they may be able, for instance, to err with a data entry but to do no more harm – often makes those working on the data more comfortable.

The use of password protection is obviously a somewhat personal choice. At the least, there should be some forms of password protection to be sure that simply examining data in the system will not expose the data to contamination. That need not mean that the passwords are secret, only that certain ones are used for certain procedures.

and so on – are well understood and properly debated. The archaeologist in charge of the database system should be seen as a crucial member of the team from the beginning of the project, and he or she must be able to assist with the overall project design. It is at this early phase that logical inconsistencies should be found and eliminated. In the real world, this will rarely happen; therefore, at the very least, the database designer must be involved early enough in the process to have the time to understand processes and procedures and to ask appropriate questions **before** the system goes into the field. (Mark Twain: "It ain't what you don't know that gets you into trouble. It's what you know for sure that just ain't so.")

Paper forms for data should be designed with the computer forms in mind and vice versa. Nobody should ever be faced with a computer screen that demands data entry from a differently-arranged paper form. It is not only time-consuming to have to search in the lower-right-hand corner of the paper form for data that need to be entered in the middle of a computer screen; it leads to errors. The paper and computer-screen forms should be as nearly alike as possible.

This may seem difficult, since the paper form will almost certainly contain information that belongs in several different data tables. However, what appears on screen to a user need not reflect – almost certainly will not reflect – the way the data are organized "under the hood" in the database. Look back at figures 8 and 9 and remind yourself that there are three tables represented in those screen forms. That is not apparent to a user.

There are issues with computer forms that are different from those affecting paper and vice versa. A form on a computer screens can scroll such that the "page" is virtually infinite in length or width. Paper space is more limited – suggesting that the screen should not be scrolled but should be "paged" when pages must be turned so that the screen and the paper are more similar. Paper forms have blanks for information that are much more obvious that the typical computer-screen form. Care should be taken with the design of the computer-screen form to make clear where the data goes and to control the order of data entry. Type fonts and sizes will be important; colors or shading of data boxes or labels may be a good idea, but one must be judicious with the use of color, lest the user be distracted. (Subtle differences in color should also be avoided since they may not show correctly on all computer screens or in all lighting conditions – and partly color-blind users may miss subtle color differences.)

The equipment to be used on a project must be considered when forms are designed. If the forms are to be used on relatively low-resolution monitors, they must be planned for those monitors. When monitors with different levels of resolution are to be used, forms should be designed for those with the lowest resolution or differently designed to fit the resolution of each monitor.

Data entry should take advantage of what the computer already "knows." For instance, the date should never have to be entered by a user, since the computer already has that information in its memory – though there should be a start-up check to be sure that the computer battery that may be required for keeping the date in memory has not died, making the computer-supplied date inaccurate. (The inaccurate date will be so far off that a check is easy and can be carried out by the computer itself.) Similarly, if the data entry person logs in at the beginning of a data entry session, the computer can enter the name anywhere it is needed.

There will often be data entry procedures that yield a natural progression of information such that certain categories can be predicted. It may be a good idea to pre-enter some of the data that can be predicted – but only if the process requires a positive confirmation of the predictions. It is perilously easy to work through multiple data items without really paying attention.

Data checks from lookup tables or related tables are very useful, but there are other ways to check data, and they should be used whenever possible. For instance, field records may show all the sherds collected from a given survey area or locus, and the user may be asked for a total as well. It would be tempting to ignore the total, letting the computer calculate it. However, requiring that the total be entered and then checking it against the computer's total is an excellent way to avoid an unintended error. If loci are entered with positional information, including top and bottom elevations, it is very simple to let the computer check that the elevation of the top of the locus is really higher than

Small Projects

The databases described here are very complex. They require software and project personnel that may be beyond the budget and time limits available for small projects. The principles can be used, however, by very small projects, and a similar level of complexity and sophistication in the data, if not the data access systems, can be created.

For those involved in smaller projects, the aim is to conceive of the data in the same inter-related and complex forms that have been discussed here, to plan to record the data in such complex structures, but to forsake a unified system for data entry and retrieval. That is, a small project may need many tables, but if each is treated simply as a spreadsheet file, using Excel or another spreadsheet program, the actual data can be as carefully and thoughtfully stored as in any database. Data entry or retrieval will require switching between and among various tables (spreadsheets) and combining data will also require determining how to unite data from multiple spreadsheets. Nevertheless, good data organization and storage should be so clear that, even on a very small project, the added difficulties would be minimal. Approaching the data – not a database system but the data – in sophisticated ways makes it possible for a small project to grow with minimal problems in converting data into more sophisticated systems and to archive data in ways that allow others to use the data in sophisticated ways. Spreadsheet programs are not database management systems, but they can store data as well – if the underlying organization is good (and the data can later be exported to a full relational database with relative ease).

It must also be said that, while data entry is not easily assisted/monitored with a spreadsheet, using formulae in the spreadsheet to find data errors is relatively easy. Some such procedures should be used on a regular basis. Although it may seem that smaller projects will suffer fewer data entry problems because there will be no volunteers, undergraduate students, or other less-invested data entry personnel, the simple fact is that some people are not particularly good at dull, repetitive tasks. In addition, smaller projects often require a fewer people to work longer hours, yielding sleep-deprived scholars. Routine tasks such as data entry are among those most in need of alert personnel. Data need to be checked.

that of the bottom.

There are many such checks of data entries, and the number employed will have as much to do with the nature of the data entry personnel as anything. If all the data entry personnel are well-qualified archaeologists, some checks might be omitted, but if many are undergraduate students or volunteers, it may be appropriate to include every possible check. One thing is absolutely certain. Once data have been entered incorrectly, it can be enormously difficult to find and correct the errors, far more difficult than to prevent them in the first instance. And, given the way computer data are treated, anything on the screen will often be taken as the gospel, even if it seems obviously absurd.

All the processes of preparing data entry forms and procedures should be punctuated with regular experiments involving project personnel using the data entry procedures that have been developed. Real users will not necessarily react as the designer expects and may – not unreasonably – demand more sophisticated procedures than expected. In the final analysis, it is their ability to use the forms and procedures that matters most. Extra time spent on system design will have been very well spent if it yields quicker or more accurate data entry.

No matter how carefully the designer has tried to control data entry procedures, the data must be checked regularly to make sure that incorrect entries are not somehow slipping through. Simply looking at data tables and ordering them by various columns will often help display errors. Of course, the designer will have a good idea what kinds of errors are likely, and knowing what kinds of errors are likely will also help with finding ways to spot them.

As the data-entry and data-edit procedures are being designed, the question of process tracking must be discussed and some decisions made. A short discussion of tracking was included above, but the crucial planning issue is not how to track information about who recorded what information when; the crucial issue is how much information of this type to record. While some would argue that every data-entry or data-edit process should be tracked, others would take a much less stringent approach and record little such information. In general this tracking information is neither difficult to build into the system nor particularly difficult to retain; therefore, it seems to me to be an important part of the total data set.

In addition to the designs of various data-entry and data-edit procedures,

Problems with Sequencing Data

In the case of pottery style such as Late Helladic, how does one order the styles without some artificial rubric? Early, Middle, and Late Helladic pottery (EH, MH, and LH will not alphabetize in the proper order; neither will Roman Numerals of styles within the early, middle, late sequence, e.g., LH I, LH II, LH III.

There are some fairly simple ways to order such styles. One is to use two fields for anything that will not otherwise order correctly. One contains a numeral for the sake of ordering, and the other contains the actual text; thus LH III might be the text entry and 30 the numeric one. This violates one of the rules of database design to be discussed below, and we saw the inefficiency of such an approach with the Moche pottery when dates were stored for individual pots.

A better approach would be to use a related table with three columns, one with the actual style name and the other two with the beginning and ending dates for the style. The style name would be a foreign key, containing LH III or some similar entry, so that the dates could be connected to any individual example. Any listing of pottery could then be ordered by the date with only the text value showing. If this process is used, it is important to retain the actual name of the style in the pottery table, not an artificial code. Why? Because there is less potential for data loss if the pottery table has the crucial information, making it more nearly self-sufficient.

Another approach to this problem would be the use of some programming in the form of a macro or script to permit proper ordering of entries. The macro could assign a number to a table row based on the pottery style. This is not as desirable a way to solve the problem because it makes the solution harder for someone to change without being intimately familiar with the database and its programming language.

a database system for an archaeological project will include various searches and data-display processes. It is crucial that on-screen data not be subject to change in such circumstances. Nobody should be able to change on-screen data without both intent and authorization. Therefore, the presentation of data for examination should be done via forms that do not permit the user to alter what is on screen.

The most important process for data protection is frequent data back-up as well as backing up the underlying database system. If all tables are regularly copied onto some media that cannot be altered (CDs or DVDs) and if the software is similarly backed-up, the potential for serious harm is greatly reduced. Since it is extremely unlikely that any harm will be surreptitious and intentional, back-up precautions – and procedures that permit reproducing entries from any particular back-up forward – should be sufficient, if and only if the procedures are followed and carefully recorded. Nothing is less effective than carefully planned procedures that are ignored – like my father's state-of-the-art hearing aids hard at work in his desk drawer.

Table Design

Individual tables should obviously have a certain logic in terms of what they do and do not include. However, since relationships with other tables may make it desirable to use multiple tables for what seems to be a single logical set of data, tables may also be designed other issues in mind. Particularly if some people may want to use one selection of data about a given object or group of objects, it may be prudent to use two related tables to store what seems to be a unified set of information. A simple example is be the *Personnel* Table; some personnel matters are likely to be somewhat sensitive and therefore not open to all; a simple way to protect such data is to put the sensitive material in its own file with separate access procedures.

In addition, it may be desirable to separate observations from analytic conclusions. That is, specifying pottery styles from a given context is an observation process. Deciding that the pottery found and analyzed makes the date of its context a specific chronological period is an analytic conclusion. If such analytic conclusions can be separated from the data tables with observations, it becomes possible to permit

Export Tables

The design of a good database involves many tables. It is likely, however, that those tables will not be the ones that the specialists involved in an excavation or survey want to carry away with them between seasons. They will prefer the multi-table screen presentations of data they have grown accustomed to on site – presentations that are only possible for them in their own offices if they own and use the DBMS software used by the project. If the specialists do not have the same software, it is helpful to be able to provide for them a table or two with just the data needed for between-seasons study. Such individualized tables cannot be built into the system in advance because the specialists may not know what they want in advance. However, the project database manager should be prepared to provide such tables (in a form such as tab-delimited ASCII that can be used by virtually all database or spreadsheet software) to anyone on the team. Given the specialized uses to which such tables will be put, these will rarely be tables that have been properly normalized, though they will have been created from normalized tables.

Some examples:

1. All finds in a locus-by-locus listing so stratigraphy can be studied and potentially problematic loci discovered.
2. All examples of specific pottery styles (according to preliminary analysis) for a specialist to use as a starting place.
3. A list of catalog objects not yet photographed.
4. A list of all items sent off to any lab.

If any of these individually-tailored tables may be used to add or edit data between seasons – creating or changing data that belong in the project data set – the database designer must be extremely careful to provide mechanisms for data entry and edit. (In such cases, the edits should be prepared in separate tables that can be examined and loaded into the primary tables by the DBMS specialist before the next season.) Those mechanisms must permit the data to be properly checked and then uploaded into the data set maintained for the project. Such a process can be dangerous; so it should be carried out on a copy of the data set and thoroughly checked before the real data are exposed to changes.

multiple competing sets of analytic conclusions to exist at the same time and to be considered and compared. If two members of a team, for example, assign various loci to different chronological periods and their determinations are stored in a separate table along with their names (and explanations), it would be possible to compare the stratigraphy according to John Jones to the stratigraphy according to Susan Smith. This distinction between what has been observed and what has been concluded is not always an easy one to make, and most scholars prefer to ignore the issue. Nevertheless, a good database system should, where possible, provide some way to handle differing interpretations.

The same distinction between observations and interpretations can be maintained when dealing with architectural units. Rooms A, B, C, and D may be assigned to Building 1 by Susan Smith but C and D may be assigned to Building 2 by John Jones. Those differing interpretations should be part of the database. If, at the end of the project, agreement has been reached and the project director wishes to present only the standard view, the analytic conclusions included in the database can be easily reduced to those of the director, but it would be much more useful to future scholarship to let the disagreements remain available in the database.

Column Design

Many issues of column design have already been discussed in passing. The avoidance of null columns, for instance, and the need for real numbers as dates have been mentioned. In addition it is important to bear in mind some problems with using either non-alphabetic characters or abbreviations. Alphabetizing will not work as expected with either, and ambiguity is a real danger with both. Abbreviations can obviously be misconstrued, but characters like <, >, ^, and * may also be misunderstood easily. They should be avoided.

Problems can also appear when using Roman numerals. Since they represent numbers, it should be possible to put items in numeric order, but, of course, that will not work because the column with a Roman numeral will be considered text and ordered alphabetically. Similarly, using mixed numbers and letters in a single column will cause problems with ordering. In short, every column should be carefully analyzed for the use of its content before the nature of that content is determined; in some cases, it will be necessary to break a single column into two or more in order to accomplish the goals of the database or to use a related table. (Multiple columns can easily be

BLObs

Yes, there are such things in databases as BLObs – binary large objects. Simply put, these are things that would otherwise be thought of as independent computer files that have been imbedded into a database; images are probably the most common. It is possible to import an image (or a CAD model or a music file) into a data table as a column. This is remarkably convenient, but it is also remarkably short-sighted for any data set that is likely to be used for a long period of time, as an archaeological database is.

When a BLOb is included as a column, it must be imported and embedded in the format of the particular file type – JPG or GIF in the cases of an image, DWG or DXF for a drawing, and so on – and then exported as that same file type for use or display, though some DBMSs recognize and display certain BLObs that are in common formats without an added program. Thus, a BLOb is a file within a file, and issues of file formats, which will be discussed at length in the chapter about data archiving, become enormously more complicated – with little purpose. It seems unnecessary to store disparate types of data in a single file. A pointer to a file (by directory and file name, for instance) should be sufficient and involve no file format concerns. BLObs should be avoided.

Some DBMSs, Microsoft's Access for example, permit links to image files to be stored simply by pointing to the proper file during data entry. The database system takes care of recording the file name and location. If such a system is used, one should make certain what the implications are for file storage. Can the resulting database be shipped to and used by another scholar? Can the files be found readily if the data files must be exported to some other format? It is critical to use a system that is designed to work in a transparent way and that permits full documentation of the system of file linkage.

combined for display.)

For those working in countries where non-Roman alphabets are in use, local characters may present problems. Even in Western Europe, accents can present problems. Until Unicode has become truly standard, this problem will continue to plague developers, often requiring them to work completely in the local script or only with Latin characters.

Another important issue for column design is the use of an explanation column. If columns are well-designed for ordering and for clarity, they may imply more certainty than is warranted; so it is often very desirable to have a companion column that can explain, temper, or modify a seemingly simple data item. That column might indicate a date range, for instance, to accompany a specific date; it might simply add "approximate," "uncertain," or "contested by John Jones." The point is to permit the data to be clear and useful without becoming too simple.

Table Joins

Modern database management systems provide many ways to use information from multiple tables together. As a consequence, older terminology has sometimes fallen into disuse. Nevertheless, the specific terms used to describe the ways tables can be used together or *joined* represent important concepts, even if the term is less often used today. In addition, table joins are used extensively in SQL statements; so some discussion of table joins is valuable. No matter the nature of a table join, the basic notion is of combining the data from multiple tables via primary-key-to-foreign-key links.

First, it should be said that in older systems the results of table joins were regularly new tables that presented information gleaned from multiple starting tables and combined into one table. (That new table would fail the normalization requirements discussed in the next section, but it provided the best way to create answers to questions by re-combining data – as was done in some of our earlier SQL statements.) It was standard to preserve the results of a join in the form of a new table, but, as you can imagine, such a process yields a huge number of tables and difficulties in managing names, currency, directory structures, and the like. Modern DBMSs permit users to create equivalent tables temporarily, and that is also possible with SQL statements. Those temporary tables can be either the result of a specific search within a table, in

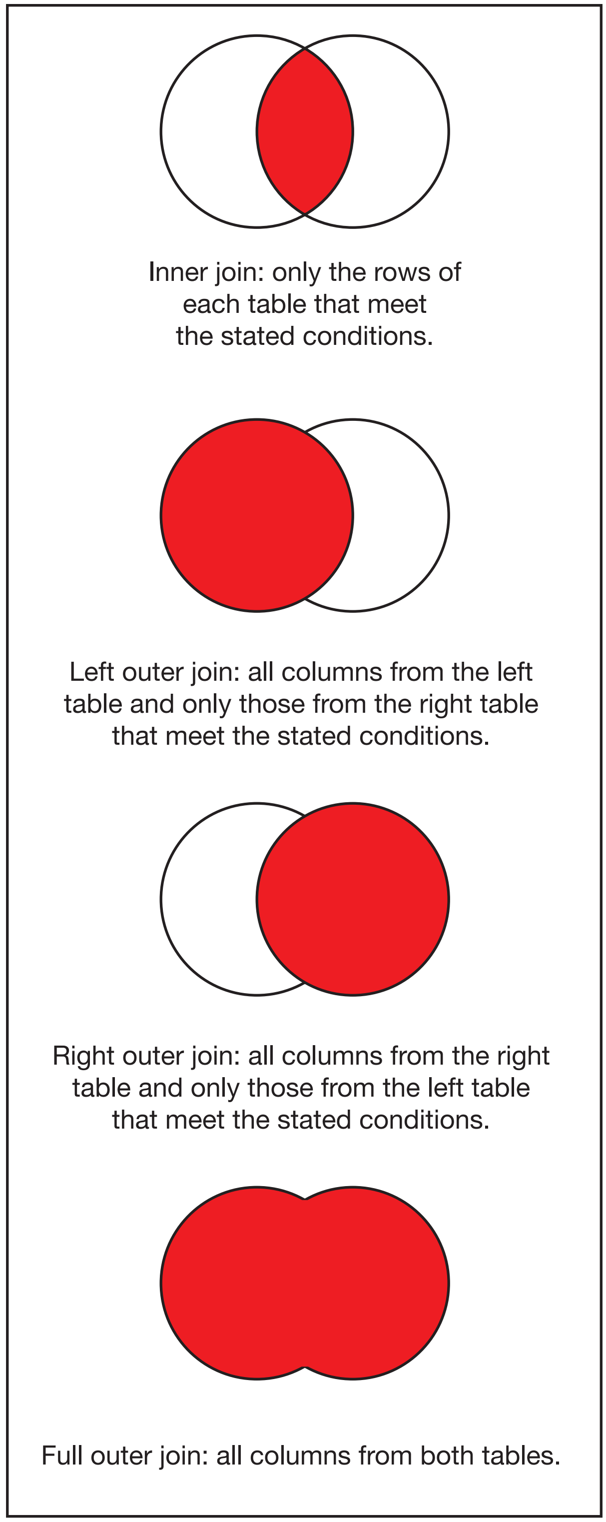


Figure 18
Two tables (the circles) with various joins as Venn diagrams. In each case the red area represents the portion of the information requested by the table join statement.

Large Projects and Networks

Large projects will often require so much time for data entry that a single computer cannot carry the load. In those cases, it is possible to break up the work; for example, people working in different trenches could use separate computers for data entry, with the resulting files put into a single database by system administrators. While that is possible, it is neither as convenient nor as easy to maintain as a networked system involving multiple computers and a single set of data files.

Databases operated over a network have specific problems that are beyond the scope of this book, but anyone contemplating a networked system should be aware of the basic issue: protecting data from the dangers of simultaneous access/editing/modification. If a data entry process for a pottery lot involves, for instance, the entry of a new locus, there must be no danger that another person, operating over the network at the same time, will be adding the same new locus to the system. There would then be two identical or nearly identical rows in the locus table, and the resulting confusion would be very damaging. More important, it would probably go unnoticed for some time, spawning additional errors.

Editing databases opens the possibilities to more dangerous and insidious problems. Imagine a change to a row made by one scholar while another has accessed the same row of the same table. If both access the row at the same time, both start with the same information. Scholar A makes a change of one sort, and scholar B makes a different change. Each commits the change to the database. What happens? In a simple system, the last one to commit the changes to the database would determine the content of the row. The prior change would disappear almost instantly and without a trace.

These potential problems have led to the use of file- and record-locking processes. (The terminology is well-established; using the table, row, column terms would yield table-locking and row-locking.) As the terms imply, a networked database may prevent editing of a table under certain conditions, of an individual row under other circumstances. The protection is essential, and when properly done access to an individual table or row is not denied, but editing permission is.

Working over a network can present other problems, and those often require considerable computer expertise to solve; so the preparation of a large database system will often involve multiple experts working together to fit all the pieces together properly.

Working on a database via the Internet is also possible today, but as with a smaller network, special expertise is usually required. The database design is not dependent upon the presence or absence of a network, though; so the design of tables and relationships need not wait upon the networking processes.

which case the temporary table is simply a sub-set of the starting table, or the result of one or more table joins.

A sample table join might be the result of searching the Moche *Pots* Table and the Moche *Periods* Table for all pots of the shape *jaguar* and displaying the columns Catalog No., Shape, Period, Beginning date, and Ending Date. The resulting table, whether stored or simply retained in computer memory until the user has finished with it, would have included only the jaguar-shaped pots from the *Pots* Table plus the related information from the *Styles* Table.

An alternate is to request that **all** pots be displayed, with the added information from the *Styles* Table shown only for the jaguar-shaped pots. The example we first constructed – only jaguar-shaped pots and related information – is called an *inner join*. The second version, with all pots shown but with the information from the *Styles* Table only for the jaguar-shaped pots, is called an *outer join*. That is, an inner join connecting table A to table B shows only those entries in table A that have a related entry in table B. An outer join shows all entries in table A, whether there is a corresponding table B entry or not.

As seems to be the case again and again, all is not so simple. It is possible to show all rows from either table; so there are two kinds of outer joins: a left outer join (the LEFT SELECT statement used above, p. 86) and a right outer join. In one case the table on the left (placed first in the join statement) has all rows selected. In the other, the table on the right (second in the join statement) has all its rows selected. Naturally, there is a full outer join: all rows from both tables are selected.

There is also a self join, when a table is joined back to itself. Such a join is rarely needed but can provide useful results. For instance, a larger table of Moche pots might be searched for all pots of the same shape but different periods, highlighting the longevity of a given shape or, conversely, the short-lived nature of others. (The results from a self join can often be found via sequential or nested searches, but the self join is more efficient. It can be very useful for comparisons within a table.)

Needless to say, joins can be nested. That is, the results of one join can be used as the starting point for another. In such cases, new tables may be necessary, depending on the DBMS or SQL syntax, as intermediate steps. It is critical that any created tables that are saved to the disk be treated with care so that their contents are clear. If they are to be used only for a temporary purpose, they should be removed from the disk after use. If not, their contents and date of creation should be clearly attached so that a potential user can be sure how they can be safely used. Nested joins require careful construction and sequencing of the join statements and an understanding of the results of the types of joins discussed here to be sure that the results are accurate and complete. (For some examples of joins and their results, see archcomp.csanet.org/dbms/joins.html.)

Honoring Scholarly Differences

The Moche database, with its *Periods* Table, takes no account of possible differences of opinion as to the dates of the Moche periods. That is, different scholars may have different views about the chronological limits of a given period/style, but the database cannot accommodate that – unless another change is made.

This requires no change to the *Periods* Table, but searches would become one step more complex. If multiple scholars applied different dates to the same period and each were recorded in the *Periods* Table with the scholar's name, each scholar-and-dates viewpoint would generate a different row in the *Periods* Table, one per style per scholar. A SQL SELECT statement could then use an addition to its WHERE clause specifying the name of the scholar whose dates were desired. (Without that addition to the WHERE clause, all scholars' views would be found, an interesting exercise in its own right.)

With this schema the database may include many interpretations of the dates for any period, each with the name of the scholar responsible for the interpretation (and, in another column if desired, the rationale for that scholar's view). The rows in the *Pots* table will be related to the rows in the *Periods* Table as before, but the relationship between object and date has become a many-to-many relationship in this approach.

This schema does add complexity; adding the scholarly interpretations certainly complicates the data table where those interpretations reside, and full documentation is obviously required. Doing this would be unnecessarily complicating in some instances, when there is general agreement about stylistic periods and their dates, but crucial in others, when disagreement is significant. This idea is a potentially valuable and important way for **any database** to deal with conflicting interpretations in archaeology. Virtually any interpretive data can be attached to data tables in this way to permit scholarly disagreements to be honored in the database rather then suppressed for the sake of conformity and simplicity.

It may seem that multiple interpretations would not be needed once a database has been finalized at the conclusion of a project. Quite the contrary, provisions to permit adding the views of other scholars to a data set should be a part of any design, and adding such alternate interpretations should be encouraged. Every new archaeological project adds not only information to our total corpus but reasons to revisit ideas formulated in the past. Knowing of the disagreements of the past can be extremely valuable.

DBMSs generally hide from the user the construction of statements that create temporary joins, allowing the user to select columns from related tables for a given form or query without regard for the ways the tables are joined. These processes may assume that all joins are inner or outer joins; so a user must be sure to check default settings and understand the consequences.

Table joins and the terminology may seem unnecessarily arcane here, but conceptually they are the backbone of database design. Their importance in SQL statements also requires that users gain familiarity with them.

Referential Integrity

The question of joins highlights issues of referential integrity. That is, when there are two tables related by a primary-key / foreign-key equivalence, what happens when a row in the parent table is removed? The related rows in the child table then have no reference, something that should not happen. For instance, there cannot be a lot that has no locus (though there can be a locus that has no lots). Therefore, a database system must have rules for enforcing referential integrity. Not only must there be rules about if and when rows can be removed, they must be explicit and carefully enforced. (If a single Gordion fibula turned out to have been a mis-labeled duplicate, it could be removed from the data table – though I would prefer to see a note added in its place indicating the problem, perhaps by placing the note in the description column, leaving the sequence number, and removing everything else. But if a group-and-sub-group category were removed by a thorough reconsideration of the stylistic scheme, the individual fibulae of the particular group and sub-group would not be removed. Instead they would need to be related to another group and sub-group; otherwise there would be no reference in the *Styles* Table. Those examples would stand outside the stylistic analysis.)

Referential integrity is so important that guarding it is a crucial job of a database designer. There are ways to design data tables so that referential integrity is well protected, but virtually nothing can prevent someone from going around the design to change tables outside the design system. For that reason, members of a project team should not work on data tables except via the systematic procedures developed by the designer. In most cases, the team members will not have the skills required to do that anyway, but those who do must be reminded that their skills may permit them to do things they should not.

Normalization

Normalization is the general term for correctly designing the columns in tables so that they follow certain rules. Those rules may seem arbitrary, but they actually serve to prevent common problems. In particular, following normalization rules assures that a

Terminology

We have already discussed the problems of terminological consistency in Chapter I, pp. 21-22. It is important, in this context, to remember that those problems with terminological consistency can be truly disabling in a database. The use of lookup tables and the other options discussed here can help enormously, but today's complex projects, which may involve groups from various countries increase the need for care. Not only must the terms used for a data table be carefully controlled, there must be an explicit and painstakingly crafted thesaurus to provide equivalent terms for each language in use. A very carefully-designed data table might even use separate tables for all terms in such a way that the primary data table includes mostly links to other tables where – given a language specification – the approved terms reside in that language and from which those terms may be inserted correctly into columns from the primary data tables. While this adds significant complexity to the database design (so much complexity that a concrete example of such a table design is beyond the scope of this book), it can ensure terminological consistency from language to language within a project, though not from project to project of course.

given item of information occurs in only one place and makes certain that table joins create accurate results. Databases are said to be normalized to various levels, each level being carefully defined and each succeeding level requiring more care and yielding smaller, though far from insignificant benefits. Each higher level assumes that the database has been normalized to the preceding level.

If a database is in first normal form, all columns are atomic. That is, there is only a single value in each column. This was discussed early in this chapter; the first version of the Moche pottery database had both multiple sources and multiple comparanda in individual columns. It was not in first normal form.

To go from first normal form to second, one must be certain that all columns are dependent on the entire primary key, not a subset of it. This may seem strange, since we have used single-column primary keys. However, many database designs involve multiple-column primary keys. When that is the case, it must not be possible to omit one (or more) of the columns from the primary key and still have a functional primary key. In such a case, the primary key should be changed. For instance, had we used the inventory number plus the object number in the Moche pottery table as the primary key, the key would have been unique, but the inventory number would have been unnecessary. All the information depended on the object number, and the use of the inventory number would have added nothing to aid our ability to identify any individual sample. This requirement may seem pointless, but it prevents any confusion about the relationship between the primary key and the other columns. Each column is attached to the object number alone, not the object number and the inventory number. Anyone using the database should understand that immediately, but confusion would exist if the primary key included the unnecessary inventory number as well.

To go up the ladder one more step to third normal form requires, in addition to the first- and second-level restrictions, that everything in a row depends on the primary key, not another column in the row. That requirement made our original Moche *Pots* Table fail to meet third normal form. The dates for each pottery style are dependent on the style, not the pot. Therefore, the dates should have been in another table with the style as the primary key, and the *Pots* Table would then be related to the *Periods* Table in a many-to-one relationship using the period number as the foreign key (as in the schemata in figures 16, 17, and 18). This is not a foolish requirement at all; the *Pots* Table should not have included the dates because for each style there is only one set of dates. The information was repeated in the initial design, something that should not be permitted. Consider what happens if the Moche II and Moche III periods are reconsidered and the date boundary between them changed. In the *Pots* Table as it originated all the Moche II and Moche III all examples would need to have the dates changed. On the other hand, a *Periods* Table would provide only one place to change the ending date for Moche II and one for the beginning date of Moche III. This difference is not simply a matter of convenience or time. It is a matter of error prevention, since there is only one change required for any "fact."

It may have occurred to you that adding a sequence number to a table as a primary key violates third normal form. With a sequence number there may often be two columns in every row that determine the others. One is the primary key – the sequence number – and the other is the more natural primary key – the catalog number, the locus number, etc. The reason for using this "illegal" addition is practical. The sequence number is rarely seen or even known to exist. As a result, it is virtually impossible for it to be changed by accident, and that helps to prevent accidental confusion. More important, it means that referential integrity is not lost if what might, at first blush, seem to be the primary key is changed. In our Moche example, for instance, the catalog numbers may be changed multiple times as the exhibit develops. That would compromise referential integrity if the catalog number were the primary key, but that is not a problem if a sequence number provides the primary key instead. Relationships to other files remain intact because the primary key does not change.

Fourth normal form seems rather trivial. If there are more than one one-to-many relationships and those relationships are independent of each other, each relationship should be expressed with its own table. It is a bit hard to imagine someone needing this requirement, since it seems obvious. The Moche database again provides an example for us. The *Pots* Table is related as one-to-many to both the *Sources* Table and the *Comparanda* Table. It would have been possible to construct a single table with comparanda and references plus the foreign key (catalog number). Each of the columns – source and comparandum – would meet third normal form requirements because each would be dependent only on the primary key. However, since they are unrelated to each other, they should be, as they were in our original design, in two separate tables.

One suggestion for relating needed photographs to catalog objects, lots, loci, and operations did not meet fourth normal form requirements. Entries in a table for more than one subject type, as suggested, would violate fourth normal form. The members of that table could have virtually nothing in common, some being catalog items, some lots, some loci features, and others operations.

Finally, there is fifth normal form, something to which all database designers aspire and which few database designers reach. Fifth normal form is not a condition so much as a sense of aiming for the perfect, the *Tao*. One may reach fifth normal form when there are no data tables that can usefully be split. In other words, database perfection consists of splitting tables until there is no longer any purpose – a nicely ambiguous end result.

Practicality often requires that some of these rules be ignored. For instance, a database with city, state, and zip does not meet third normal form requirements. The zip determines city and state. But how many databases will – or should – include a zip code table? As discussed, the use of a numeric sequence as a primary key may also violate third normal form. Nevertheless, keeping these normalization rules in mind, even while consciously breaking some, helps in designing any database. Fifth normal form may be more a dream than a concrete goal, but fourth normal form is a good and appropriate aim. In any case, normalization should be discussed in the documentation of the entire data set. In the end, the most important reason for attending to normalization is to make sure there are no duplicate entries, no ambiguities, and no incorrect responses to data queries.

More Documentation

All the design issues are included in the documentation required for any database, as is the discussion of normalization. The information about the design process, entry procedures, normalization, and so on, while much less important for a user of the database, is critical for anyone evaluating it. These are the issues that tell an outside reviewer or the project director whether all the bases have really been covered, whether data entry was properly designed, whether data integrity was properly guarded, and so on. These issues are critical for anyone trying to see whether the design was implemented well. Taken as a complete package, the documentation provides both guidance for users and a demonstration of the professionalism of the database designer. It therefore provides the primary means for judging the data set.

There is one more bit of information that is important for evaluation. That must come from the project director and it is simply an evaluation of the database system in use. Was data entry "as advertised"? Did the pieces work together? Was too much training necessary? These evaluations may not need to follow a database as the other information must, but they are critical to a full evaluation and important for the designer who expects to do similar work again.

Conclusion

To conclude this discussion of databases, let us return to the seemingly simple example with which we began – the *Weather* Table. If this were a database retained by a newspaper wire service for widespread distribution, how might it be structured now that we know more about how real databases work?

First, each city would exist in its own row of a table with a sequence number for its primary key (or with a primary key consisting of city plus state plus county plus country) to be sure that all the Springfields in the U.S. can be kept separate and to make certain that Athens, Greece, and Athens, Georgia, are not confused – not to mention Cairo, Egypt, and Cairo, Illinois; Lima, Peru, and Lima, Ohio. In addition to the city, state, county, and country columns, there might also be columns for the city name in a multitude of languages – or related tables for various languages.

Second, each subscribing newspaper might have its own table of cities (by foreign key, not name), selecting those to appear in its daily weather table. An international paper might include the major cities of the globe plus the codes for the larger cities in the countries where it is most widely distributed. An American paper in the northeast might include more cities in the northeast and fewer in the rest of the U.S., along with a smaller selection of international cities. The subscribers' tables would probably be the only ones to which the subscribing newspapers had real access, since they, not the wire service, would decide which cities to include. Indeed, those tables might reside in the offices of the newspapers rather than the wire service.

The daily *Weather* Table might include precipitation and barometric pressure readings for all the cities in the database in addition to high and low temperatures; newspapers could choose to download that information or not. Of course, any data would be held in the unit of measurement chosen by the database designers, but it would have to be supplied in terms preferred by the users. Thus, temperatures might be stored in degrees centigrade, but they would actually be supplied in Fahrenheit or centigrade, as requested. Therefore, the data might pass through a translator between retrieval and transmission. (Of course, a decision about how to deal with missing data would be required. Recording a temperature of zero would be wrong if, in fact, no temperature had been reported. Nothing recorded for the temperature yields ambiguity. Perhaps 1000 degrees C. would be used for a missing report, permitting averages to be calculated after selecting all reports other than those indicating a temperature of 1000 degrees C. (Since there would doubtless be a highly automated process for obtaining information and adding it to the table, using a null column in this case might be acceptable. The chance of an accidental null entry could be taken as vanishingly small, making the null column clearly indicate an absence of data.)

There might be another separate table for every newspaper using the system. In that table would be the language, units of measure, and desired attributes for that newspaper. Again, that table might be held by the newspaper rather than the wire service.

Given this system, each newspaper would initiate a request for weather information by sending the date and information from its tables (specifying cities, languages, units of measure, and attributes desired) to the wire-service computer. That computer would fetch the data – in correct units and languages – from the appropriate tables and transmit it back to the paper as a daily weather table, ready for insertion in the proper place.

As you have probably realized, the scenario described is by no means the only possibility. City names, for instance, need not be in the wire service data at all and might reside only in the local newspaper's table where language and script issues can be ignored. A numeric foreign key in the local table could be used to connect to the wire-service table's weather information via the primary key in that table.

When a data set such as this weather data set is created, it is not constructed in isolation. News organizations may want other information about countries, states, and cities, information totally unrelated to the weather issue; so other information categories may be indicated. Those other categories may lead to additional tables for other kinds of information that reporters might want to be able to access quickly. The process can expand indefinitely; the limits are only those of practicality. At some point the problems – and costs – of maintaining all parts of a huge data set become too numerous for the benefits.

One can easily imagine tables such as these weather tables being used in many archaeological data sets – tables of pottery styles, for instance, or a table of fibulae styles that could be plugged into excavation, survey, or catalog data sets. Such tables could assist in encouraging the use of more standard vocabulary in archaeology. In fact, the absence of widely-used vocabulary standards in archaeology remains such a problem that the use of common tables for terms, no matter how desirable, remains very unlikely. Nevertheless, the various ways data can be organized for something so seemingly trivial as a weather table illustrates well the potential for any well-designed set of data tables in archaeology. The discipline of archaeology adds other problems because the "facts" in archaeology are often less simple than the facts of the weather. When the temperature is measured, that is a reasonably simple data item, and much data in archaeology is equally simple, but much is not or is subject to dispute. As a result, archaeological data sets should be more complex and should include ways to honor multiple views of the same finds. It is not easy to make a database subtle; it seems a positivist item. For archaeology it cannot be. Too much is interpreted.

Selected Further Resources:

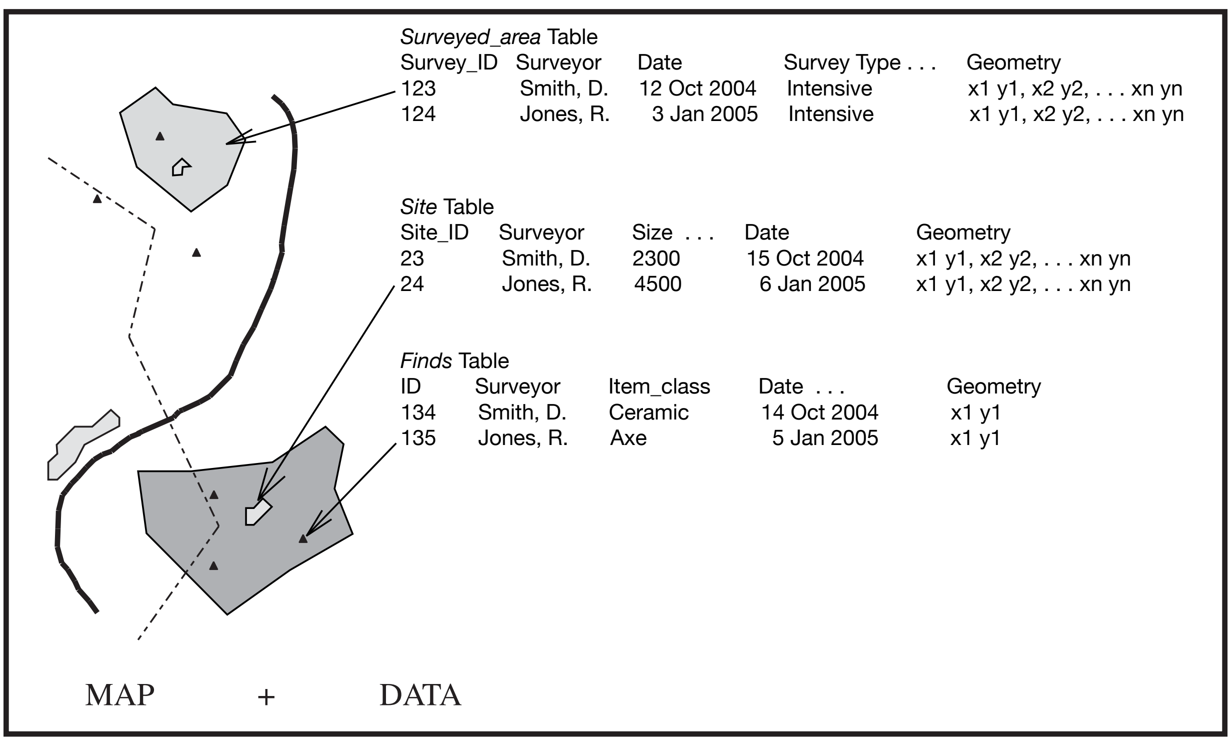
There are many books and web sites about database design. Virtually any of them may be of help, but, assuming this chapter has been helpful, the best place to go now is probably to a computer with a specific program (preferably one of the database management systems that runs on many operating systems), a manual or one of the commercial training books for that program (or both), and a problem to solve. That may mean going to a computer center to avoid buying hardware or software (a good plan), but the real key is making your problem complex enough to push yourself.

Start with table design, and spend some serious time making sure that the table design is good. Then try out various procedures for data entry, queries, and reports. Data entry procedures should include some complicated limits on entries enforced by macros and some automatic entries (date or user name, for instance), and reports should be suitable for presenting data to colleagues or even paper publication of tabular results.

I believe that your own interest in the data with which you experiment is critical. If you know and care about the data, you will also know and care about the subtleties of its storage, preservation, and presentation. As a consequence, you will see the problems and pitfalls that sub-optimal organization might cause.

IV

Combining Maps and Data: Geographic Information Systems



Glossary

Attribute: equivalent to a field or column in a data table and also the standard English usage; an attribute is simply a characteristic of something.

Buffering: An operation that creates rings of specified distance from the selected entity. If the entity is a point then buffering creates a set of bull's-eye type rings.

Cell: a square or rectangular unit of the earth's surface defined by a GIS system so that data about the area encompassed can be attached to the cell. Cells are equivalent to pixels in some forms of imagery.

Control point: a specific location for which the x, y (and often z) values are known so that it can be used to orient an image or map properly. See rectification.

Coordinate system: a system for supplying mathematically precise coordinates for any point on the earth. A coordinate system must be based upon a chosen datum to retain its accuracy, and it may be related to a specific projection. (UTM and state plane are examples of coordinate systems.)

Cost Surface (Friction Surface or Movement Surface): a measure of the effort/expense/time required to traverse and area because of impediments to travel.

Coverage: A term used in earlier versions of the ArcInfo GIS software system for a feature class or layer.

Datum: a mathematical representation of the shape of the earth. A datum must lie at the heart of any system intended to model the earth, but no datum is completely accurate; therefore, several are in common use.

DEM: digital elevation model, a grid of elevation values, equivalent to DTD or digital terrain data.

Digitize: 1) to convert from analog to digital form, generally in some automated fashion; 2) to convert manually from an analog original into a digital format, as when plans or drawings are copied with a digitizing tablet or scanner.

Digitizing tablet (digitizer): an electronic drawing tablet connected to a computer. The tablet can function as a mouse, controlling cursor movement in a relative sense. With some CAD programs a digitizer can also be scaled so that it functions more like a drafting board. (A digitizer that has been scaled may be used to digitize a paper drawing; such a drawing, placed on the digitizer, may be traced to create a digital version of it.)

DTD: digital terrain data, a grid of elevation values, equivalent to DEM or digital elevation model.

Feature Class: See layer.

Layer (Theme, Feature Class): a portion of the GIS data separated from others for any reason whatsoever, be it spatial, temporal, chronological, or conceptual. The GIS layer is roughly comparable to a table in a database.

Line: a basic element of vector data composed of the connection of two or more pairs of x and y (and sometimes z) values.

Multi-band imagery: Digital imagery composed of measurements from multiple portions of the electromagnetic spectrum. A common example is an RGB image with separate values for the red, blue and green portions of the spectrum. When combined, the result is a full-color image. Other examples are CIR (color infrared) and multi-spectral, such as many satellite images.

Multi-spectral images: See multi-band imagery.

Orthorectification: A multi-step process that first removes the distortion due to optical properties of a camera and lens (see rectification) and then removes distortions that are due to elevation differences across an image. The result is an image for which accurate ground measurements can be made across the entire image.

Overlay: a process that places a second layer "over" a first either to create a new combination of both or to select items in the first that have a relationship with items in the second, for example all the findspots in one layer that are "inside" the soil polygons in the second.

Point: the basic element in vector data, a single pair of x and y values. It may also include a z (or height).

Polygon: one of the basic elements of vector data, along with point and line. A polygon is a closed area bounded by lines that meet at the corners (vertices) of the polygon; a polygon is defined by its vertices.

Projection: the method used to project the nearly spherical shape of the earth onto a flat surface such as a piece of paper. All projections distort geography to one degree or another.

Raster: a term used to describe an image that consists of individual points of color or shades of gray. A standard photograph or a satellite image is a raster image.

Reclassify: a process that converts values in a map or file to a new set of values based on a set of rules or operations. For example, all elevation values in a map that are greater than 100 may be reclassified as "high" in a new map.

Rectification: the removal from an image of optical distortion created by a camera and lens. See also orthorectification.

Register: Apply known locations to selected points (usually four or more) in an image or map. The first step in one form of rectification.

Resolution: in GIS systems resolution refers to the size of the individual cell to which data may be attached. At 10 m. resolution, the cell is 10 m. x 10 m., or 100 square meters.

Select: use attributes or geometry or both to identify some elements in a layer. The resulting elements are referred to as a select set.

Shape file: An ArcView file format for vector data. The shape file is actually multiple files in a single computer folder. These include geometric data in a file.shp and attribute data in a file.dbf as well as others.

Snapping: a process that connects lines that are close to each other. A line that comes with in a specific distance of an original line may be snapped to the first.

Theme: See layer.

Topology: the relationships of areas, lines, and points to one another.

Vector: a term used to describe an image that consists of vectors, lines (curved or straight) that can be defined mathematically and therefore reproduced at any scale on command.

Vertex: the point where two lines composing a polygon meet.

Visibility map (viewshed analysis): a map showing areas of terrain that can be seen from a given point.

Voxel: a three-dimensional cell, normally having the same resolution in all three dimension.

Introduction: A simple GIS

Consider a map of the city blocks in a small town. Lines on the map enclose each block and within each block other lines define the boundaries of each parcel, the sidewalks, and maybe even each house, as in figure 1. We might want to create a map that shows all the blocks that are larger than a specified size or all the parcels that have a value of more than some dollar amount or all the sidewalks that are brick. To do that we could create multiple copies of our base map with a copier. We would then refer to a book (or perhaps the town database) that had the parcel information and sidewalk properties and use those data to color the areas that meet our requirements, as shown in figures 2 and 3.

If we were more inquisitive, we might want to make a map of all the parcels that were worth more than the specific amount and that were adjacent to a brick sidewalk. To do that we would need to look up the data about parcels and sidewalks, visually identify the ones that were adjacent, and then manually color them as in figure 4 and figure 5. In doing all these things we would be performing the operations that a GIS performs with the aid of a computer.

Geographic information systems are most simply defined as digital representations of maps, attributes of items in the maps, and attributes of related items; to this are added ways to query and manipulate the maps, their contents, and the attributes. The first key to the functionality of geographic information systems lies in their ability to store map information in tables or table-like forms along with attributes of the map entities so that the map entities can include and/or be connected to text/numeric information about them. Data in other tables – either maps or ordinary data tables – are connected via primary-key-to-foreign-key links to permit standard database operations (e.g., *select* statements and table joins). As a result, GIS software can utilize relationships between graphical map entities (parcels and sidewalks or sites and features) as well as relationships between those map entities and standard data tables. Thus, GIS programs tie the traditional graphics of the map to attribute information stored in data tables.

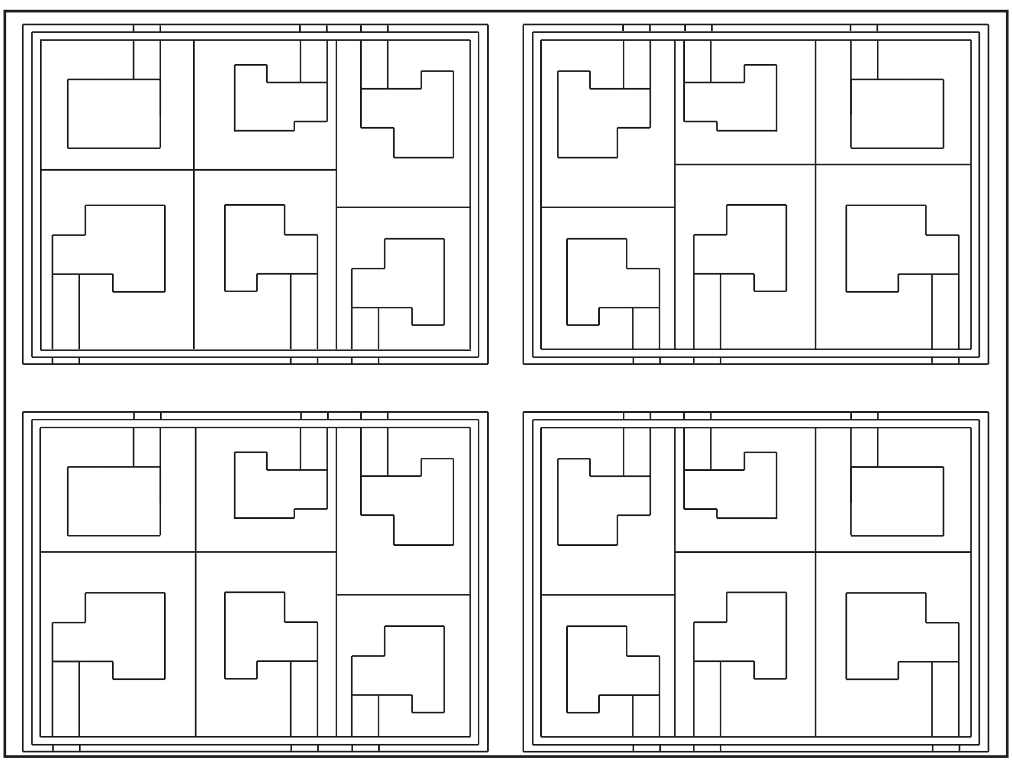
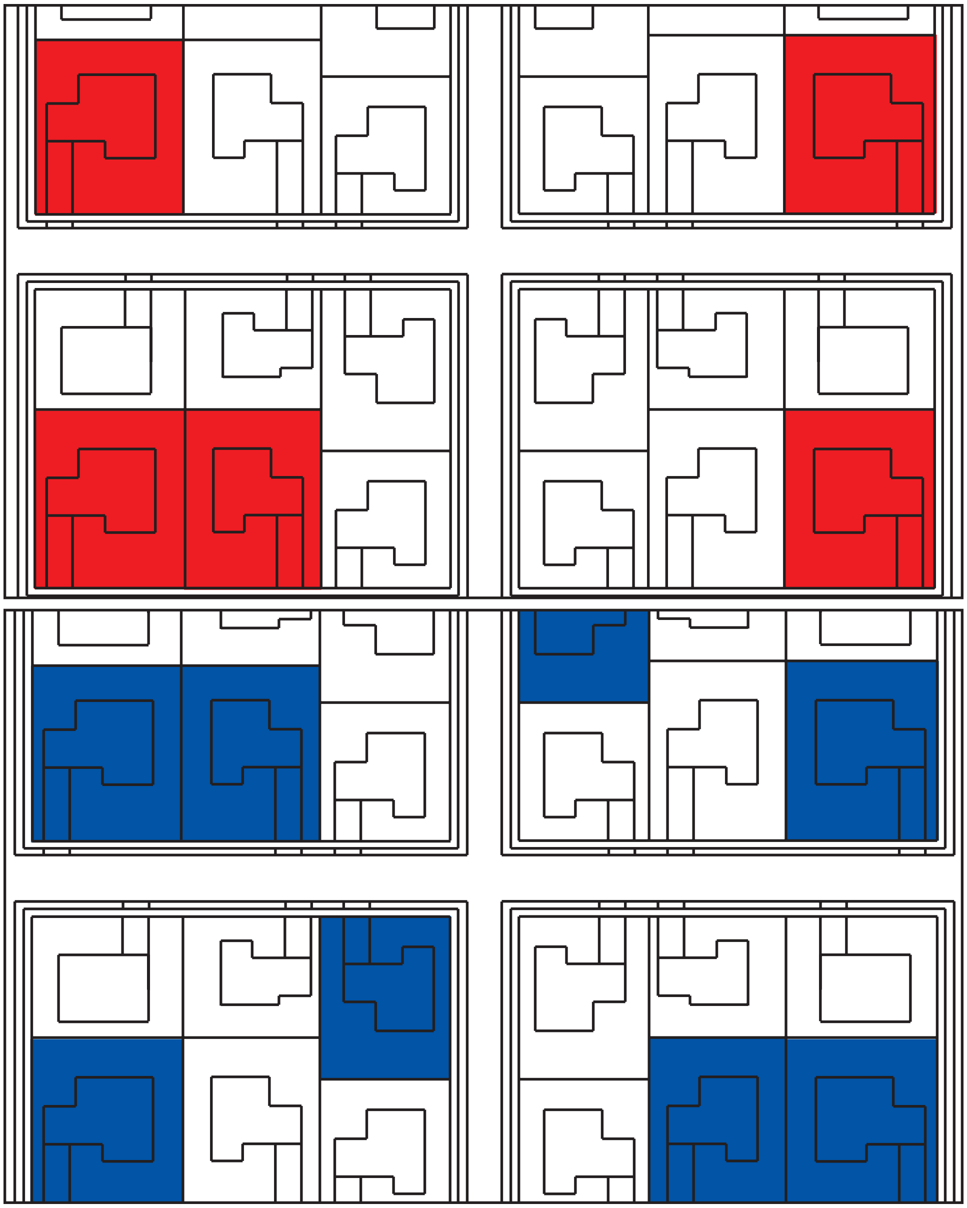


Figure 1

Hypothetical town map showing four adjacent blocks, with parcel boundaries, house footprints, driveways, and sidewalks. The sidewalk serves as one of the boundaries for all these parcels.

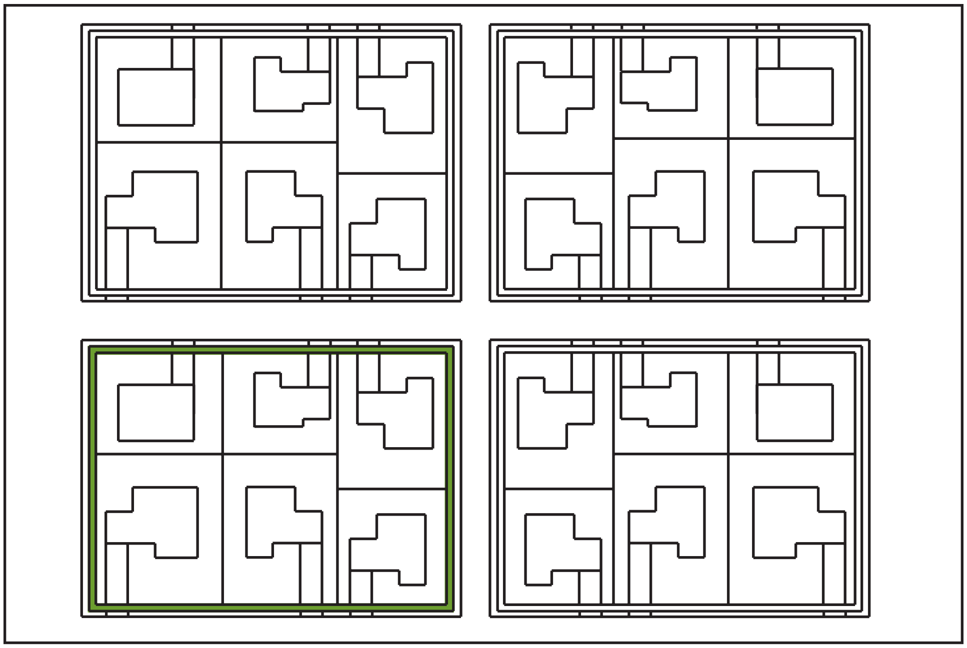


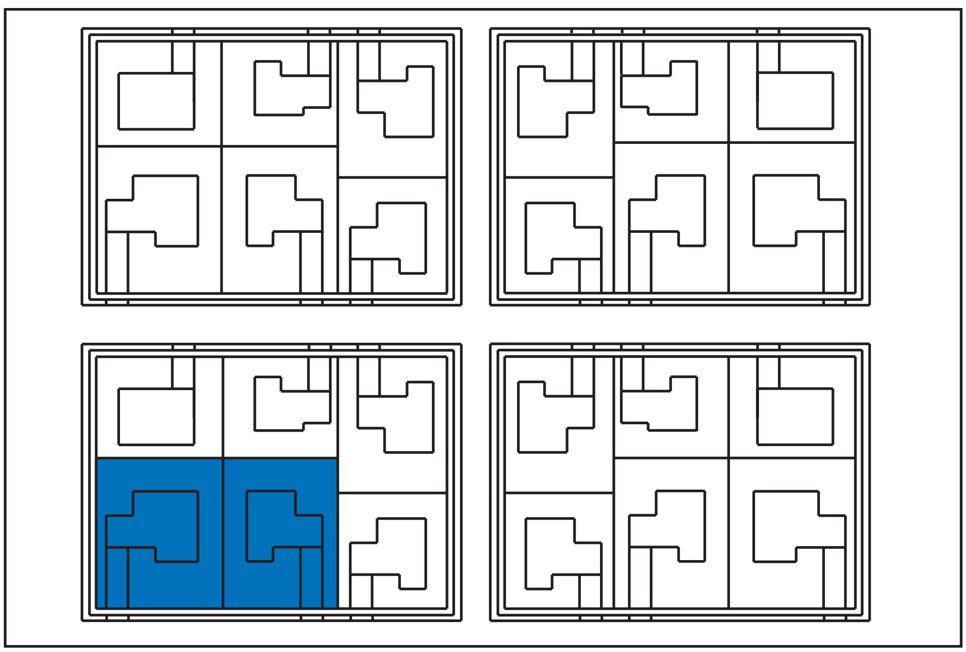
Figures 2 and 3

Truncated maps of the hypothetical town with parcels larger than a stated size shown in red (above, figure 2) and those appraised for more than a specific dollar amount shown in blue (below, figure 3).

The second key element is the built-in understanding of topology. Topology refers to the types of associations that map or geographic entities may have to one another. A water line may *cross* a sidewalk. A parcel may *be adjacent to* the street. The house may *be inside* the parcel. These relationships are topological (not to be confused with the archaeological term *typological*!) and are "understood" by GIS software. As a result, the software can identify those map entities that have a topological relationship to one another.

GIS systems use the database-style relationships to access data according to primary and foreign key equivalence. With GIS, however, the relationships can connect map entities and text/numeric data, not just text/numeric data. Furthermore, GIS software can identify topological relationships that are not made explicit in the data but are only implicit. Thus, GIS can utilize both spatial and attribute relationships – either separately or at the same time – to assist archaeologists in understanding and analyzing their information.





Figures 4 and 5

The hypothetical town with brick sidewalks shown in green (above) and with properties appraised for more than a stated amount and adjacent to a brick sidewalk in blue (below).

Some GIS terms

Before we continue to explore the GIS world, we need to start with some basic definitions. First and most basic are *vector* and *raster*. Though defined earlier (see Chapter II, pp. 44 ff.), these terms must be more fully defined for this context since they refer to the two basic ways in which data are stored and used in a GIS. Vector data are composed of points, lines and polygons. We can represent these just as we did in geometry class. Take out a piece of graph paper, put a dot on the paper and you have a point. A point is a single location (with a single x value and a single y value specifying position in a grid); in a GIS that might represent the location of a fire plug or a lamp post. Lines go from one point to another, each point defined by an x value and a y value. A polygon is a set of lines that enclose an area; each point or vertex also has an x and a y value. The lines that surround one of the city blocks create a single polygon.

Figure 6 shows vectors on a graph. The location of a point, line, or polygon is defined by the x and y coordinates of each point, line end, or vertex. These might be in arbitrary values – as on our graph paper – or they might represent locations in latitude or longitude or other map coordinates.

Note that the beginning and ending coordinates of any polygon, regardless of the staring point chosen, must be the same. This serves to close the polygon on itself and so to distinguish it from a series of connected line segments. Thus, a triangle has three corners but four sets of coordinates, with the ending coordinates being the same as the starting coordinates; the rectangle in figure 6 has five sets of coordinates for its four vertices; the six-sided polygon has seven sets of coordinates for its six vertices. Normally real-world lines and polygons have many, sometimes hundreds or thousands, of these coordinates. Each of the x and y coordinate pairs in a line or polygon is referred to as a vertex.

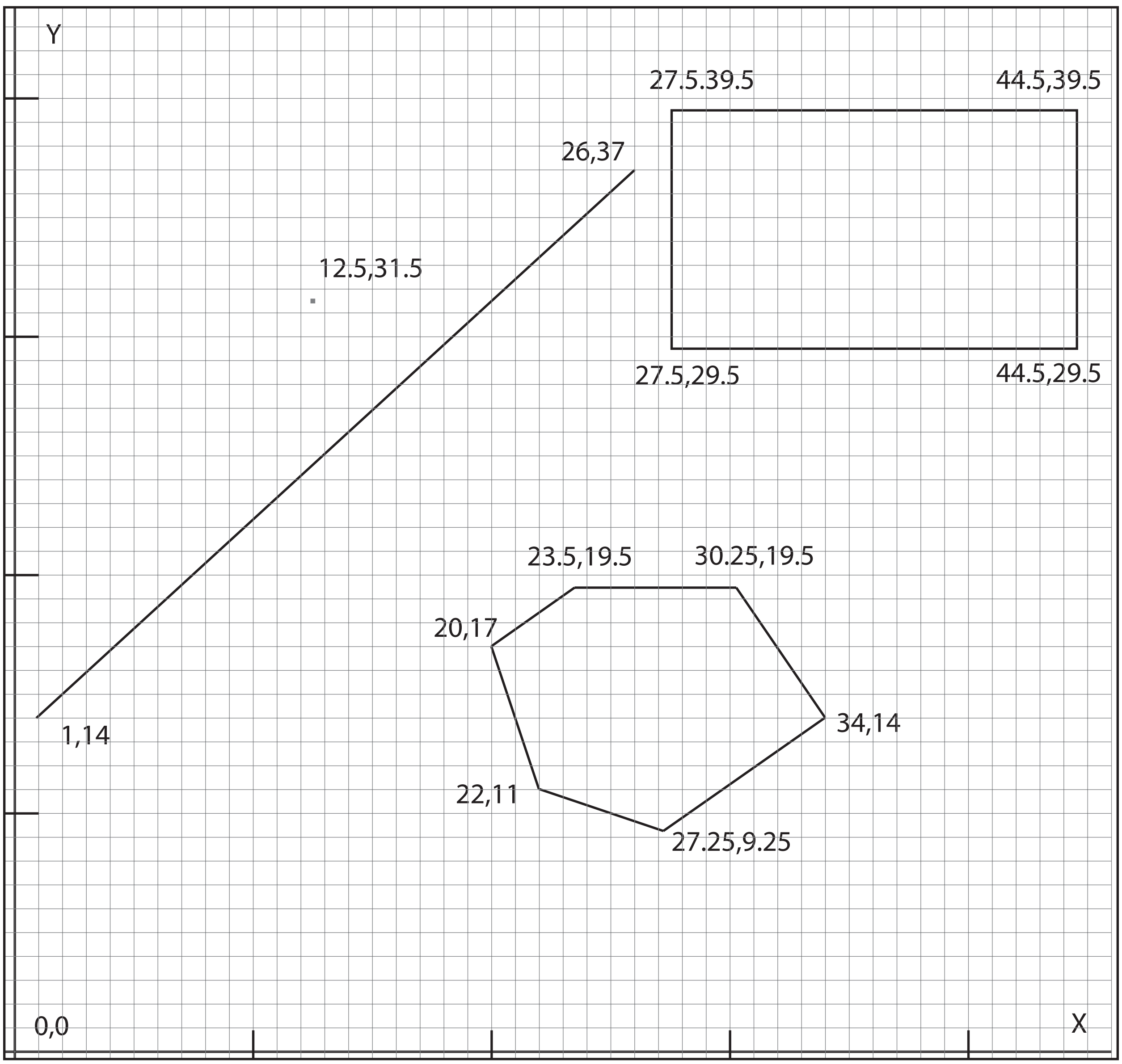
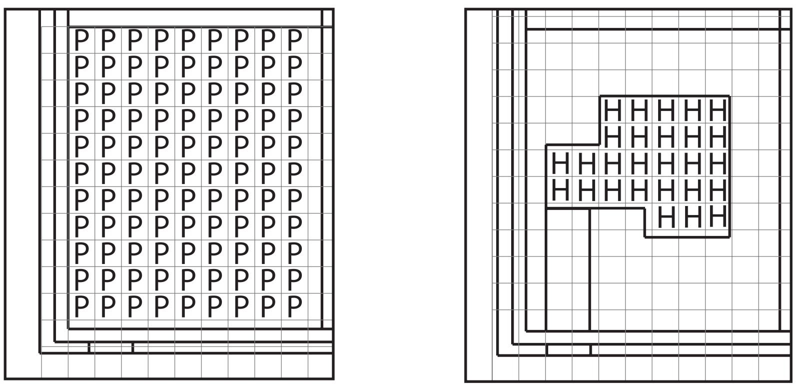


Figure 6

Graph paper with a point, a line, and two polygons, one a simple rectangle and the other a more complex and irregular polygon. All points, line ends, and vertices have x and y coordinates. Note that, regardless of the starting point when making a polygon, the beginning and ending coordinates of the polygon must be the same so that it closes on itself.

A vector map may consist of the vector entities alone or those entities with associated attributes, as in the *Parcels* Table shown below (p. 115). In any case, each vector entity must have an associated ID – the primary key of any relational database – to permit it to be unambiguously identified for the sake of relating it to other data, whether map data or other tabular data.

If vector data resembles lines on graph paper, raster data refers to data when they are simply filled grid squares on the same graph paper. Imagine taking another piece of translucent graph paper and placing it down on the vector map of our little town. But now we treat each grid square as a single data item. That is, we would put a P in every grid square that is within a parcel, an H in every one that is within the footprint of the house, maybe a G if the cell is in a grass area, and so on. Each of the grid boxes would be a cell in the raster GIS. It is a useful simplification to think of a vector GIS as based on lines and their coordinates while a raster GIS as based on the spaces within the grid lines, the cells. Thus, the smallest raster unit, the closest thing to a point, is a single cell.



Figures 7 and 8

The lower, left property from the upper left block in the hypothetical town. Figure 7, left, shows the entire parcel, with each grid square included within the parcel boundary filled with a "P." Figure 8, right, shows the portion of the parcel occupied by the house, with each included grid square filled with an "H."

In a raster map a point is represented by a single cell, a line by a set of single-width adjacent cells, and a polygon by a group of adjacent cells, as shown in the illustration, figure 9. The boundaries of the cells can be given precise coordinates, but the relationships within the GIS database using raster data are based on the content of each cell, not the boundaries of cells. Of course, cells often do not share the boundaries of the represented feature. In the drawings in figures 7 and 8, for instance, there are cells lying partly within the parcel boundary that are not marked with a "P" as a result of not being fully contained and cells not fully within the footprint of the house and therefore not marked with an "H."

Photographs are commonly stored in a GIS in a raster form. In this case each cell (called a pixel when speaking of images) has a number. If it is a black and white photo the number in the cell is the brightness of the image at that place. A zero is black and a larger number (commonly 255) is white. In a color photo (or other multi-band imagery like satellite data) the photo is actually represented by three (connected) raster layers each with values between 1 to 255. The first layer is the value of the red component of the photo at that location, the second the green value at the same place, and the third the blue value.

For the raster map the amount of real-word space represented by a cell in a map is termed its resolution. For example a cell might represent 1 m. on the ground in the x direction (usually north-south) and 1 m. in the y direction (normally east-west). The resolution of a cell defines the limits of the real-world measurement

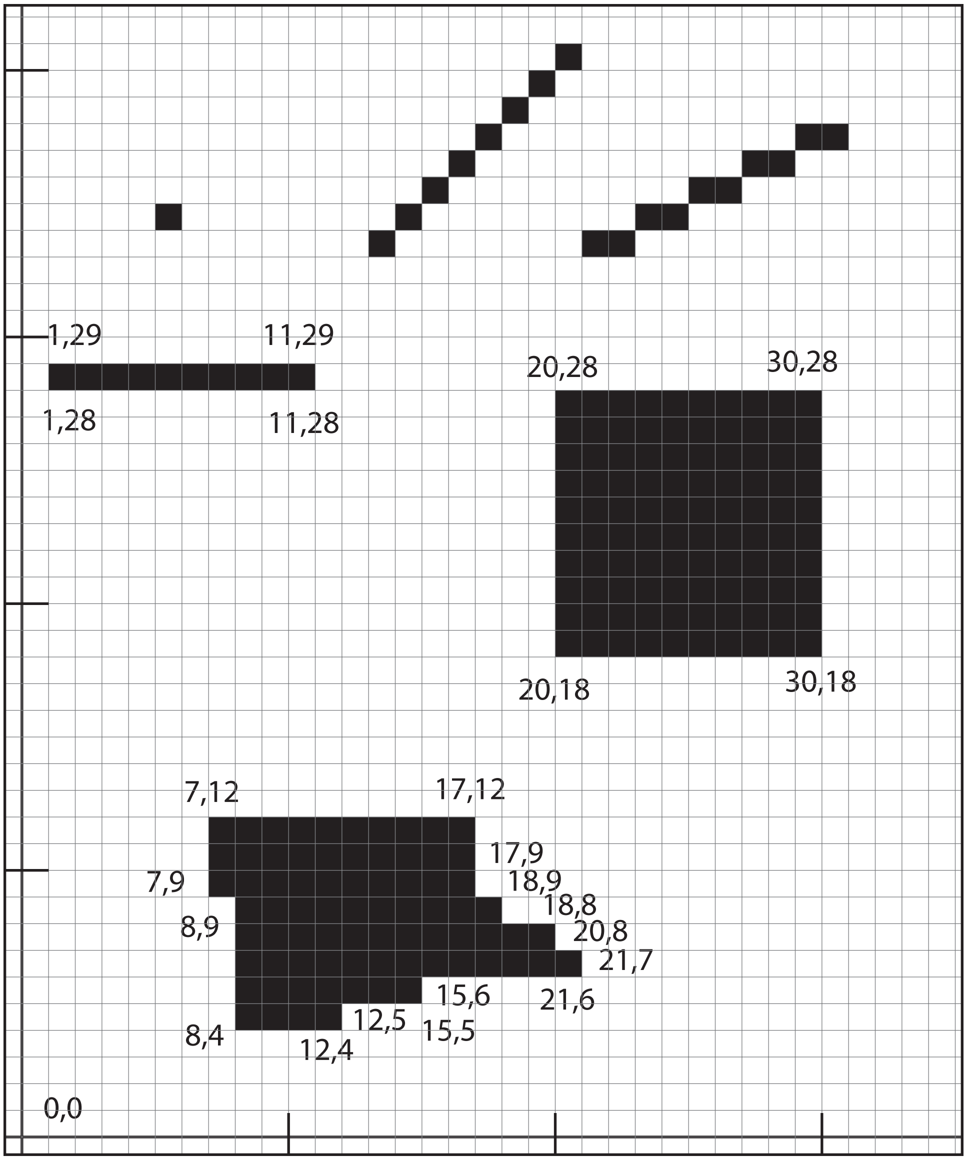


Figure 9

Graph paper with raster-format data. In this instance cells are filled to indicate areas of interest. Each cell here 1 unit by 1 unit.

that the cell can represent. A cell that has a resolution of 30 m. in each direction then may have only one value to represent its area of 900 sq. m. (30 m. x 30 m.). A resolution of 10 m., would yield nine cells (3 across and 3 down, each covering 100 sq. m.) to cover the same real-world space that a single 30-m.-resolution cell would cover. While it would seem that a smaller resolution would always be desirable, there is a trade-off. As the cell resolution becomes smaller, the number of cells grows, as does the processing time required for any operation using the cells. Take a study area that is 10 km. by 20 km. If you recorded data with a cell resolution of 100 m., each cell would represent an area of 10,000 sq. m. (100 m. x 100 m.), and you would have a raster map 100 cells across and 200 down. That would be a total of 20,000 cells. At a 25 m. resolution there would be 400 cells across and 800 down for a total of 320,000 cells, each representing 625 sq. m. (25 m. x 25 m.). At 5 meters there would be 2,000 by 4,000 cells (each of 25 sq. m.) or a total of 8 million! Calculations on the 5 m. cells would take 400 times longer than calculations on the 100 m. cells. Given those processing differences, scholars must consider carefully the pluses and minuses of different resolutions, especially in larger study areas.

Map Scale and Raster Resolution

At first blush it may appear that it is always advisable to use the smallest possible cell resolution in a raster analyses. This is not the case, even if processing demands are ignored.

It is possible to use an inappropriately small resolution that is not supported by the source data. If data are derived from maps, for example, then the map scale is a key factor in determining the appropriate cell resolution. Common map scales for field maps used by archeologists in field surveys are 1:24,000 or 1:50,000. A map scale of 1:24,000 means that a real object (a stream for example) that is 100 feet wide would be represented by a line that is only 0.05 inches thick on the map. Conversely, given that a pencil line on a map is normally at least 0.02 inches wide (often more) the actual level of detail that this reflects on a 1:24,000 map in the real world is 40 feet. With a map scale of 1:50,000 the pencil line at 0.5 millimeters represents 25 meters. (Note: the calculations with meters are relatively easy, often involving no more than moving the decimal. Scaling using English measurements requires much more care, and the results may seem counter-intuitive. Care is required.) In addition to the issue of line width there is the question of map accuracy. That is, how accurately does the position of a point on a map reflects the real-word location of the object? The US Geological Survey has a National Map Accuracy Standard. This states that for maps of a scale larger than 1:20,000 in general the points will be located with an accuracy better than 0.033 inches on the map (nationalmap.gov/gio/standards). For a 1:24,000 scale map, this is the equivalent of +/- 66 feet when scaled to the real world.

There are many different ways to express map accuracy. For example, there are different accuracy parameters associated with the various British Ordnance Survey maps, and these are expressed using the root mean square error (RMSE) method. The process involves comparing the mapped location of a number of points to the actual locations. The differences are then squared, added together for the group, and the square root taken. As a rule of thumb, the RMSE is the same or a bit larger than the average amount of the error. Ordnance Survey publications state that the RMSE for their 1:2,500 scale map (outside built-up areas) is +/- 1.1 meters and that no single point should be in error by more than 3 m. (Ordnance Survey 2004).

It would make little sense to have a resolution substantially finer than the map sources. Even when data are acquired in the field with a GPS, a similar situation prevails. Depending on the quality of the GPS system the locational accuracy may be in the range of +/- 1 to 30 m. depending on the specifics of the system and the way it is used. Similarly, a satellite photograph taken so that each pixel represents a 100 m. x 100 m. area on the earth automatically presents the user with the scale to be used, whether a finer scale is desired or not. In this case, the relationship between the incoming data and the scale to be used is so clear as to be effectively inescapable.

Combining the map's starting point coordinates (usually the upper or lower left corner of the map) and the resolution of each cell, a user – more likely a computer – can calculate the location of any cell, based on the number of columns over and rows up or down from the starting corner's reference coordinates.

Whether a raster data file is imagery or simpler data, it consists simply of numeric values. Those values are stored in a specified sequence so that the underlying software can determine the value applied to any cell in the map; database operations, in turn, can link cells to more data via the numeric values. In other words, the relationship between the location of a cell and its value is implied by the position of the value in the data file (and information about starting point, resolution, orientation, and so on); it is only explicit to the extent that software can make a translation from file position to cell location. As a result, manipulating a raster file outside the GIS software can be a recipe for disaster, causing the loss of the implicit relationship between cell and data.

As we will see, vector and raster maps – our two different types of data – are usually used for very different types of archaeological analyses and management.

Map Projections, Datums, and Coordinate Systems

An equally important factor in the proper use of any GIS is an awareness of map projections, datums, and coordinate systems, which are among the most complex issues relating to the use of digital map data.

When any map data are displayed on a map or a computer screen the 3D real world information must be converted to a drawing on a flat surface. There are three parts to the conversion process that we will review here. Before we start, however, it will be reassuring to know that most capable GIS systems have the internal capability to convert GIS data from one datum, projection, and coordinate system to another at a click of a button.

The first task in the process is to define the shape of the earth mathematically. As we all know, the world is not a perfect globe but is actually a bit pear-shaped. A mathematical representation of the earth's shape is called a datum. The specific elements of a datum change as the precise shape of the earth is better measured, and different datums are used in different parts of the world. None of them provides a mathematically perfect fit, but it will be critical for you to know what datum/datums is/are in use where you are doing your field work. A common global datum, and the one used by the GPS system, is the World Geodetic System (WGS-84). Many older maps and the digital data derived from them were prepared based on a range of different datums. In the US, for example, many maps are based on NAD-27 or the North American Datum of 1927. Coordinates that are derived from the NAD-27 datum may differ by as much as 100s of feet from the same location expressed in WGS-84 or other, more recent datums. It should go without saying that importing data from two different datums could result in improper registration of one set of data compared to the other – and both would be improperly located if the underlying GIS maps were based on a third datum.

Once the datum is selected, a map projection must be chosen. A map projection is the mathematical formula that transforms the 3D location values onto a flat surface. Imagine a glass globe – but one using a datum that more nearly reflects the real shape of the earth than a simple sphere – with lines inked on the surface and a bright light at the center. Depending on where you hold a piece of paper and how you shape it, the lines will be projected on the paper in different ways. Common projections are Mercator, Lambert, and Albers; there are many, many others. All are ways to try to make a flat map that properly reflects the three-dimensional earth; so all must fall short in one way or another.

The final element in the mix is the coordinate system. For many purposes geographic locations are expressed in latitude and longitude, though latitude and longitude are not a coordinate system in the normal sense of that term. Commonly a projection and a coordinate system are linked. The common "UTM" (Universal Transverse Mercator) combines a Mercator projection with a specific coordinate structure. Others in common use in the US are the various State Plane Coordinate Systems. There are literally hundreds of datums and thousands of combinations of datum, projection, and coordinate system.

Most archeologists need not deal with these matters because most archaeological grid systems, particularly for individual sites, don't have to deal with projections, datums, or even – in some cases – coordinate systems that apply beyond the project. If the area mapped is small (relative to the shape of the earth) then the coordinates "on the surface" are essentially identical to a flat surface. Cover a larger area, however, and the curvature of the earth and its relief, come into play. Then these issues become important. These issues have created considerable confusion when an archaeological team is using a GPS receiver and finds out that the coordinates it produces cannot be reconciled with those on a map. Most GPS receivers produce their readings in WGS-84. If the maps in use are based on a different datum, the coordinates for the same place will be different!

So long as the area covered is small, it is also possible to work with a local coordinate system (a Cartesian grid, most often) and then to move everything to a real-world coordinate system when desired. The key values needed are the point of origin of the site grid and the orientation of the grid relative to the projection and coordinate system that is to be used. Unfortunately all is not so simple in this process. The different map projections and coordinates systems all have some element of distortion inherent in their structure. (If that were not the case, of course, there would be a single projection and coordinate system in use.) The smaller the area on the earth that is included in the projection's area the less this distortion. The easiest way to see the possibility for distortion is simply to look at a Mercator map of the world. Greenland is shown on this map as being as large as South America! (It is actually less than one-eighth of South America's size.) The distortions in other maps are usually less obvious, but all maps must have some form of distortion because of the twin problems of mathematically modeling the too-complex shape of the earth and of projecting that complex shape onto a flat surface.

A common projection used by many archeologists is the UTM. The UTM system divides the world into sixty 6-degree "zones." This projection is useful for maps covering moderately large areas such as a US state or a British county.

An in-depth, web-accessible discussion of *datum*, *projection* and *coordinate system* is part of the Geographers Craft web site developed by Peter Dana that can be accessed at www.colorado.edu/geography/gcraft/notes/coordsys/-coordsys_f.html.

Commercial GIS systems

There is a wide range of different commercial GIS software and some open source programs. Among the popular vector packages are Environmental Research Systems'® (ESRI) ArcGIS® (www.esri.com), Intergraph's® GeoMedia® (www.intergraph.com), MapInfo's® MapInfo (www.mapinfo.com), and Laser Scan's® Gothic® (www.lasserscan.com). Popular raster GIS programs include Clark Lab's IDRISI (www.clarklabs.org), Keigan System's® MFWorks® (www.keigansystems.com) and its GeoMedia GRID® version).

ESRI's raster GIS is an extension to its ArcGIS package called Spatial Analyst®. The GRASS open source raster GIS is available for free download (grass.itc.it/) and has played an important role in many archeological GIS studies. Many remote sensing software packages also serve as capable raster GIS software, these include Leica's® ERDAS® (gis.leica-geosystems.com), PCI's® GeoMatica® (www.pcigeomatics.com), Research System's® ENVI® (www.rsinc.com).

GIS and Databases

Now that we have the geographic component of our system we need our database component. We have discussed databases at length in Chapter III. In a GIS database we can store attributes (columns) about each of the geographic entities in the map representation and link them with a key. Our parcel database might have columns for the parcel value, owner's name, etc. In a GIS the additional element is the linkage between the geographic/map entity and the database attributes. Each parcel might have a parcel number, and that parcel number can be used to link the geographic component to the attributes. As we will see, we can use the standard database operations to perform queries on the attributes of the GIS data and use the specialized geographic query and analysis features to query and analyze the geographic data. The latter are the queries and analyses made possible via the system's understanding of topological relationships – e.g., what geographic entity lies adjacent to, inside of, or across another geographic entity.

We can consider a GIS as a form of a database that contains spatial data; a GIS extends the database by adding geographic (or geometric) data. Two rows of a simplified (vector-based) geodatabase for a municipality's parcels might look something like this, with parcel ID, address (street only on the assumption that the municipality does not need its own name), zip code (possibly not needed in the case of a single municipality), the name of the responding fire department, and the geometry of the parcel:

Parcels Table

Parcel ID	Address	Zip	Fire Dept.	Geometry
12	12 Main St.	12345	Central	x1 y1, x2 y2, … xn yn
13	14 Block St.	12346	Central	x1 y1, x2 y2, … xn yn

The coordinates that define the boundaries of the parcels are stored in the geometry column as a set of x, y coordinate pairs (or x, y, and z if the system can accept 3D data) defining the vertices of the polygons bounding the properties. The precise, formal, term for this type of database is *object-relational* since the geometry is not a single value but a set of values – an *object* in database terminology. (Given our work with databases in the last chapter, many will have seen that the best database design would separate the geometry from the textual information about the parcels, leaving two tables, linked by the parcel ID, one with the boundaries of the property and the other with the other attributes.)

To make this approach effective it is necessary either to add geo-processing operations for topological relationships to the standard SQL operations in the database (as done in Oracle® Spatial®) or to use separate GIS software for the topological relationships (as is done in ArcGIS). In the former case, adding topological relationships to SQL operations permits a geographic/topological query to be performed within the database (e.g. "is x within the boundary of y?") along with standard SQL operations. Added SQL operations would, for instance, permit a direct query of the database to find all features of one type falling within the

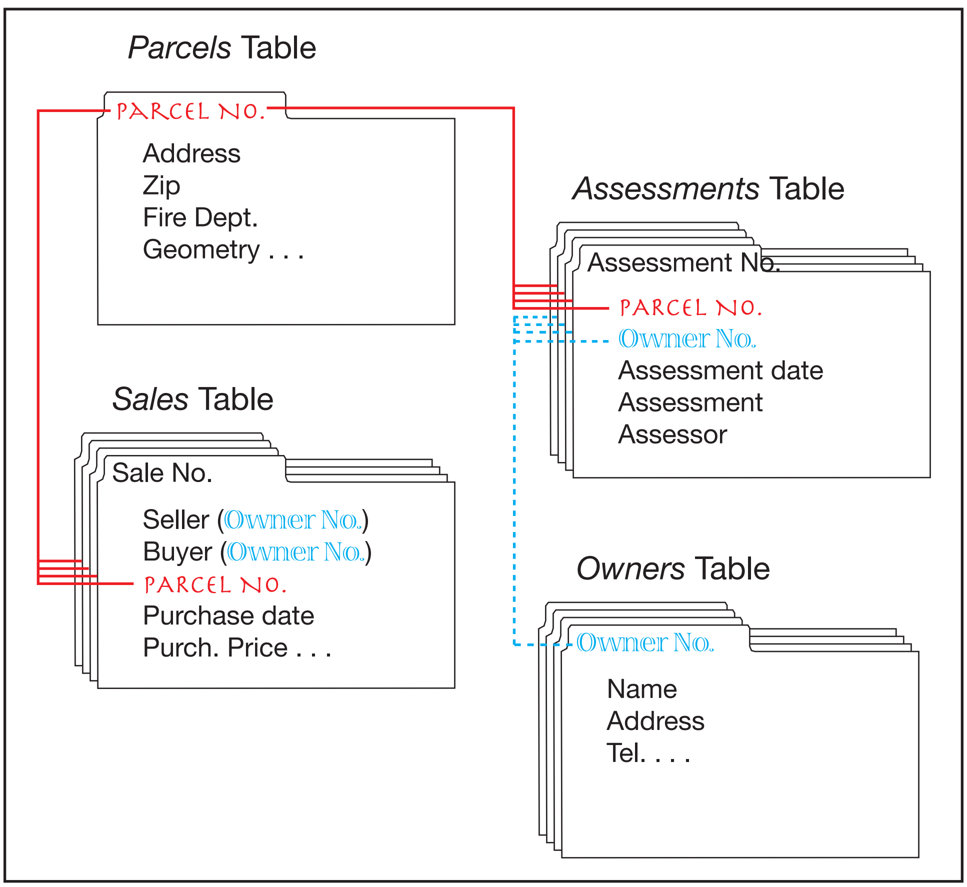


Figure 10

A moderately complex community database with tables for parcels (geometry in vector form), owners, assessed values (many for each parcel, since parcels are re-assessed on a regular basis), and sales (again many rows for the same parcel since each may be sold many times.) Not all links or likely tables are shown.

boundaries of any feature of another type, e.g., all kilns within features defined as courtyards.

The alternate approach requires two sequential processes. First, SQL (or equivalent) operations extract the entities from the database based on attributes; then the topological relationships are queried with separate GIS software. This process would first get from the database geometries of all entities of one feature type (kilns) and then all the geometries of entities of the second type (courtyards). Then routines supplied as parts of the GIS package would calculate which kiln geometries are within courtyard boundaries.

For raster GIS systems any data table is linked to the map cell-by-cell. The value stored in the map/file for each cell will normally link to the primary key

The evolving nature of GIS

Unlike database management systems, GIS software is a rapidly evolving area of computing; so the subject of this chapter is a bit of a moving target. In fact, there is a considerable debate about what to call these systems. GIS used to be the commonly accepted term but today you will see a wide range of terms used including *geospatial systems*, *geo-information systems*, *spatial information systems,* and others. This confusion of terms reflects the rapidly changing status of the area.

Most early GIS software basically coupled CAD or CAD-like graphical files to a database. Early examples included systems like Arc-Info® from ESRI and MGE® from Intergraph. The early ArcInfo systems had a database (called Info) with most of the capabilities that have already been described in the database chapter. Attached to the database was a specialized file that held the geographic data. Together these were called a coverage and were stored in the computer in a complex file directory and subdirectory structure. Specialized programs allowed the creation (digitization) of the geographic data, its editing, and its analysis. It was not uncommon for the linkages between a specific geographic element and its attributes in the database to be broken. In the Integraph MGE software the graphic files were DGN files created and maintained by the MicroStation CAD software. ESRI later introduced the ArcView® software family. In ArcView the geographic information is stored in a shape file while the attributes are stored in a database table. A single layer (e.g. sites) in ArcView would consist of a shape file and a data table – as well as some auxiliary files. The MapInfo GIS package has a similar structure with its TAB and MIF/MID formats. A TAB file is actually a directory with a set of files with extensions .TAB, .DAT and others. The MIF/MID MapInfo format refers to a file structure that can be used by other software – a kind of an export format.

In the mid 1990s a new form of GIS software became common. This includes the geographic data as a single column in a relational database. One of the first companies to popularize this structure was Intergraph with its GeoMedia software. There had been earlier systems (even from Intergraph) with this structure but they had limited impact. Because the geographic data and the attributes are stored in a single table it is much easier to keep these together and organized. The vendors still had to develop specialized programs to create, edit, and analyze the geographic part of the data but were able to use standard database operations for all the other needed capabilities. In fact, Intergraph's first version of GeoMedia simply used the standard Microsoft Access database system for both attribute and geographic data. After Intergraph's release of GeoMedia, ESRI also developed a relational database version of its software – now renamed ArcGIS. The structure was called a geodatabase by the ESRI marketing people.

Both ESRI and Intergraph now permit the use of databases from MS-Access, SQL Server, Oracle, and other database management systems.

While the GIS companies were moving to adopt a database format some of the database companies were extending their database to include geographic data and geographic operations. The most aggressive database company in this area has been Oracle. The standard Oracle database now has the capability of store and manipulate/analyze geographic data within the database. Stay tuned. More change is surely coming.

in a data table. Thus, any cell(s) with a value of *12* would have a *12* stored in the proper place in the map/file and be associated with the *Parcels* Table via the *12* as a primary key; via that link, of course, the other attributes of this specific parcel – its address, zip code, fire department, and the like –are also linked to the appropriate cell(s) on the map. Therefore, a raster version of the *Parcels* Table shown above for a vector GIS would be the same except that the geometry would not be included. The geometry would be supplied by the raster cells in the map file. (Using the preferred design of a separate parcels text table and a map table, as described above, would yield a *Parcels* Table for the vector map identical to the *Parcels* Table for the raster map.)

A more archaeological category of information might be soil types. Part of such a *Soils* Table might look like this:

Soil Type	Soil Name	Depth to Bedrock	Permeability	Type
1	Blakeley	40 cm	3	silt loam
2	Smithville	120 cm	4	clay
3	Jones	40 cm	6	loam

The values in the Soil Type column would provide all the potential values for the individual raster cells of a map or the polygons of a vector map – so a cell/polygon containing Soil Type 1 would have a 1 in the proper location in the map file for that cell/polygon and so on; this is the primary key in the *Soils* Table and effectively a foreign key in the map. Using the map alone, any user could see all the cells or polygons with any specific soil type and, adding information from the

Multi-File GIS Data Structures for Vector Data

In the original versions of many types of vector GIS, there were two file systems, one with the geometries of the features and another with the attributes of those features. In these systems a file might have many sets of X, Y strings stored in it that defined, say, soils polygons or parcels or water lines. Each soils polygon or waterline segment had a specific code for identification. In a separate file there were attributes stored, and each geometric entity in the graphics file had a related record. Thus, polygon 13 in the graphics file might be linked to a record in the data table with soils properties. Or line segment 356, representing a water line, might have attributes that recorded pipe diameter, pipe age, and the like.

A common example of this type of data is the shape file used by ArcView software. In this structure there are at least two files, one with the .shp extension and one with a .dbf extension. A soils example might be soils.shp and soils.dbf while our waterline example might be represented by waterline.shp and waterline.dbf. These .shp files stored the geometries (X and Y pairs) while the .dbf files stored the attributes. Both were needed to create the complete data set. Each stored the crucial attribute, the parcel number, which might be viewed as a primary key in the .dbf file and a foreign key in the .shp file.

In practice there is one additional required file and some optional additions to the .dbf and .shp files. The required file is an .shx file that contains a spatial index to the geometries and permits more rapid access. A common, but optional, file is the .prj file. This file provides for the projection information for the geometries. If the .prj file is absent, the data will only "fit" with other data if it happens that both used the same projection system.

The shp/dbf/shx file system had a number of technical problems, and it has largely been replaced. Within the last few years the creators of databases have found ways to allow the storage of multiple geometries within the database. As a result, it is no longer necessary to have two files and the problems that can create. Instead, a single column in the data table is used to store the geometries for each feature. The popular GIS software, ArcGIS uses this format in its "geodatabase," as do many others, including the Oracle Spatial database system.

data table, such a user could know where Blakeley silt loam lies within the study area – or Smithville clay or soil with 120 cm. depth to bedrock or soil rating 6 for permeability. By linking the database to the raster map, much more can be done.

Once the data table and the raster map can be used together, a process called *reclassification* can be used to make a new map for easier use. If there were a need to examine depth to bedrock in our example, for instance, a user would need not only to know the depth for each soil type but also to remember that Blakeley silt loam and Jones loam have the same depth. So a map showing only the depth of the soil to bedrock would be useful. Given the map-to-data-table linkage, making that is a simple procedure. Using the Soil Type as a primary key in the *Soils* Table and a foreign key in the map file, a direct replacement of the map's cell contents (Soil Type) with the related depth to bedrock is possible. Saving the result as a new map file yields a map with *40* in all cells where *1* occurred in the original map and in all cells where *3* had been. The cells with *2* in the original map would now have

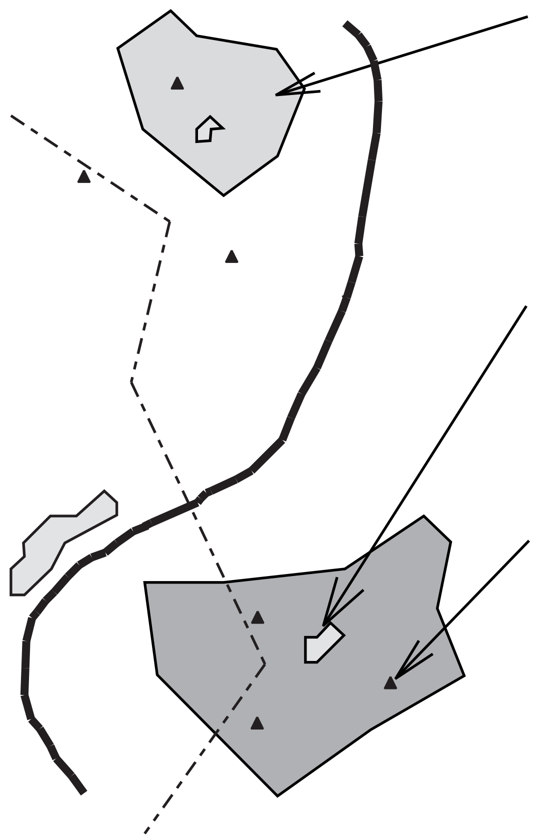
CAD and GIS – Some Differences

The next chapter of this book deals with CAD. Since CAD is also used to make maps, however, it is important here to define some key differences between CAD and GIS. A primary difference is that a GIS supports operations based on topology. In a CAD drawing we can visually see that a waterline crosses a parcel. That information is provided to us by our vision, but it is not a data item "known" by the CAD software. As a result, there is no computer operation in a CAD system that would "find all the waterlines that cross parcel 1." However, this type of operation is integral to any GIS.

Another difference lies in the way a user defines lines in CAD and GIS. A vector GIS requires either points, lines, or closed polygons. CAD systems permit disconnected lines that APPEAR to make closed polygons but are not. CAD drawings can therefore, when brought into a GIS map, be very disruptive if they include lines that appear to be, but are not, a closed polygon. This may best be explained with an example. In a CAD system a simple 2 m. x 2 m. excavation unit might be drawn as four straight lines that visually connect at the corners; that is not proper CAD procedure, but it could be done; it would be adequate for most maps and illustrations. In a GIS the same excavation unit must be represented as a closed polygon. Two problems may result from using improper constructed CAD entities. First, improperly drawn CAD lines may appear to be connected when they are not (the user may need to enlarge the image greatly to see that they are not). Second, even if they are connected, they may not compose a closed polygon. In either case, the lines, when moved into a GIS, may not be recognized as a closed polygon. To insure that the excavation unit is properly processed, the boundary should be made as a closed polygon. (A GIS may be able to import multiple lines as a closed polygon if they are truly joined, but virtually all CAD programs permit the creation of a closed polygon, though the name may be different from program to program. The CAD entity bound for a GIS map may be a line or a closed polygon but not an open or improperly connected polygon.)

Many of the modern GIS systems now allow the GIS to connect directly to a CAD file and use or extract the data. There are also specialized software packages, such as FME® from Safe Software, that are designed to convert CAD data to correctly formed GIS data. In all these cases, however, if serious thought were not given to organizing the data from the very beginning to meet future GIS requirements, the process of CAD to GIS conversion could be very painful and time consuming.

An unrelated issue that can arise in the movement of CAD data to a GIS is the coordinate and projection system used. There are many archaeological projects where an arbitrary grid is used and the geographic location of the site or project grid's origin (the zero-zero point) is unknown. Unless the grid can be linked to real-world coordinates, it will be impossible to link existing map data to the archaeological data. Important for larger-scale projects, CAD systems assume a simpler Cartesian grid, with x, y, and z. As noted above, this is not an adequate approach to deal with the surface of the earth.



Surveyed_area Table

Survey_ID	Surveyor	Date	Survey Type . . .	Geometry
123	Smith, D.	12 Oct 2004	Intensive	x1 y1, x2 y2, . . . xn yn
124	Jones, R.	3 Jan 2005	Intensive	x1 y1, x2 y2, . . . xn yn

Site Table

Site_ID	Surveyor	Size . . .	Date	Geometry
23	Smith, D.	2300	15 Oct 2004	x1 y1, x2 y2, . . . xn yn
24	Jones, R.	4500	6 Jan 2005	x1 y1, x2 y2, . . . xn yn

Finds Table

ID	Surveyor	Item_class	Date . . .	Geometry
134	Smith, D.	Ceramic	14 Oct 2004	x1 y1
135	Jones, R.	Axe	5 Jan 2005	x1 y1

Figure 11

Schematic of a vector-based GIS data set, with tables for survey areas, sites, and finds.

120. This *reclassification* yields a new map with cells containing data expressed just as needed for a specific problem. (Note that reclassification removes the data table primary key from the cells, replacing it with the value of another column. However, the original file is not changed; it is retained, and a new table with the reclassified data is created. The original data file must remain intact and inviolate.)

Two Simple Archaeological GIS Cases

A Vector GIS Archaeological Example

Our first example will be one that uses a vector GIS to track and manage the results of a regional archaeological survey. In the GIS we can suppose that we have information on the local area including roads, property ownership, and similar information. There is mapped environmental data detailing the soils of the area and the agricultural fields and their crops as well as maps showing the locations of rivers and streams. From the archaeological team there are records of findspots of individual artifacts, areas surveyed, and site boundaries. (We ignore for the moment the question of what is a site.)

We will assume that the background information on the area has been obtained from some public or commercial sources. Our information on the location of finds can be derived with a number of strategies. The archaeologist going out in the field today may very well have a GPS system. The more sophisticated GPS units have the ability to record the locations of points and even polygons and to download these into a GIS. We will revisit the issue of GPS and GIS later. For many areas or projects, however, there is probably much information on findspots, areas surveyed, and the like that has been previously recorded on existing paper maps or aerial photographs. In such cases it will be necessary to digitize the locations.

Whether we are using a GPS receiver or other sources, it is essential that we develop and carefully use an annotation system (often referred to as a data dictionary) that allows us to link database information to the geographic coordinate information. In the database chapter the concept of a primary key was developed. We can think of the geographic data in the GIS as simply a table in the database, and it is critical that each geographic element be properly linked with its attributes via a primary-key-to-foreign-key link. Such links in an archaeological system are commonly things like site, feature, and lot/locus numbers.

Each category of information will be structured in the GIS – as in any database – as a separate portion of the whole, generally speaking, a separate table. Various GIS packages call these portions layers, themes, or feature classes. A layer, theme, or feature class in a GIS is equivalent to a table in the database. (We will use the term *layer* here.) One layer might be the areas that have been surveyed, another might be finds (including locations, of course), while another might be the various sites. The finds and sites are two different layers in this example because it is likely that they have different types of data recorded about them. It is commonly the case that different layers have different attribute structures, different table designs. A table about surveyed areas would have the date of the survey, who conducted the survey, the funding, or the permitting source. A table about finds would need to contain different information. (See figure 11.)

Assuming that we have all the data entered into our system, the value of the GIS approach becomes clear since we can begin to ask a wide range of questions because of the relationships between/among the maps and the data tables. The simplest are select statements (shown here in English, not as SQL statements) which are equivalent to a select statement in a database, but, in this case, the select statement returns a geographic element. Some simple select statements might be:

• Select (and display) locations for all finds that are stone hoes. (One table will provide this information – the point location associated with each hoe in the *Artifacts* Table – and the points will then be plotted on a map.)

• Select (and display) all the river valley soil polygons. (A single table is sufficient here too. The polygons in the *Soils* Table categorized as river valley soils are selected and plotted on the map.) See figure 12.
• Select (and display) all boundaries of all sites that have architectural features determined to have been hearths. (This will require joining two tables, a *Features* Table and a *Sites* Table. The *Features* table will be searched for all hearths, and a foreign key in each row will provide a link to the correct site in the *Sites* Table via the primary key in that table. The site polygons will have been defined in the *Sites* Table, not the *Features* Table.)

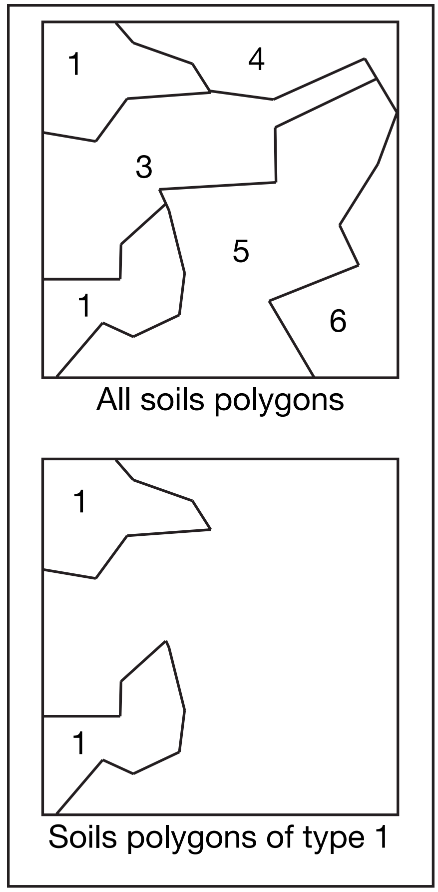


Figure 12
Soils of type 1 as the select set.

The geographic features that meet our selection criteria are commonly called the *select set*. We can use the select set approach to create a number of valuable map products even without utilizing any advanced analytical operations. Using attribute selections, we can create a range of thematic maps. Perhaps we were interested in looking at the relationships of prehistoric agricultural sites to the potential agricultural productivity of soils. In most soil mapping there are many soils polygons linked to an extensive data table of soils properties via the primary key in the *Soils Properties* Table and its companion foreign key in the *Soils Polygons* Table. The soils properties commonly include factors such as soil texture, depth to bedrock, productivity and the like. As a result of the connections of data tables to one another, each polygon describes an area with a variety of attributes, and we might want to create a map for one or more of those attributes. To do so, we might perform the following operations:

• Select soils polygons where the attribute soils productivity is "high" and call this "good agricultural soils." (This requires joining two tables, the *Soils Properties* Table and the *Soils Polygons* Table and selecting, for those soil types that are rated as highly productive, the polygons defining the boundaries. The select set would be placed in a new table, one that may be stored as a named table or used only until the session is concluded.)
• Select soils polygons where the attribute soils productivity is "moderate" and call this "moderate agricultural soils." (This would also require joining two tables to create the select set, and another new table would result.)
• Select soils polygons where the attribute soils productivity is "low" and call this "poor agricultural soils." (Another two-table join and another newly-generated table).
• Create a map where the polygons from the "good agricultural soils" table are deep green, those from the "moderate agricultural soils" table are pale green, and those from the "poor agricultural soils" table are brown. Print out a report showing the amount of the study area in each category and its percentage of the area. The map and the report will effectively join the three new soils tables created in this process.

All the data actually originated in just two tables, the *Soils Properties* Table and the *Soils Polygons* Table.

We can create select sets using attributes from the data tables, as we did above, or we can increase the complexity of our analysis by using the GIS capability for topological operations as well. One of the common geographic operations is called an *overlay*. In a basic overlay operation geographic elements from one layer (or

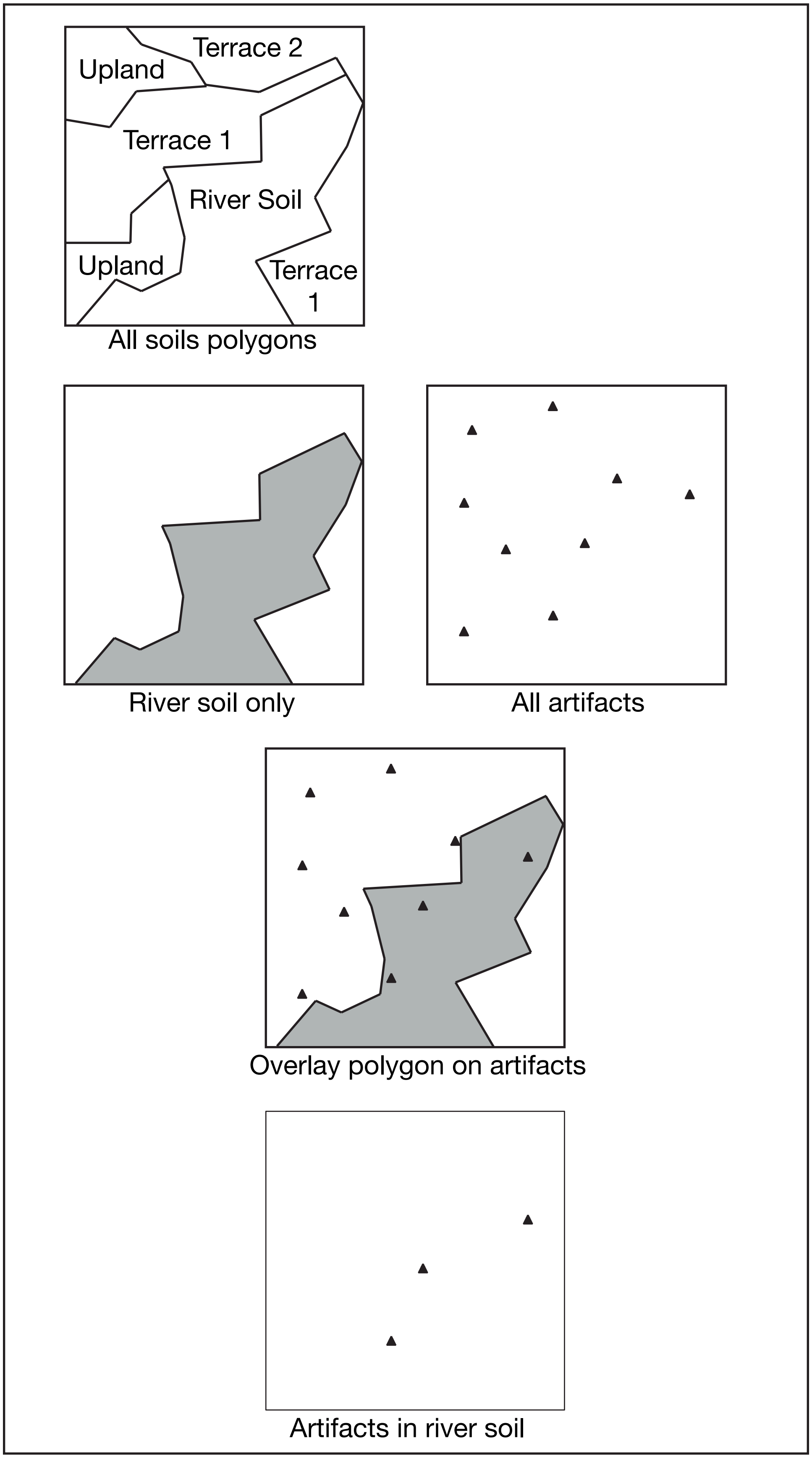


Figure 13
Process of overlaying artifacts on river soil, culminating in a map with only the artifacts found in river soil.

select set) are placed "over" geographic elements from a second layer and only the areas that are common to both are selected. Thus, the overlay selection is based on topological properties rather than primary-key-to-foreign-key relationships. A common operation using an overlay would be:

- Select the "river soils" polygons from the soils map.
- Overlay the finds locations on the soils polygons.
- Select those finds that are "on" river soils.

In this example the locations of the finds that were found within the river soils polygons would be the select set. (See figure 13.) The topological relationships have now come firmly to the fore, and simple database queries via SQL statements are no longer sufficient to provide answers to important questions.

In many overlays (and in many other operations) the vector GIS actually creates new geographic elements. Essentially a new geographic element is created from the vertices of the features from each layer in the overlay. Suppose we performed the following overlay operation:

- Select the "river soils" polygons from the soils map
- Select the "pasture" polygons from the land use map
- Overlay pasture polygons on river soils polygons (Note that the process here finds only that portion of the two sets of polygons that are common to both – in set theory terms, an intersection, not a join.)

In this case entirely new polygons would be created that combine vertices of the pastures and vertices of the river soils. There may even be entirely new vertices created by the overlay, as in this example. New vertices are created at points where the two polygons intersect. Figure 14 shows the steps and results of this process.

More complex operations can be built up in steps. Thus, the pasture and river soils areas might, in turn, be overlaid on sites polygons or – probably more useful – combined with a map of all sites so as to display which sites fall in contemporary pasture/river soils and which do not to determine if the sites are located on or near good agricultural soils.

A GIS can easily calculate areas and percentages. For example, we can ask:

- What is the area (or percent) of the project that has flat river valley soils? or
- What percentage of sites are located on these soils?

Area calculations would be based on areas of polygons, and percentages would as well. (In a raster system, on the other hand, areas and percentages would be calculated by starting with the area represented by an individual cell and multiplying by the number of cells in a given category.)

Topological Relationships and Overlay Operations

Prof Max Egenhofer (Egenhofer et al., www.spatial.maine.edu/~max/RC3.html "A Topological Data Model for Spatial Databases," Symposium on the Design and Implementation of Large Spatial Databases, Santa Barbara, CA, *Lecture Notes in Computer Science*, Vol. 409, 1989) has formally defined the different topological relationships that may exist in a vector GIS. These include relationships such as "touch", "inside", "contains" and six others. The great majority of commercial vector GIS systems utilize these operations.

Prof. Dana Tomlin (*Geographic Information Systems and Cartographic Modeling*, 1990) has published a similarly seminal work on the types of relationships that exist in the raster domain. These are referred to as local, focal and zonal operations depending on whether they apply to a single cell in a raster data set (local), areas of cells around a single cell (focal) or areas of cells with similar values (zonal).

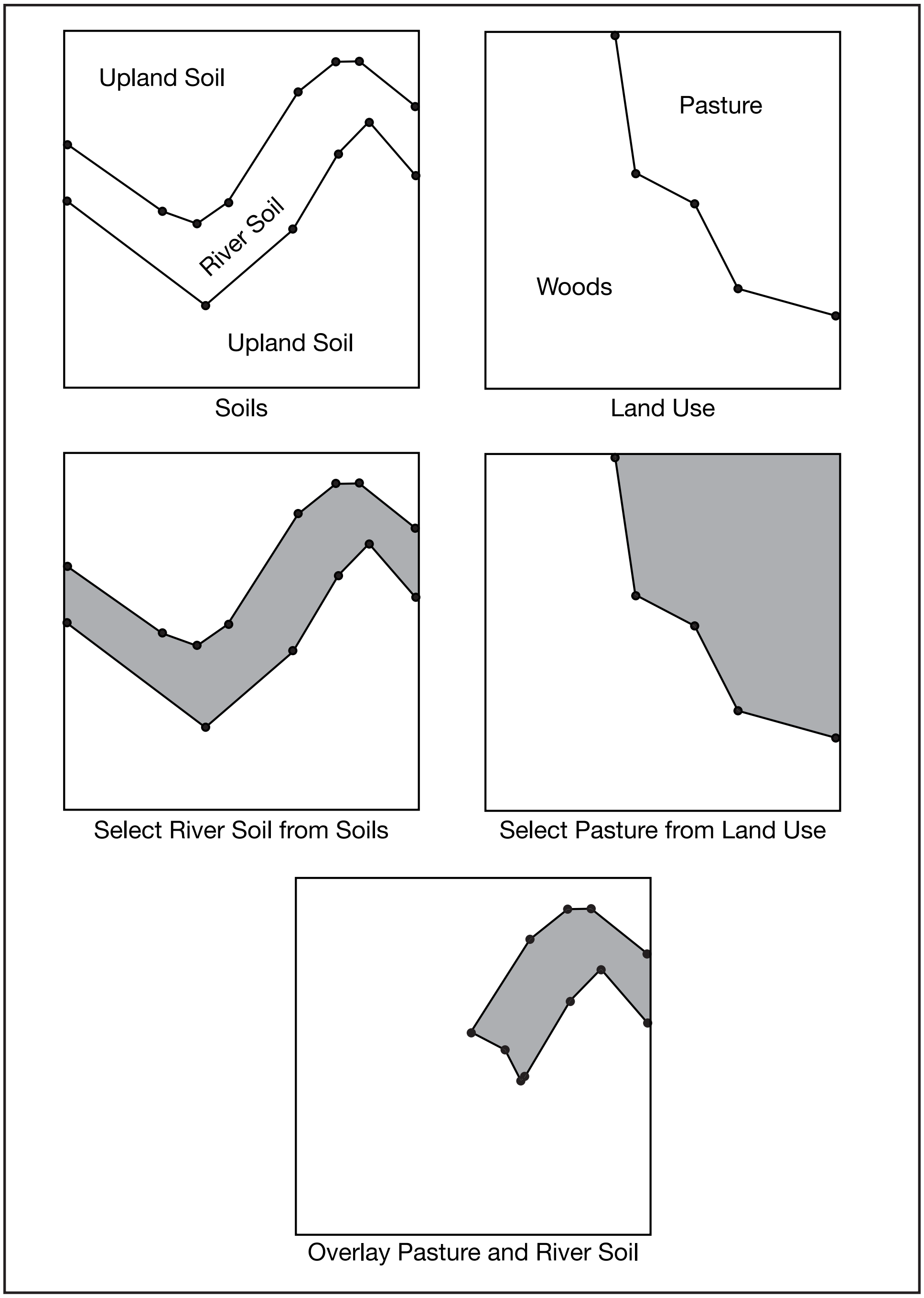


Figure 14

Process of overlaying pasture and river soil. (Filled circles are vertices defining the polygons.)

We can concatenate various selection criteria (both attribute and geographic) and ask questions such as these:

- Show all the sites that have surface evidence of hearths and have been found in fields that are pasture and are on river soils.

Note that, as with any database query, it may be important to put the select statements in the proper order to obtain the "correct" answer. In this example, all the conditions are expressed with *and* so the order is not critical. Were there *or* conditions, however, order could be critical, changing the outcome by changing the order of the processes.

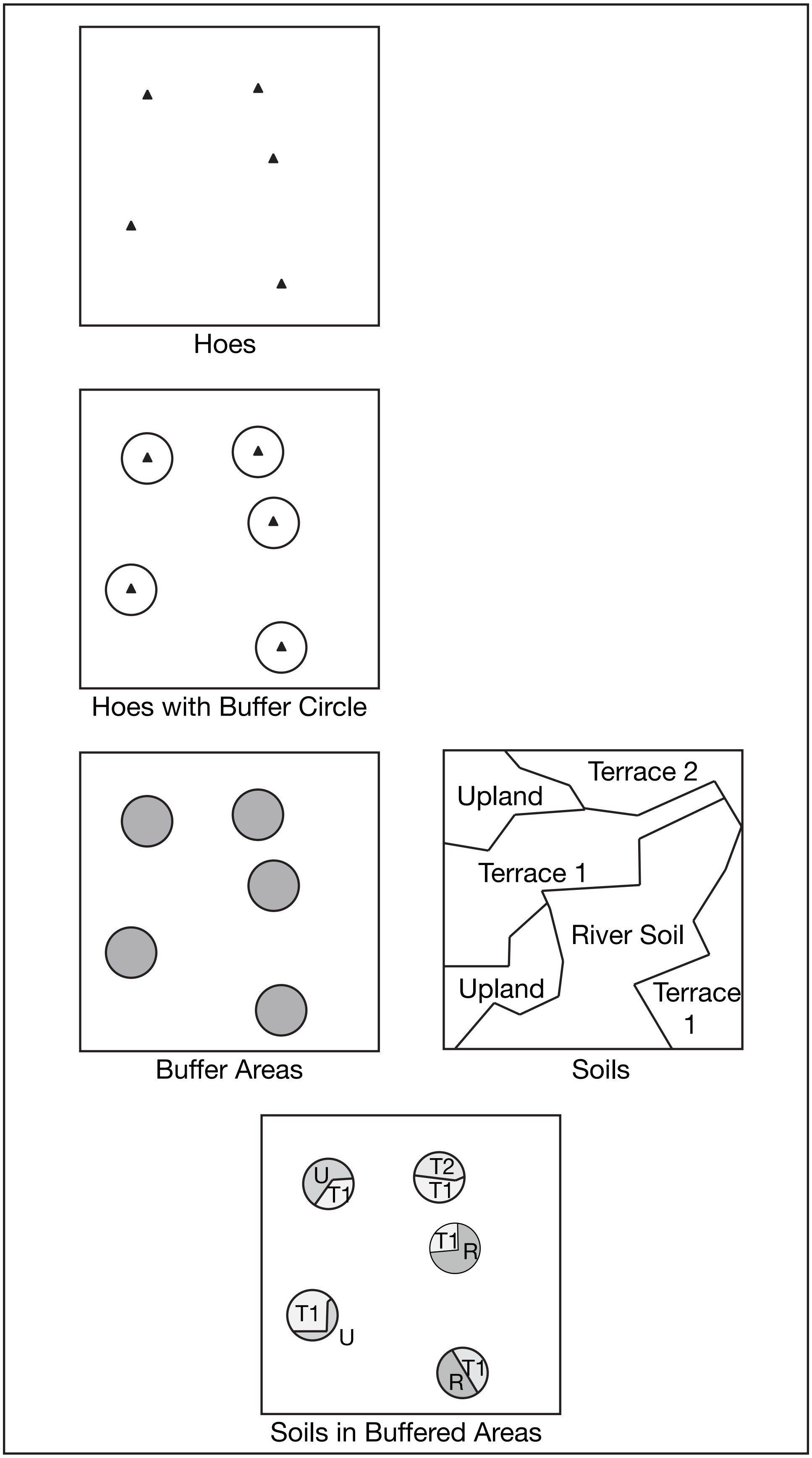


Figure 15

Process of overlaying buffered findspots for hoes and soil types.

There are a number of variations on the overlay operation, and combining it with other, more sophisticated analytical operations let us investigate more complex questions. A common GIS operation to approach more complex questions is called buffering. In the simplest case a buffer is one or more concentric rings around a single x, y point. For example we might be interested in the possible relationship between the locations where a particular agricultural tool (say a stone hoe) has been found and the area's soil properties. To answer the question we would begin by selecting the findspots for stone hoes and proceed, step by step:

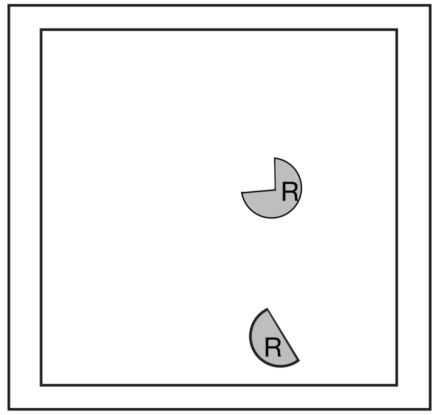


Figure 16
The buffered areas of the hoes – as in figure 15 – but overlaid on river soil rather than all soils.

- Select all finds locations for "stone hoe."
- Buffer each hoe location some distance (say 500 meters) to create a new layer with circles of 500 m. in radius around all the locations.
- Overlay this layer with the soils layer:
- Report the areas (percentages) of each soil type in the buffered area. A mapped version of the process is shown in figure 15.

If we were interested in just the areas around the hoes that were river soils we might first do a select on the soils:

- Select all soil polygons that are "river soils."
- Overlay these with the stone hoe buffer areas to see the areas that are within 500 m. of the hoes and were "on" river soils (result shown in figure 16).

Buffering can also be done around linear and polygonal features. We could buffer the streams in our study area, perhaps 100 m., and then overly that result with find spots or site polygons. Suppose we believed that sites of a particular type were to be found in closer proximity to streams than would be expected by chance. A processing sequence might be as follows

- Select all sites that have our desired attribute(s).
- Buffer all streams in a set of relevant distances, perhaps 0-50 m., 51-200 m., 201-500 m., 501-1000 m., and greater than 1000m.
- Report the areas and percentages of the study area within each category.

Lets suppose that 10 % of the area was within 50 m. of a stream, 20% 51-200 m. away, etc.

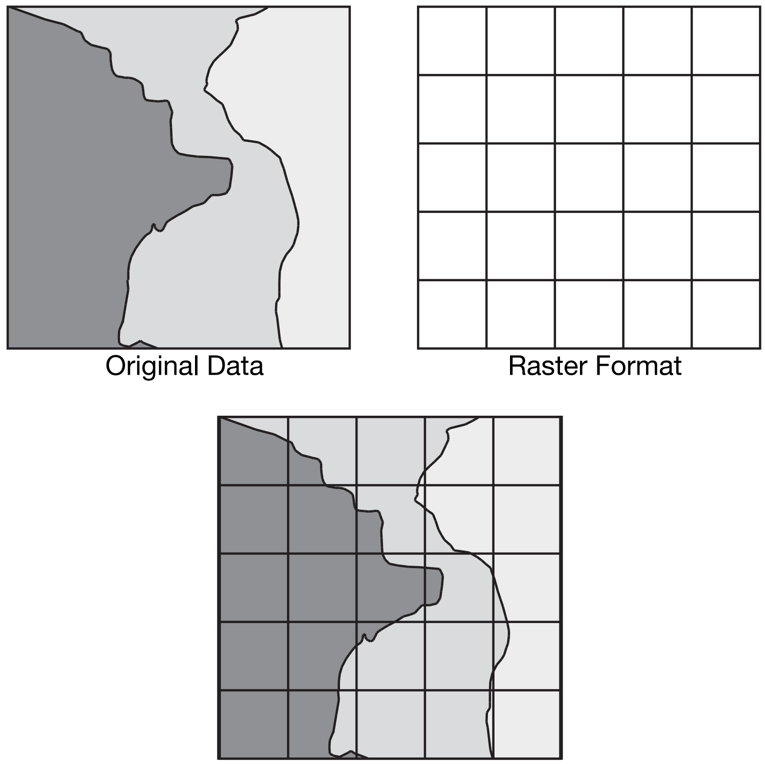
- Overlay the selected sites on the buffer map to learn how many sites were located in each category.

Suppose that 50% of the sites were in the 0-50 m. category where only 10% of the area lay. This would tend to support our idea that sites were located close to streams.

We can show the added value of the GIS by using it to uncover a bias that might have crept into our survey work and affected the question of sites located near streams. We can overlay the surveyed areas map onto the stream buffer map and locate those areas that coincide. Suppose that we find that 60% of the total area we surveyed was within the 0-50 m. buffer zone. Now the fact that 50% of the sites were in this category is no longer evidence for a preference for locations near streams but may be simply a result of the fact that we spent a disproportionate amount of our survey efforts in the areas near streams.

It is also possible to use buffering on polygons. An example might be to buffer a site polygon by 20 m. to create a map for developers showing them the protection zone that they should honor while working in the area.

We will return to the vector case study later but will now turn our attention to a raster example.



1	2	2	3	3
1	1	2	3	3
1	1	1	2	3
1	1	2	2	3
1	1	2	2	3

Resulting Numeric Values

Figure 17
Raster data resulting from conversion of originally vector-based data to cell-based data.

Raster GIS Operations

It is possible to perform many operations, selects and overlays for instance, in both a raster and vector context. A *select* operation on a raster data set will return a new raster layer with only the cells that meet the selection criteria. A raster overlay, for example, returns a selection set that consists of cells not polygons or points. Virtually all operations are on cells. (See figure 17.)

One raster example might be:

• Select cells from the elevation map that have elevation values greater than 1,000 and call the map of these cells “High” (“0” otherwise).
• Select cells from the slope map that have slope values greater than 20% and call the map of these cells “St” (“0” otherwise).

As you may have guessed, those two select sets provide an ideal beginning for an overlay:

- Overlay the map "steep slopes" on the map "high places" creating a new layer and call it "high and steep." (The new layer is actually a new raster map file, this one with 1 for the cells that are both high and steep, 0 for the others.) Note that this process cannot produce anything comparable to the new vertex that might be generated by a vector system; each cell is either in the overlay or not. (See figure 18.)

An alternative approach similar to *select* that is frequently used in raster operations is called *reclassification*. As discussed earlier (pp. 120-122), a reclassify operation places new values in cells via an equation or algorithm. The result is a new map with the replacement values; the original is left unchanged. Thus, the operation just completed on the elevation and slope maps could have been done in this

1020	1010	1010	1000	1000
1030	1020	1010	1010	1000
1030	1020	1010	1010	1000
1030	1030	1020	1010	1000
1030	1020	1020	1020	1010

Elevation Map

25%	10%	10%	25%	10%
35%	25%	10%	10%	10%
30%	25%	10%	20%	10%
35%	25%	25%	25%	10%
35%	20%	10%	25%	10%

Slope Map

High	High	High	0	0
High	High	High	High	0
High	High	High	High	0
High	High	High	High	0
High	High	High	High	High

High Places

St	0	0	St	0
St	St	0	0	0
St	St	0	0	0
St	St	St	St	0
St	0	0	St	0

Steep Slopes

1	0	0	0	0
1	1	0	0	0
1	1	0	0	0
1	1	1	1	0
1	0	0	1	0

High and Steep

Figure 18

Raster data from elevation map and slope map creating a new "High and Steep" map.

1	2	2	3	3
1	1	2	3	3
1	2	1	2	3
1	1	2	2	3
1	1	2	2	3

Soil Types

12	25	25	35	35
12	12	25	35	35
12	25	12	25	35
12	12	25	25	35
12	12	25	25	35

Soil Depth to Bedrock (cm.)

Figure 19

Soil Types map reclassified as a map of the soil's depth to bedrock.

way via reclassification:

- Reclassify all cells from the elevation map that have elevation values greater than 1,000 as 1 and reclassify all others as 0.
- Reclassify all cells from the slope map that have slope values greater than 20% as 1 and reclassify all others as 0.
- Overlay the map "steep slopes" on the map "high places" creating a new layer . . .

Multi-Valued Cells

Implicit in much of the discussion has been the notion that any data table attaching physical characteristics to a cell or a polygon may link only one value to that cell or polygon. This is less a problem with polygons; they are normally constructed to match natural boundaries. Cells, being more arbitrary portions of the terrain, cannot be limited to a single value; we all understand that the real world does not work that way. So it is standard to determine the value of a cell with some systematic approach (value in a specific location, majority value, etc.). Thus, a cell may be labeled as representing an area with flat river valley soils, but that may, in reality, mean only that the soils in the upper, left corner of the cell were flat river valley soils or that the majority of the soils in the cell were flat river valley soils. It certainly does not mean that the cell contains only flat river valley soils. Similarly, the boundary between pasture and woods will not fall neatly along cell boundaries; some systematic approach will be required to label cells with one or another value, despite the reality that many cells have both pasture and woods. (See figure 17.)

There is another way to deal with the problem of multiple cell contents, though it is complex and, as a result, little used. Taking as an example the choice of pasture or woods, cells with only pasture may be labeled "1," cells with woods labeled "2," and cells with both labeled "3." That system can be expanded, regardless of the number of basic choices, so that any number indicates a unique combination of categories. This can be accomplished by using a number for each basic category based on the powers of 2 (1, 2, 4, 8, 16, . . .) and simply adding together the values of all the components of a cell. Such a system can be expanded indefinitely, and any number from 1 to the highest possible will provide a unique combination of basic categories (not with proportions, simply presence in the cell).

This approach is little-used because it is rather difficult to use, especially if the number of basic categories is large (though the database work already discussed suggests that data entry could be made easy) and also because fine-level distinctions are generally not asserted with GIS analyses. A GIS specialist would not take cell boundaries to represent real-world boundaries of significance. Nevertheless, it is a system to be considered when the complexity of the data seems to demand it.

Another example might take a raster map of soil types and create a new raster map that has the depth of the soil to bedrock instead (assuming 12 cm. for soil type 1, 25 cm. for soil type 2, and 35 cm. for soil type 3 – all found in a *Soils* Table.) The process would simply call for the replacement of each soil type with the depth to bedrock available from the *Soils* Table for that soil type. The resulting map now has the depth of the soil to bedrock instead of the soil type. The new map would be much more useful for certain problems. (See figure 19.)

As we have already seen, a reclassification can be done with user-specified replacements, not just data from a data table. So we could have replaced all occurrences of "1" with text or numbers of our choosing in order to produce a map of more utility for a particular task. Also as already noted reclassification should not alter the original table but create a new one. The original data must not be put at risk in the analysis.

In most raster GIS systems the value stored at each cell is a number. A vegetation map that had pine, oak, and grassland might have a raster map with cells holding 1s, 2s and 3s. Most systems allow an associated label with each value so that maps or reports can show the text label "Pine" instead of the category value. Because a raster map is simply a rectangular set of numbers (a matrix), it is possible to perform many standard mathematical operations on maps. For example, you can add or subtract one raster map to/from another.

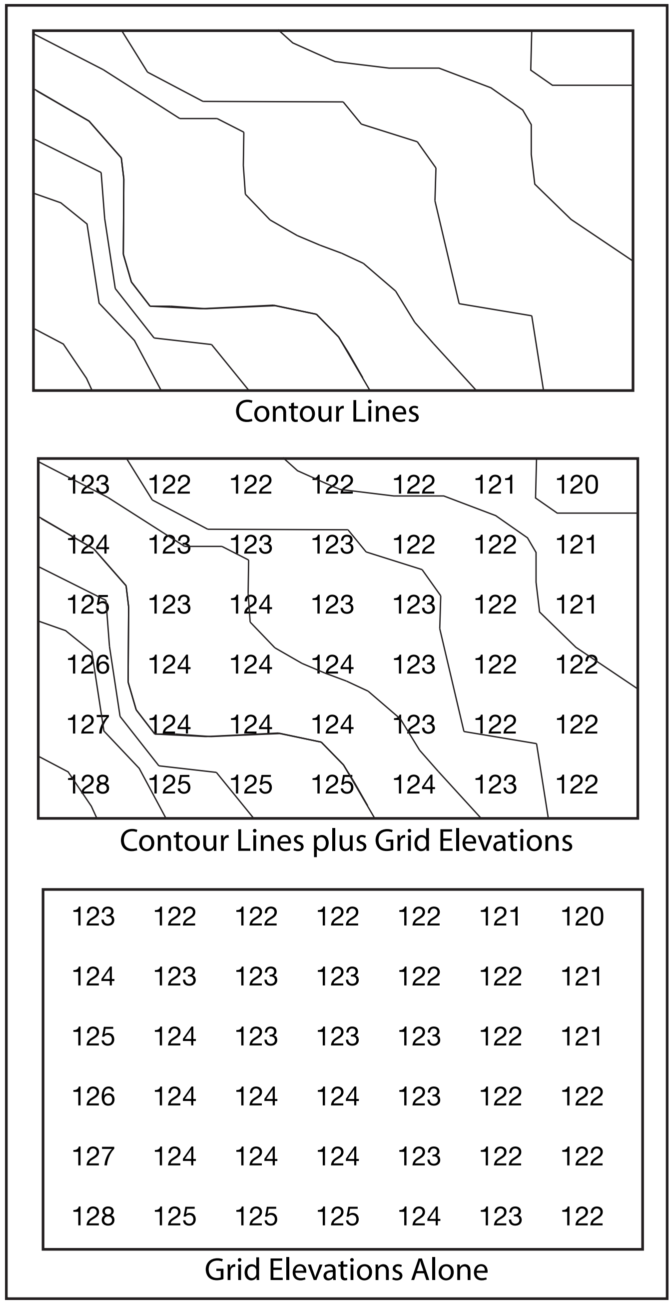


Figure 20
Vector contour lines and raster cell elevations.

While there is a broad range of analytical operations possible in both a vector and a raster GIS, there are some important kinds of analyses that can only (or best) be performed in a raster setting. Analyses that deal with elevation are typically performed in the raster domain. Elevation data can be expressed in a CAD system or a vector GIS as contour lines. Though contours are a common visual form for elevation data they do not provide good data for analysis. A raster map of elevations is much more effective. Common terms for raster versions of elevation data are Digital Elevation Models (DEM) and Digital Terrain Data (DTD). In a DEM each cell has a value that represents the elevation at that location. (Note that the value in each cell in a DEM is a single number which represents the entire area. In a USGS 1:24,000 DEM each cell is 30 m. on a side. In the "real world" there might be a small, narrow (say 2 m. wide) deeply-cut stream running through that location but it would not be represented in the DEM. Different data acquisition and data processing techniques determine what the elevation value will be. In most situations it is a reasonable first approximation to assume that it is the "average" of the area's elevations. (For more on how this process works you might want to refer to the text *Digital elevation model techniques and applications: the DEM users manual,* by Dave Maune, 2001, ASPERS Press.) These maps are far more useful than line drawings from a vector map of contour lines.

Another important operation that can be done with a DEM is to convert the elevation data into derived products. The most common are slope and aspect. Most raster GIS software packages have the capability to do this conversion with a single operation. Aspect is the direction of the slope, south, east, etc, and slope, of course, is the steepness of the terrain. Since both can be calculated by comparing adjacent cells (normally all of the same size) to one another, raster GIS systems perform these conversions quickly and efficiently.

Site visibility maps

Another common raster archaeological operation is the creation of a site visibility map, sometimes called a viewshed analysis. A site visibility map shows all the cells that can be seen from a particular point. In effect the software extends a straight line (as if it were a ray of light) from the origin to all the cells in the map. If the line can strike a cell, that cell is visible. If some cells intervene, then it is not. If a higher location occurs between the origin point and a cell, for example, then that cell cannot be seen. Here again, the use of cells of equal size makes the mathematics of this process relatively straight-forward, reliable, and efficient.

To simulate a real-world view from any map point, it is common to add a meter and a half to the elevation of the origin point's cell, based on a typical person's eye height. Some GIS software packages also allow you to add additional amounts to the surveyed elevation of any cell based on the presence of features that would block visibility such as vegetation or buildings. Suppose you estimated that the study area during the period of interest was forested with dense trees that were 20 meters high. You would first add the 20 meters to the surveyed elevation for those areas that were forested and then do the visibility computation.

The operation might be:

- If cell in vegetation is forested then add 20 to elevation and create new layer called "elevation and forest."

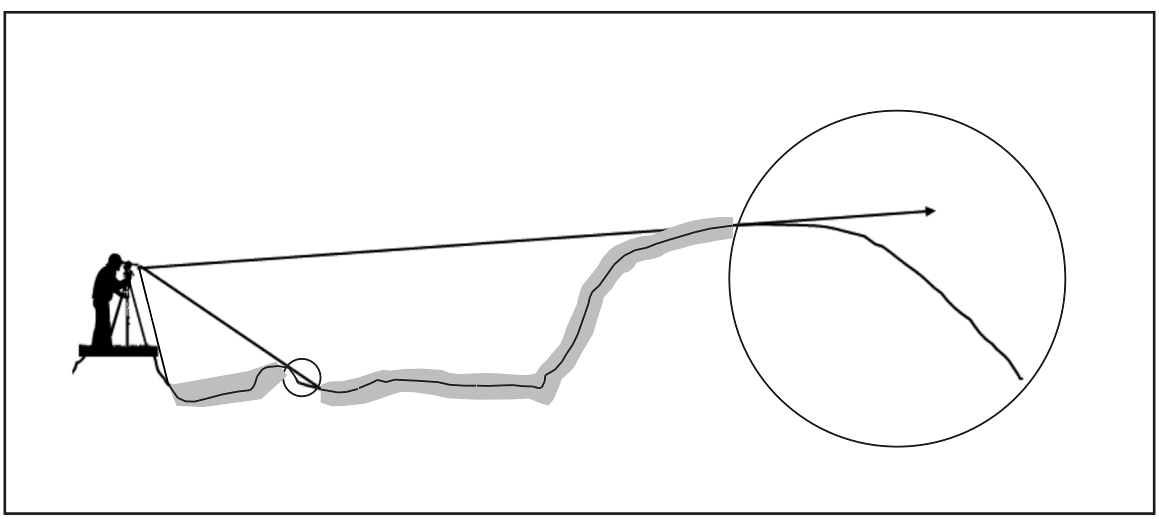


Figure 21

A person standing at a transit looking out on the world can only see (the wide, gray line) that which is not hidden by intervening terrain. Those things hidden from view (in the circles) may be obscured by the terrain or vegetation. Visibility maps or viewshed analyses simulate this process.

In more sophisticated operations you might have information on the distribution of different vegetation in different areas. Perhaps you estimated that plants in some areas were 10 m. tall, in others 20 m., and in still others only 2 m. You would then add different amounts to the base elevation depending on the vegetation thought to have been present. A common way to do that would be to do a reclassification followed by an addition:

- If a cell is hardwood in the "vegetation" layer reclassify that cell to 20 in new layer "veg height."
- If a cell is pine in the "vegetation" layer reclassify that cell to 10 in the new layer "veg height."
- If a cell is grass in the "vegetation" layer reclassify that cell to 2 in the new layer "veg height."
- If a cell has no vegetation in the "vegetation" layer reclassify that cell to 0 in the new layer "veg height."
- Add the layer "elevation" to the layer "veg height" creating the new layer "elevation and veg."

This new layer "elevation and veg" would provide the base for the visibility analysis.

These visibility maps have been used in archaeological studies that deal with the locations of hill forts or for investigations of various ceremonial sites that seem to be located on prominent spots.

Cost surfaces

Another type of raster analysis that has been common in archaeological studies relies on something called a *cost surface*, also termed a *friction surface* or *movement surface*. In these analyses we can assign a cost to the traversing of a single cell in the map. This cost might be time or effort or some other variable. A common approach has been to use the slope of the cell to determine the effort required to cross it on foot. A steep slope would require more effort than a flat one, and there would be some steepness that would make the slope too steep to cross. An alternative to cost has been time where the time to cross the cell has been determined. There is a substantial literature that relates slope to effort with some common examples being the "backpackers equations" (see Wheatley and Giddings 2002:154-156). The process to create a simple cost surface based on slope would be as follows:

- Create a slope map from the elevation map.
- Reclassify slope categories to time or effort values using backpackers equation creating the map "movement cost."

We then would have a raster map that had the cost/time to traverse the cell as the value of the cell. Using slope as the only measure of difficulty to traverse a location is obviously a gross simplification. There may be many environmental factors that effect movement such as ground conditions (e.g. swamps) or vegetation (dense cover). There are many human factors that can also affect travel. The presence of prepared trails or roads obviously reduce travel time and cost, while blocks due to political controls or other factors increase them. We won't go into all of these, but we can show how some of these might be used in our movement studies. Consider swampy soils. If we have paleo-environmental data, we can assign some additional friction or cost to those locations that were swampy. If we know where trails were located, we can reduce the movement costs for these locations. If we believe that a certain area was under the control of a particular group and travel was limited, we can assign a higher movement friction to that area. Starting with our movement cost map we would perform the following steps:

- Select cells in soils map that are swampy soils and create swamp map.
- Reclassify swamp map cells to a value that reflects the estimated impact of

crossing (say 5 units), place a 0 in all non-swamp soils call it "swamp travel."
- Reclassify trail map so that all cells in trails have a -3 (negative 3) value and other cells zero and call it "trail travel."
- Reclassify the political control map so that cells in the dangerous areas for travel have a value of (say) 10 and others 0 and call it "danger."
- Add "swamp travel", "trail travel," and "danger" to "movement cost."

Obviously this process may be quite complex, and the selection of the values to be applied is critical. The absolute values assigned may not be as critical as their relative values. If, for example, we have determined that a flat cell can be traversed at a cost of 10 units, then using an additional friction of 5 for swamp locations says that swampy travel is half-again as costly / time consuming as flat ground. Is this supported by studies or historic literature? Similar careful assessments should be done to determine all the values used.

GIS Public and Commercial Data Sources

One of the major changes that has affected GIS in the last decade is the growth of massive amounts of ready-to-use (or nearly ready-to-use) digital map data, at least for first-world countries. For the US the most popular starting point to find these data is www.cast.uark.edu/local/hunt/. In the UK the primary source of digital GIS data is the Ordinance Survey (www.ordnancesurvey.co.uk). In Canada a good source is www.geobase.ca. In Europe a good starting place is Eurographics at www.eurographics.org and its Geographical Data Description Directory.

A good starting point for digital map data for Japan is www.cast.uark.edu/jpgis/. Data for other areas of the world are quite variable. One useful source for global elevation data is NASA's Shuttle Radar Topographic Mapping Mission at srtm.usgs.gov. Elevation data at a 90 m resolution is available for much of the globe.

Particularly valuable sources of data for archaeological field studies are aerial photographs. Fortunately most first-world countries have accessible aerial photography. In the continental US the US Geological Survey provides the Digital Othro Photo Series. These photographs are at a map scale of 1:12,000. Each pixel represents 1 m. on the ground. For the UK the Ordinance Survey has the Ordinance Survey's Master Map Imagery product has a resolution of up to 25 cm. in selected areas. For counties without accessible aerial photography there is a variety of satellite data sources that may be used. High-resolution satellite data at 0.6 m and 1 m. resolution are available from Digital Globe (www.digitalglobe.com), OrbImage (www.orbimage.com) and Space Imaging (www.spaceimaging.com). These commercial data sources cover much of the world and can be relatively inexpensive when archived (e.g. existing) data are acceptable. An alternative source for many areas around the world is the declassified "spy satellite" data from the US Corona, AEGON, and LANYARD missions (taken between 1959 and 1972) and from the former Soviet Union's SPIN-2 which is much more recent. Resolution for the US data ranges from 6 ft. to 25 ft. while the SPIN-2 data has a resolution of 1.5-2 m. In addition to being inexpensive sources, these also can provide a historical perspective. Information on the US data is available at (edc.usgs.gov/products/satellite/declass1.html), and the SPIN data can be viewed at teraserver.com. The French satellite Corporation has a massive archive of images from around the world, some of which are 5 m in resolution (www.spotimage.com).

Setting up an archaeological survey GIS

We have discussed some of the ways in which GIS can be used to develop an effective approach to assist archaeological site survey. In this section we will walk though the process of developing such a system, and in a following section we will look at a more challenging application, creating a GIS system for an archaeological excavation. Before we start with the site survey system, however, we need to restate the role or purpose of such a system. As we have repeatedly emphasized, it is critical to understand fully the goals of any computer system before implementation. Developing a site survey GIS is not a trivial task, and it is critical to ensure that the results will merit the effort.

Site survey database systems currently in use generally seem to be described as falling into one of two categories: management or research. This distinction may have some merit, but it is an ultimately artificial one. This is especially true for a site survey GIS since there is a large number of research questions that can be considered through use of the GIS that would be difficult, if not impossible, with just a site database.

Accessibility to data and the potential for environmental determinism

Starting out with a GIS for archaeological purposes, scholars in most areas of the world will be struck by the surprising amount of environmental data that is in ready digital form. There are commonly maps for elevations, soils, stream courses, and many other categories. Because of the ready access to these data and their easy use, it is quite possible to slip into a form of technological environmental determinism and begin to see all the problems through these data. Clearly there are, indeed, many interesting questions that can be asked and answered using these data, but the limitations need to be considered. Unless expressly gathered for archaeological purposes the environmental data will reflect current conditions and requirements only. By way of example, there may have been massive colluvial or alluvial processes that have occurred in an area that are not, directly at least, reflected in modern soils data. The assessment of a soil for agricultural suitability is based on modern agricultural practices – not those of the past.

A more challenging issue is to include in a GIS analysis significant social and cultural factors that may not readily be reflected in the basic environmental data. A group may not have desired to farm a particularly productive area because that area was controlled by another group or because there was some cultural prohibition on its use. A pastoral group may view an open plain as a desirable location while an early agriculturalist may have seen the same place as impossible to plow. Careful thinking about these issues, innovative approaches to converting these factors to "map-able" elements and good documentation of the processes will characterize the more interesting and effective studies.

Step 1. Select the software and hardware.

A key decision in any process is selection of the appropriate software and hardware. The first decision point revolves around deciding if your application efforts will largely be vector- or raster-based. While there is probably 50-75% overlap in capabilities between vector and raster systems, vector system are probably preferable for inventory and management purposes while raster systems are better for a number of landscape analyses. This choice need not be an either-or one. There are now several vendors who offer capable vector and raster options. It will probably be necessary to convert data from one format to the other – so if your work will predominately be in one that should be a factor in you selection. Another factor to consider at the outset is the interrelationship between your DBMS, your CAD activities (if any), and your planned GIS. All the software must work together to the extent possible.

Selection of the software should drive the hardware and operating system choices; choosing software on the basis of a preferred hardware/OS platform should never result in a consequential compromise as to the capabilities of the GIS package. The great majority of both raster and vector GIS packages run under Windows. These include ESRI's ArcGIS, AutoDesk® Map®, Intergraph GeoMedia, Clark Labs' IDRISII and many, many others. At this point, unfortunately, there are few capable GIS packages that run on the MAC OS (though Windows – and therefore Windows GIS packages – can be run under the MAC OS). One of the most capable raster based systems for the MAC is MFWorks® by Keigan Systems,® which runs under OS-X, and there is the extensive image processing and GIS package TNTMips® from MicroImages®. The OpenOSX Foundation just (fall, 2006) released a version of GRASS that runs on MACs (openosx.com/grass). Relatively fewer GIS software packages are available under Unix and Linux, but GRASS (open sources) and TNTMips are two very capable systems that will run under UNIX and LINUX.

One intangible that may be the single most important factor in your selection is whether or not there are others (not necessarily archeologists) using the candidate software that you can easily talk and consult with. Even the most technically appropriate system can be challenging to learn to use; being able to work with a community of other users can dramatically improve the ease of use and move problem-solving from a nightmare to a simple telephone call. Fortunately, most GIS users are willing to help others. Be sure, however, to take the training courses that are available from the vendor or others. Many archaeological projects scrimp on training because of budget issues. Training is very much a "pay me now or pay me later" concern, and lack of training will lead to many costly problems later. If you haven't become trained in the basics you will also find that others who would be willing to lend a hand if you have done your homework are not willing repeatedly to help a "newbie" who has not taken the initiative to get the important basic training first.

Step 2. Identify available existing spatial data.

The second step in the process for a regional or survey-oriented GIS is to identify the different kinds of spatial data that may be available. Be sure to do a through search BEFORE you begin. The nature of the available data can strongly influence what you may or may not be able to accomplish down the road.

Be alert to the timeliness of the data. The US Geological Survey provides an enormous range of data but much of it is many years, even decades old. The timeliness of some data that are relatively stable, such as soils or elevation, may not be as important as the timeliness of transportation or land ownership data.

Another key concern will be the scale or resolution of the available data with respect to the objectives of your field investigations. Because development of data is so time-consuming, it is not uncommon that the objectives of a study are adjusted to meet the available data. We can illustrate this issue with an extended example.

In the US there are two different publicly available digital soils mapping data sets from the Natural Resource and Conservation Service. One, STATSGO (www.ncgc.nrcs.usda.gov/branch/ssb/products/statsgo/index.html), is available for most of the US but has been developed at a mapping scale of 1:250,000. Soils are generalized and only substantial variation is mapped. Soils for a major river valley would be mapped differently from upland soils but, within the valley mapping unit,

Vector-raster conversion

Most, though not all, major GIS packages provide the capability to create raster maps from vector ones and vice versa. The process is relatively straight-forward and can best be understood using our translucent graph paper analogy. The translucent graph paper is viewed as a blank raster map and is placed over the vector map. In a simple case, that of find spots, we would place a number in each grid square that covers a find spot. Commonly this raster map would be a binary map with just 1s for cells with finds and 0s for those without.

As we have discussed before, a key initial decision would be the cell resolution. Suppose we have a vector map with a scale that allows us to plot findspots within 20 m. of each other. If our raster map has a cell resolution of 100 m. then we could easily have multiple find spots in one cell. In such a situation many of the GIS software packages will count the number of points in the cell and use that count as the cell value.

If our vector map is one composed of lines (say a stream map) the situation is a bit more complex. In this case we overlay the translucent grid and any cell through which a line passes would be given a value. The value might just be 1 if stream is present and 0 if not, but we might have coded the streams by their attributes such as name (a different numeric code for each name) or stream type. In this case the cell would be given the code value.

When converting maps with lines, there may be a cell with more than one line present. In such a case the software normally provides a rule to manage the situation.

Converting a polygon feature to a raster is similarly straightforward, but there are some complications. Suppose more than one polygon is in a single cell. In this case the software usually has various rules encoded – a common one is that the polygon with the most area of the cell is used to code the entire cell.

Conversion of vector contours to raster presents a special case. Contour maps are neither a line nor a polygon. Each line has a value (the elevation) but the space between is understood to have changing elevation values – in the direction of the adjacent contour. Most GIS packages have the ability to process vector contours into raster elevation data. The basic idea is easy to understand. Suppose there are three cells in the space between the 100 m and 200 m contours. The software would allocate the value 100 to the cell at the contour, 125 to the next, 150 to the next, 175 to the next and 200 to the cell under the 200 contour.

Digitizing Existing Maps to Make Vector Maps

In many situations the existing data have been recorded on paper maps or hardcopy aerial photographs. In order to move these data into a vector-based GIS it is necessary to digitize the map. Digitizing of a paper map is essentially a process of electronic tracing. There are two main ways to perform the process. In the first situation the map is firmly taped down over the translucent, backlit surface of a digitizing tablet. Locations on the map (at least four and usually more) with known geographic coordinates are selected with an electronic pen or sometimes a digitizing puck (a specialized device akin to a mouse but with cross-hairs and special function buttons). The actual coordinates of each of the control points are entered, allowing the software to calculate the proper geographic coordinates for any position on the map. (Actually, of course, the coordinates are calculated for points on the surface of the tablet just below the map; it is the tablet that is directly linked to the computer, not the map itself.) The pen or puck is then used to trace or pick the relevant information. For each point or area digitized the appropriate attributes – such as find spot number or site number – will need to be typed into the system to associate all the other attributes with these geographic entities. Digitizing paper maps needs to be conducted with care as there is the very strong possibility of distortion. If you are planning on doing such work you should first refer to a more detailed review of the process and potential pit falls. For example it is commonly recommended that line information be transferred from a paper map to a stable base such a mylar sheet before digitizing.

Another approach for digitizing information from maps is to scan these into a digital form and then process the scanned image. Scanning can be as simple as using a small desktop scanner or it may involve using large commercial scanners that can handle full-sized maps. Depending on the scanner and the requirements, the map might be converted to a simple black and white image or to a full-color one. The image is then brought into the GIS software, displayed on the computer screen, and traced (by using a standard computer mouse to select points, placing the cursor over the appropriate location on the screen). The scanned image is first registered, the same process used for the digitizing tablet. Four (or more) locations on the scanned image are selected and their coordinates entered into the software. Although it is commonly the case that you will use the mouse to trace or select the relevant data, there are a number of GIS and related software packages that have the ability to follow lines on the scanned image. The software recognizes the dark pixels that make up the line as different from the white ones around it. Commonly you would use the mouse to point to a starting point on a line and the software would then move along the line, converting the information to sets of geographic coordinates.

Needless to say, both sources and digitzing processes must be carefully documented.

terrace soils would rarely be differentiated from flood plain soils. The finer-resolution SSURGO soils (www.ncgc.nrcs.usda.gov/branch/ssb/products/ssurgo/index.html) were developed largely at a map scale of 1:24,000. The SSURGO digital soils would, for example, map differences between floodplain soils and terraces and, in some instances, smaller areas of natural levee well-drained soils within a flood plain. If your project only had STATSGO soils, you would not have the data required to evaluate site distribution and local environmental properties. You could, however, look at larger regional distributions – perhaps how different classes of sites were located in the river valleys versus the uplands. In contrast, with the SSURGO soils you would have data that could support a much more detailed analysis. While we would all prefer that the research questions drive our studies, the practical implications of data access may have a significant impact.

Step 3. Develop data models, data standards, and data development approaches.

While there may be a substantial amount of digital data available, it will probably be necessary to develop project-specific data. The first and most obvious case will be locations of archaeological sites. In some areas local, regional, or national archaeological agencies will already have developed these data, and they can simply be acquired – assuming that you are able to meet the agency's concerns, such as those dealing with site location confidentiality. There may be a conversion requirement. If these data are not available in digital form it will be necessary for you to develop your own. If not in digital form, presumably the locations of the archaeological sites are on paper maps. If the sites are simply dots on a map, then it may be the case that you can determine their geographic locations from the map. If their coordinates have been previously calculated, you will simply need to enter these coordinates into your system. It is typically the case that you can also import a database or a spreadsheet with the coordinates in the file. Most GIS packages have the capability to take these text values and create the appropriate geographic data formats. In many situations the locations of archaeological sites have been mapped at a sufficient scale that they are not dots but actually polygons (areas) on the map. In many situations sites are being plotted on high resolution aerial photographs. In either case you will need to digitize these polygons.

Remember that aerial photography that is not ortho-corrected has substantial distortion, particularly as you move out form the center of the photo; therefore, simply digitizing site data from non-ortho-photography will introduce substantial errors into the system.

In addition, there is a considerable body of literature about the precise definition of a site, with some authors arguing that there is no such beast but simply different densities of evidence of human occupation across the landscape. This is not the place to deal with this issue, but it a matter that the designer of an archaeological GIS must consider.

Given GIS and GPS technology, it would be feasible, though perhaps not practical, to record a location (x, y, and z) for each individual object, feature, etc. rather than draw a bounding polygon that conceptually delimits the "site."

While it may appear that getting the geographic data into your GIS is the goal, the real objective is to get *both* the geographic data and the site attributes into the GIS's database. Again, there may already be a site database with a defined schema, but there may not. In this latter case all the issues raised in the chapter on databases now apply. You need to give considerable thought to the database structure, the individual tables, the attributes (columns), and the relationships between/among tables. If there is an existing database (whether in a database format, a spread-sheet, or on paper forms), then you will

Digitizing data from aerial photographs

A common source of archeological data is information drawn on aerial photographs. This process works almost identically to the process of paper map digitizing – but with one very important difference. Normal photographs taken from an airplane have a number of distortions present. These distortions can be removed through a process called orthorectification. How and why this process works is beyond this text (see Paine and Kiser, *Aerial Photography and Image Interpretation* 2003). The key point is to know if the photograph you are planning on using is or is NOT ortho-rectified. If it is not then the coordinates that will be digitized (or scanned) will be distorted – this is particularly true in oblique photographs such as one taken by simply leaning out of a plane window and snapping a shot of the archeological site. Scollar et al. (*Archaeological Prospection and Remote Sensing* 1990) has a lengthy discussion of the ways in which oblique photographs can be used in archaeological studies. Normal aerial photography is taken with the camera pointed vertically, down to the earth. If you are planning on using data from aerial photography in your work, you should take time to determine the type of the photography and the limitations that it may have, particularly if it is not orthophotography.

One particular note: many archeologists have attempted to digitize site or survey data from oblique aerial photographs and later to correct its location using various warping algorithms that are available in many GIS packages. While there may be some improvement if this is done properly, the conversion of an aerial image to a map-able structure is a complex mathematical transformation called a *projective transformation*. There are very few, if any, GIS packages that support this type of transformation.

need to link the attributes to the geographic data. Just as joining different tables is a critical database operation, you will need to join your attribute data to the geographic data. The key to success here is to make sure that you have a proper identification field and that it has been filled in all the correct places. For example, you will probably use the site number as the vehicle to join the existing site database with the mapped locations in the GIS. This assumes that you have previously entered the site number as you digitized the site locations AND that you have used the same entry there as in the site database. For example, a common site numbering scheme in the US is the "Smithsonian" system where there is a number for the state, followed by two letters for the county and then a sequential number for the sites in that county as they are discovered. A site number like 3WA123 is an example. To join the datasets properly you would need to make sure you followed the same numbering structure in both sets. The site numbers 3wa123 or 3WA0123 or 3Wa123 are not the same, and the primary-key-to-foreign-key link would fail. Defining the specific attributes and data dictionaries for the site tables are critical, as discussed in the earlier section on databases.

When developing a database structure that is a part of a GIS effort, important site attributes that were defined when the system was only a database may become ineffective or unnecessary in a more complex system. Many site databases record attributes for environmental factors such as the distance to the nearest stream, slope, soil type, and elevation. If the site location is being entered into a GIS and if (a big if) you have digital elevation, soils, and hydrography (streams, lakes, and rivers), then you can determine the environmental properties for anything in the area in the GIS itself by overlaying the site layer with the environmental one(s). If sites are entered as points you would necessarily have a single observation for each. If sites are polygons then there might be percentages of the site surface that were different soil types and so on.

At each stage in the development of a GIS understanding the original scale, accuracy, and precision of the different data sources is critical. In an earlier section we compared the STATSGO and SSURGO soils. It would be technically very easy to overlay a site layer on either of these soils and find out what soils underlay the site. However, the STASTGO soils were mapped using data at a 1:250,000 scale. If a site were located near the boundary of a soils unit, it could easily fall on the "wrong" side of the line just because of the scale of the data. If you were working with SSURGO soils (mapping scale of 1:24,000), this problem would be less likely. As you use the GIS to compare sites to various other digital data layers, this issue needs to be clearly front and center, and you need to make sure that the types of analyses or reports you are developing are consistent with the nature of the data that have been used.

It is easy to miss the impact of scale on things like river courses and coast lines. As the scale changes, so does the nature of the river course or the coastline – or highway or railroad line. The river that is a straight line between two points at 1:250,000 may meander wildly at a scale of 1:25,000. As a result, mapping any point location with point coordinates and being sure of its relationship to a river or road requires care and planning.

All these issues deal with getting existing data into your GIS, but there may be quite different strategies for data acquisition in the future that may influence system design. Mapping grade For example, GPS receivers can already be used to record sites locations

GPS systems and GIS mapping

A key technology linkage exists between GPS (global positioning systems) and GIS. The GPS can serve as a basic data source for archaeological applications.

When planning on using a GPS systems there are number of critical aspects to be considered. The most important is to acquire a unit that (1) supports a data dictionary, and (2) permits storage of coordinates associated with the data dictionary entries. Additionally factors such as the number of satellites tracked and other technical factors are important. A series of accessible on-line courses on the use of GPS and GIS are available at outreach.cast.uark.edu:8080/courses/course_list#1.

and downloaded to a computer. In these cases the geographic information can be directly transferred from the GPS to the GIS. For those GPS systems that support data dictionaries or for GPS receivers linked to field computers, it is possible to record much of the site data and transfer both the geographic and attribute data into the GIS data set with little human intervention.

Step 4 Developing data models, data standards and data entry for area surveys.

In the chapter on databases we emphasized the importance of developing effective data models and standards. These are particularly important in the development of GIS data on surveyed areas. In developing the data model and standards for surveyed areas it is important to think through the various ways in which the data may be used in the future. There is a critical linkage between the data model and the way in which the surveyed areas are stored in the computer that may not be initially obvious.

We have previously mentioned that for all our GIS data we have the spatial data and the attributes. There is the implication in this that perhaps we can create the spatial data and then later add attributes. In some cases this is possible, but survey GIS data, for example, is so complex that adding data later can be dangerous. Consider the following situation. Suppose that we have conducted a survey of a particular area. Possible survey attributes might include the name of the surveyor(s), the dates of the work, and other attributes associated with describing the team. These attributes would apply to all the areas that were investigated by the group. Consider further, however, that the same team might have used different field walking methods in different areas and/or that there might have been different surface conditions in different areas. Perhaps part of the area surveyed had been recently plowed and tilled followed by a rain while in other portions of the area the ground surface was covered by heavy grass. Or suppose that, because of project budget constraints, some areas were surveyed intensively by walking at, perhaps, 5 or 10 meter intervals while other areas were covered by survey crew spaced at 100 meter intervals. While these differences must be recorded as attributes of the survey areas, they are attributes about the way the survey process evolved, not the collected attributes that resulted from the survey work. In addition, these attributes of the survey process will apply to areas that are different in size and shape from the distinct survey collection areas. The data will be linked to the spatial data in ways that collection attributes are not and to areas that are different. As a result, we must record BOTH these survey process attributes and the unique areas to which they applied if we believe that we may want to use this information later. If we anticipate that we might want to compare surface exposure or survey intensity to the number of small sites (assuming perhaps that the smaller sites were potentially overlooked in areas with less survey), then we must record the areas to which each different set of methods was applied. It may be necessary to create separate polygons for areas that were covered with different field walking methods as well as separate polygons for different surface conditions. Failing to do so will limit the potential analyses.

It is also very important to consider the types of attributes that you anticipate recording – or that future users of the data might want – in the survey before developing the survey database. If future comparisons of the site distributions and survey conditions are not important, or if all areas were covered in the same way and had the same surface conditions, then you would not need to worry over this. If, however, you suspected that various classes of sites were underrepresented because of field conditions or if site densities differed as a result of different exposure rather than "actual" differences then you would want to include both. Waiting to record this information after the fact is a recipe for disaster.

If we decide that surface exposure and field walking characteristics (e.g., intervals between crew, etc.) are or may be important, then the second major decision to be made is how to map the various combinations of these. Suppose you have an area to be surveyed, part of which was recently plowed and part of which is grass-covered, and that you use the same method to survey both areas. Should you have one layer and record two polygons in the surveyed areas map, with one of the polygons covering the area that was plowed and one covering the area that was grass-covered, or should you instead have two layers? With two layers, one layer would have a large polygon that covered all the area surveyed with the same methods and a second layer, the one that mapped "surface exposure," would have two polygons – one for the plowed field and one for the grass. As you might imagine, it could easily get much more complex. Suppose that you covered part of the plowed field at 10 meter intervals and another part at 30 meters or that part of the grassy area was shovel-tested while another was just walked. While there is no absolute rule, it is probably the best rule of thumb to use a single layer that has an individual polygon for each relevant, different combination of method and exposure. The survey crew should draw polygons around each different exposure area and then subdivide these if different survey techniques are used. Of course the collection form would require that the attributes be recorded for each. The multiple polygon approach is superior to the multiple layer one because overlaying polygons can create problems with slivers; it is usually not possible to insure that two different maps of the same area have exactly the same drawn boundaries.

It is critical to insure that the attributes used to record conditions and methods are defined in advance so that they can be applied in a consistent and reproducible manner and that the field mapping methods used are adequate to record the information. Typically high-resolution aerial photography is used and the survey polygons are drawn on the map. Smaller scale maps (e.g. 1:24000 or 1:50000 etc.) are usually inadequate to record carefully both the area investigated and the surface conditions with enough detail but could be used if they are the only resource.

Using GPS and GIS for Survey Recording

It is possible to integrate GPS and GIS into a very effective combination for survey data recording. Once a useful set of attributes has been developed and tested, then a survey data dictionary can be created and loaded into the GPS. In the field the team can then use the GPS to record accurately and quickly the surveyed areas. This can be done in a number of different ways. The quickest is to locate oneself in the rough center of the area of interest and record a GPS coordinate along with the attributes of the polygon, (e.g. plowed surface, 10-meter interval survey, etc.). This single point can serve as the centroid of the polygon. The actual boundary can be drawn on the aerial photography. Alternatively, with many mapping grade systems you can record "offsets" that delimit an area. An offset is a distance and bearing from a point that the GPS has recorded. A rectangle could be recorded from its center with four offsets, for example. It is not always easy to record these accurately; a more precise but more time consuming method would be to circumnavigate the polygon with the GPS constantly recording points.

If there were an adequate number of GPS receivers, perhaps the ideal method would be to give one to each surveyor. They would be placed in continuous recording mode and the surveyor would quickly record each change in condition as encountered by keying the data dictionary entry at that point. In this way the precise locations examined by each surveyor would be recorded as well as the field attributes. This would also be an excellent approach to recording individual findspots and/or shovel tests. Through development of an effective data dictionary all the information on both conditions and items located could be recorded.

Vector or Raster for Site and Survey Data

With the background we have provided on raster and vector systems as well as the information on site and survey data, it is probably useful to revisit the issue of which approach is "better" – raster or vector – for sites and surveys. Not to beat a dead horse but, as always, the complete answer is that it depends. That said, however, there are some general aspects that we can look at, and these lead us to suggest that probably vector systems are superior for site-inventory-based systems and raster is superior for survey-based ones. If you plan to work with both types of data, ideally you should select a software package that has strong capabilities with both raster and vector data and the ability to transfer data from one form to the other. For many, however, this raises the bar on cost, complexity, and training requirements.

Archaeological Sites, 3 Dimensions, and GIS

A careful examination of the archaeological literature will quickly demonstrate that GIS packages are frequently used in regional and survey level studies, but they have not been commonly used in comprehensive archaeological site investigations. When they are used in excavation settings, it is more likely to be in an inventory and management mode to keep track of objects and site information. There are a couple of reasons for this. One revolves around the technical and training requirements that successful GIS analyses require, and these are discussed an a later section. A more significant issue is the fact that GIS packages, with very few exceptions, manage only the two horizontal dimensions (x and y) and have a hard time with the third – the vertical dimension (z). At first this may seem counter-intuitive because we have already seen the important role of elevation and we have talked about the topological questions of *inside* or *below* – both of which would appear to deal with the third dimension. However, each layer in a GIS can have attributes of a third dimension, but they cannot be used in the analytical processes as we might expect. Let's walk through an example that may help make this clearer. Suppose we have a site-based GIS in which we have the plan-view location of our excavation units, architectural features, post molds, and the like. We can easily locate these in the x and y directions. The beginning and ending depths of these, however, can only be recorded as attributes, and we are unable to record changes of the shape through depth. If we had a pit feature, for example, that was a circle 2 m. in diameter at 30 cm. below the surface and ended in a bowl shape 1 m. below the surface, we could record this in the database but could not easily create a map layer that showed all that three-dimensional complexity. Suppose we wanted to create a map showing all the features that were present at 60 cm. below the surface. We would know from our database attributes that the particular feature was present but we could not know its dimensions at that level.

One approach that has been successfully used is to consider each depth increment to be a different layer. Thus, separate maps are created at the surface, 10 cm., 20 cm., 30 cm. below, etc. This is probably best done in a raster format. If the vertical layer increment is the same as the cell resolution (e.g. 10 cm x and y), then it is relatively easy to conceptualize each as a 3D volume. These three-dimensional cells are termed *voxels*. In the case of our bowl-shaped pit feature, we would have to digitize its outline at each depth to make this approach successful.

While we could create a plan-view GIS we can also prepare an excavation GIS that holds all the site profiles – these would show the x and z values or the y and z – along a single profile but would present a similar problem if we were to create a horizontal representation at some point. The challenges of doing this are substantial and at the limits of current software, but the potential seems great. One application of 3D site analysis is Nigro et al (2002) Swartkrans Cave project. An online version is available at www.cast.uark.edu/local/swartkrans3d/.

There are some new (circa 2006) commercial software packages that are beginning to address the 3rd dimension. A basic voxel (e.g. 3D raster) based software product, not surprisingly called Voxler®, is available from Golden Software (www.goldensoftware.com), and there is a suite of voxel-based products from Ctech (EVS etc at www.ctech.com). The Oracle Spatial version of the Oracle database (www.oracle.com/database/spatial.html) supports 3-dimensional data in the 11g release, and a number of vendors are working on 3D software that will perform 3D operations on this data but there is, as yet, no truly effective 3D system.

Documenting Your Data Sets

Many of the data files used in GIS will be data tables very much like those discussed in the previous chapter. Documentation needs have already been discussed there, but it is important to note one critical difference when documenting data tables used in a GIS setting. Many GIS packages can utilize data tables from multiple software vendors in multiple formats. As a result, it may be necessary to spend extra time making certain that any relevant discussions of file format issues are clear and explicit as to file formats.

It is also important to be clear about the source of all data tables. In a GIS setting, after all, data may come from a wide variety of original sources, from government agencies to commercial vendors. Later users of the data set will need to be able to trace the lineage of all data used; so careful descriptions of the sources of data tables are required.

The maps used in any GIS are, of course, the core of the data set. Like the data tables, they may come from many sources. To make matters worse, they may be vector or raster, and they may be at virtually any scale. Documenting the maps is therefore both time-consuming and critical. At least the following must by specified: map type (raster or vector) and format; scale or resolution; datum, projection and coordinate system, coverage (including all information necessary to locate and orient the map precisely), content (elevations, soil types, etc. – including data dictionary information required for use), source (along with URL and similar information to access the same maps), and date for its creation. When possible, information about the creator and method of creation should also be documented.

Because much of the information required may be available from the vendors who supplied either data tables or maps, there is a legitimate question about the quantity of the documentation required for data acquired from commercial or public sources. This question will be discussed more fully in the chapter on archiving, but, in general, the scholar should err on the side of too much documentation rather than too little. In the end, the documentation will determine how effectively the data set can be used by others. It is the scholar's responsibility to maximize the possibilities.

Technical and training constraints on the use of GIS in archaeology

The primary limitation on the use of GIS in archaeology may be in the relatively limited access to training for archaeologists. Almost all universities and colleges in the US, Australia, Canada, Europe, and many other areas have one or more courses that introduce individuals to the basics of GIS. These courses in the US are frequently located in geography departments while in Canada, Australia, or Europe they may be in geomatics programs, surveying, or similar somewhat more technical settings. In most situations the courses are not taught with a specifically archaeological focus but in many the general method and theory that is offered can be easily extended by the student to another discipline. To be an effective user of GIS, however, it is probably necessary to go beyond the typical introductory GIS course and take more advanced offerings. For many students this may be complicated by interdepartmental issues such as prior requirements, number of seats in the class, and others, but they can usually be overcome by concerted effort. There are a number of on-line opportunities to learn the basics of GIS. These include the ESRI On-line campus (training.esri.com). The UNIGIS on-line program has participating institutions around the world (www.unigis.org/), and Penn State University has an on-line GIS certificate course (www.worldcampus.psu.edu/GISCertificate.shtml). In the last few years textbooks focusing specifically on GIS applications in archaeology have been written. One particularly valuable one is Wheatley and Gilling's *Spatial Technology and Archaeology* (Taylor and Francis

2002). A second accessible source is the Archaeological Data Service's *GIS Guide to Good Practice* (ads.ahds.ac.uk/project/goodguides/gis/).

Selected Further Readings

Two current summaries of work in GIS in archaeology are the chapter by Gillings and Wheatley in the *Handbook of Archaeological Methods* edited by H. Maschner and C. Chippindale. Almitria Press 2005, Lantham, and the 2006 Cambridge Manual in Archaeology volume *Geographic information systems in archaeology* by J. Conolly and M. Lane (Cambridge)

Other valuable sources are the following:

Aldenderfer, M. and H. Maschener (editors) 1996 *Anthropology, space and geographic information systems.* Oxford University Press, New York.

Allen, K., S. Green and E. Zubrow (editors) 1990 *Interpreting Space: GIS and archaeology*. Taylor and Francis. London.

Lock, G. and M. Harris 1995 *Archaeology and geographic information systems: A European perspective*. Taylor and Francis. London.

Wheatley, D. and M. Gillings 2002 *Spatial technology and archaeology: the archeological applications of GIS*. Taylor and Francis, New York.

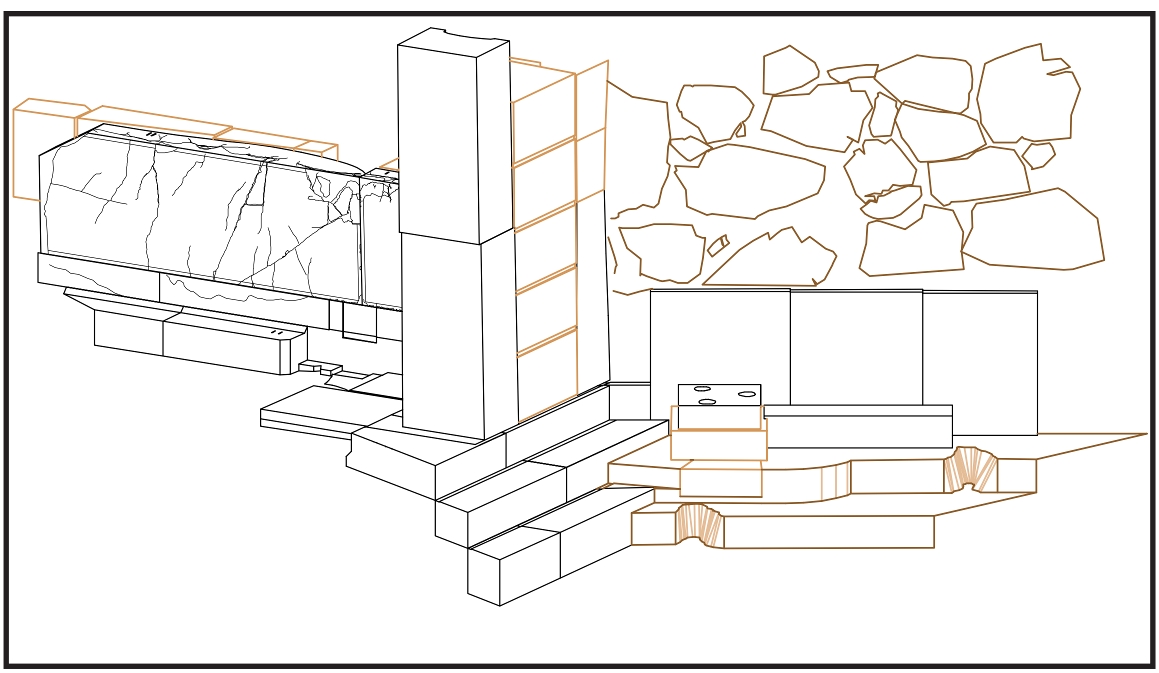
A good source of earlier references on archaeological applications of GIS can be found in Petrie, Johnson, Cullen and Kvamme's *GIS in Archaeology: an annotated bibliography.* It is available on the web at felix.antiquity.arts.usyd.edu.au/acl/products/databases/gis_biblio/preamble.html.

An important early publication on GIS applications in archaeology was Gaffney V. and Z. Stancic, 1996. *GIS Approaches to Regional Analysis: a Case Study of the Island of Hvar. Ljubljana*. It is available on the web at www.arch-ant.bham.ac.uk/research/vince/contents.htm.

Issue 16 of *Internet Archaeology* is largely devoted to GIS in archaeological applications. It is available at intarch.ac.uk/journal/issue16/.

V

Modeling Objects, Loci, Trenches, Features, . . . from Archaeological Projects: Computer-Aided Design Software



Glossary:

AutoCAD: one of the most common and widely-used CAD programs, produced by Autodesk, Inc. AutoCAD was originally available for UNIX computers as well as PCs, and a MAC version was produced for a short time. It is now available for Windows only.

Block: in AutoCAD and some other CAD programs a defined object, consisting of any number of entities, that can be inserted into a model at any point, on any layer, and at any scale or orientation.

Digitize: 1) to convert from analog to digital form, generally in some automated fashion; 2) to convert manually from an analog original into a digital format, as when plans or drawings are copied with a digitizing tablet or scanner.

Digitizing tablet (digitizer): an electronic drawing tablet connected to a computer. The tablet can function as a mouse, controlling cursor movement in a relative sense. With many CAD programs a digitizer can also be scaled so that it functions more like a drafting board. (A digitizer that has been scaled may be used to digitize a paper drawing; such a drawing, placed on the digitizer, may be traced to create a digital version of the information.)

DWG: the file format for CAD files used by AutoCAD. It has become an industry standard and is often specified by government agencies. The format is not without its drawbacks, even when used within the Autodesk program family. Some programs produced by Autodesk in the past, for instance, extended the capabilities of basic AutoCAD and added new ways to model entities, creating entities not supported by the basic version of AutoCAD. Using a DWG format document in another program has some inevitable problems because of differences in the ways various programs operate and the kinds of entities they support.

DXF: drawing exchange format. A file format developed by Autodesk but made public. This is a widely used exchange format, permitting model entities to be moved easily from one model/program to another. Some complex model entities may not be supported.

Entity: a generic term for any portion of a CAD model that is treated as a single item for the sake of copying, moving, or editing. A line, a rectangle, or a circle may be an entity, but so may a group of lines (if created as a group), or a surface (bounded by lines), or a solid.

Hidden-Line Drawing: a drawing (usually a 3D one) that suppresses those lines in the model that should not be visible in the chosen point of view. (See wire-frame.)

Layer (Level): in CAD parlance, a portion of the model separated from others for any reason whatsoever, be it spatial, temporal, chronological, or conceptual. Layers can be included or excluded in any view or paper drawing, individually or in groups. Layers are critical in the scholarly use of CAD.

Microstation: a widely-used CAD program produced by Bentley Systems, Inc. Microstation was available for MACs as well as PCs for many years but is now available for Windows only.

Model: a CAD creation or file. Even the simplest CAD creation is too complex to be called a drawing, a term that suggests a single, discrete view of something; so the term model is preferred. A drawing is a single view of the model, reduced to screen or paper, representing the model from one point of view and at one scale.

Planar: having to do with a single plane in space; flat and lacking modulation.

Rendering: a three-dimensional view that includes artistic effects, possibly including shadows and even reflections, to make the result appear more lifelike.

Scanner: normally used to indicate a device that measures tone or color on a piece of paper or film to create a digital version of the original, much as a photocopier might produce a paper copy. New 3D scanners detect the locations of points in space rather than tone or color on paper. They use reflected light beams to calculate point locations on any object in the area at which the scanner has been aimed. The 3D locations of all points in the field of view are measured, much as they might be with a surveying instrument. The points located, however, are not individually selected; they are the points selected automatically according to a grid resolution.

Scanner resolution varies with the distance from the object but may, with some scanners, be manually adjusted.

Surface Model: a model that explicitly includes surfaces (as opposed to only lines, some of which may bound surfaces). A surface model may include lines or other simpler entities as well as surfaces.

Solid Model: a model that explicitly includes objects defined as solids (as opposed to surfaces or lines only). A solid model may include simpler entities –lines or surfaces – as well as solids.

Spline: a continuous curve drawn through or near specified points. There are various forms of splines, all of which create a continuous curve. Depending on the mathematical representation used, the curve may pass through all points (non-rational B-spline) or only near those points (cubic or quadratic splines).

Total Station: a surveying instrument that combines an electronic version of the traditional theodolite with an electronic distance measuring device (often called an EDM) so that the 3D location of a point in space may be determined. Some total stations require a reflecting prism at the point to be surveyed. Others can survey a point on any surface that reflects enough light.

VR (virtual reality): not really a CAD term, virtual reality refers to systems that use 3D models such as those produced by CAD programs but add the ability to navigate through the model in real time. VR programs often permit such navigation through a rendered version of the model, making the result very life-like.

Wire-Frame: a three-dimensional modeling process or drawing type that deals only with lines in space, not surfaces or solids. A wire-frame view of a model (even a surface model or a solid model) shows all lines and edges, whether or not they should be seen from the chosen point of view.

Introduction

Drawings have been used in archaeology since the beginning of the discipline. They are as important as object records in any excavation or survey since drawings are necessary to provide context information.

We often miss the fact that drawings actually present three distinct kinds of information at once:

1. information about location, layout, relative size, and orientation of trenches, features, structures, and survey grids;

2. information about absolute size (via either specified dimensions or a scale to permit measurement); and

3. information about the character of the surfaces shown (via textures, shading, hatching, and the like).

The drawing shown here in figure 1, taken from *Pseira IV: Minoan Buildings in Areas B, C, D, and F* (eds. P. P. Betancourt and C. Davaras, Philadelphia, 1999), Ill. 40, p. 121, is a typically triple-pronged one. It shows Building BO from the excavations of a Bronze Age structure on the island of Pseira, just off the coast of Crete.

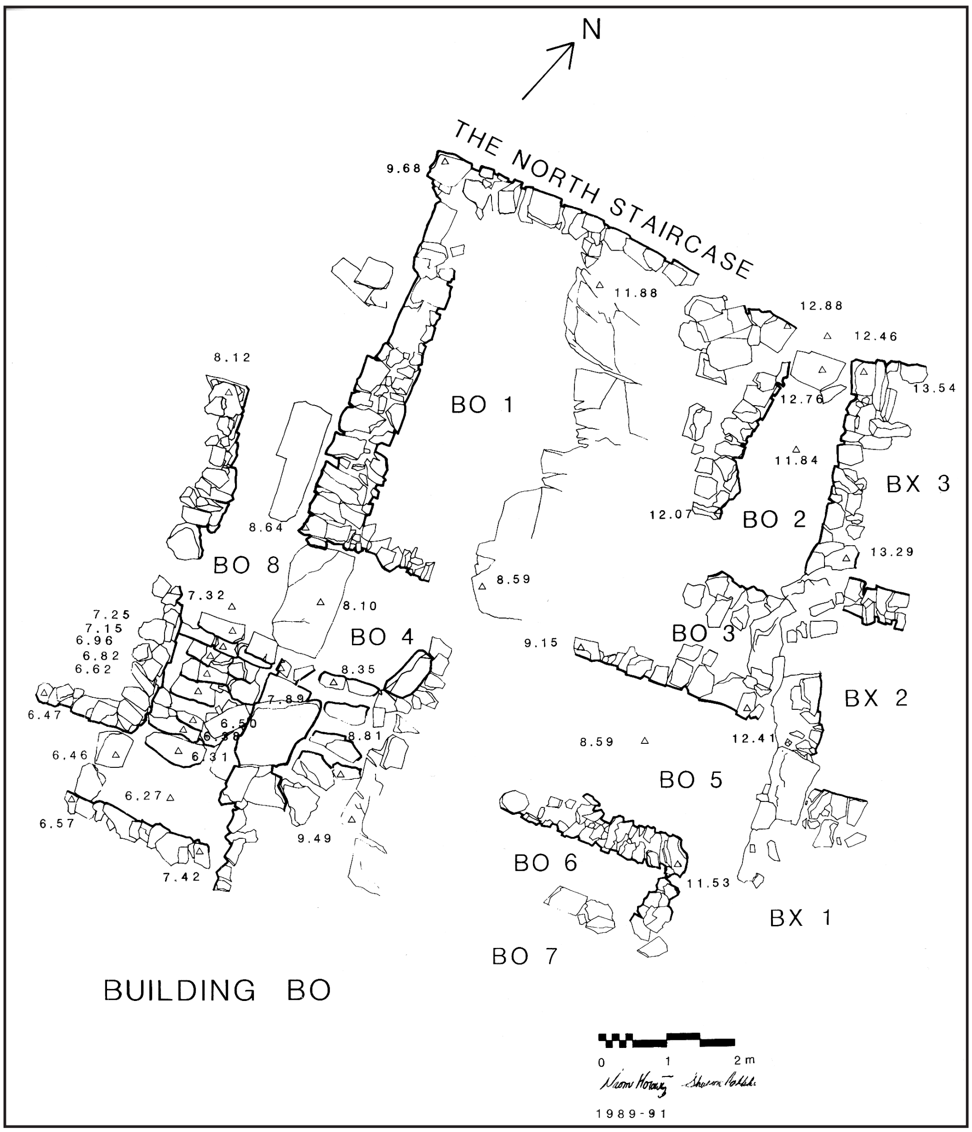


Figure 1

Plan drawing from Pseira IV publication. The published drawing was scanned for reproduction here. The original should be checked for a full appreciation of drawing quality, and this version of the drawing may appear better printed than on-screen.

1. We can see clearly the layout of the excavated structure, the relationships of the parts to one another, and their relative sizes. In many cases the scale – and the information derived from it – would be effectively ignored, as might the various artistic effects.

2. Using the scale, we can also measure any distance to learn, for example, that the room called BO 5 measures about 2.2 m. wide at its widest point.

3. The way the drawing has been made, with heavy lines to mark wall faces and lighter lines for the individual stones making up the walls, illustrates well the bounded spaces in the building and helps to make the whole drawing more intelligible. The use of different line weights also gives the drawing an artistic quality that is typical of the best archaeological illustrations.

The multiple information types – layout, dimensions, and special features – are all needed, and it is important to see that these different kinds of information are all conveyed at once, especially since the distinctions between and among those kinds of information are rarely made explicit and may seem somewhat arbitrary.

The drawing is very informative, though it provides little three-dimensional

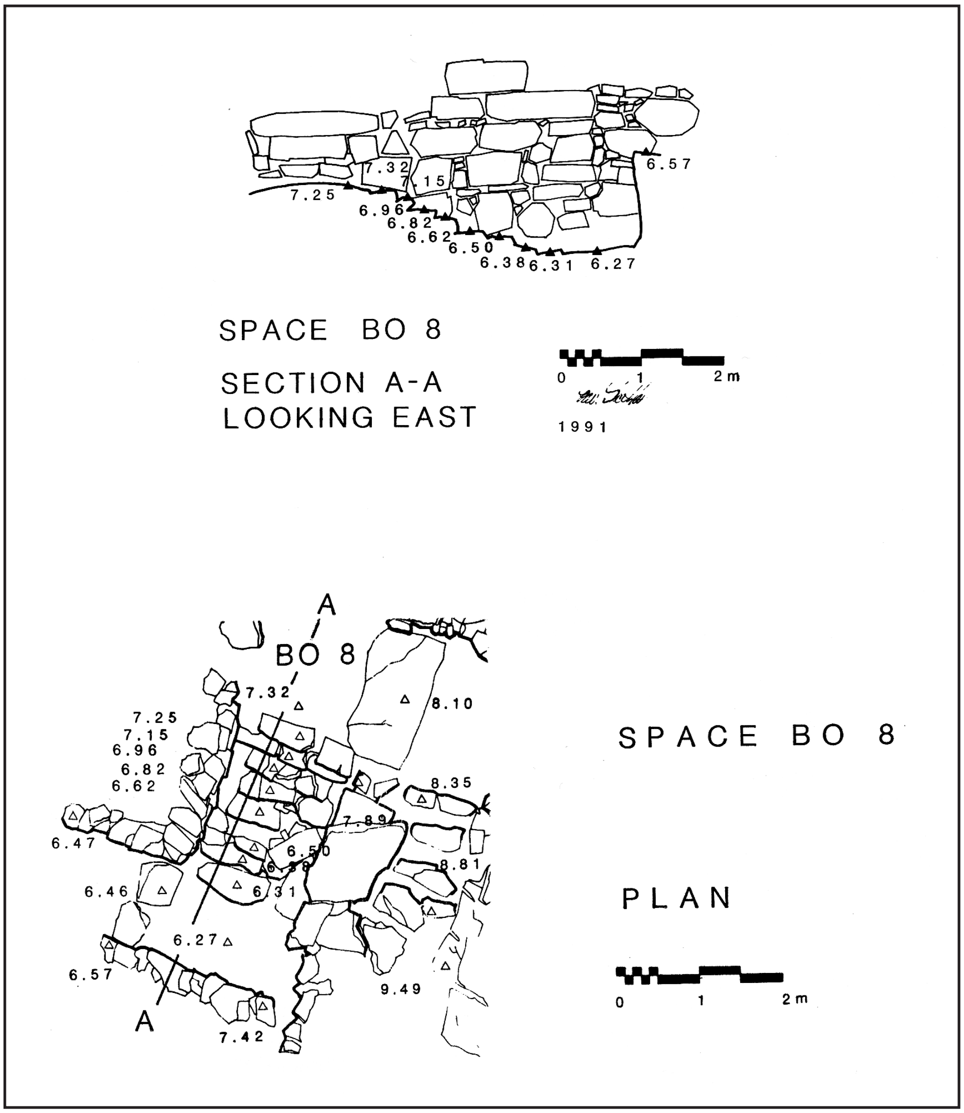


Figure 2

Plan and elevation drawing from Pseira IV publication. The published drawing was scanned for reproduction here. The original should be checked for a full appreciation of drawing quality, and this version of the drawing may appear better printed than on-screen.

information. Indeed, most archaeological drawings are made in a manner that provides little or no 3D information. Surveying of wall tops, which yielded the drawing here, without an elevation drawing of each wall or block within the wall does not provide enough information for a 3D view. Even a combination of wall tops with elevations – provided for Space BO 8 in the following illustration in the Pseira volume (Ill. 41, p. 122 – our figure 2) does not provide enough information for a fully three-dimensional understanding of the wall, since neither the top nor the face of the wall is a simple planar surface.

Drawings such as these drawings from the Pseira publication are produced from survey data, but they are also true drawings in the sense that they are drawn by hand with some artistic flair and in ways that make them more than a geometrically accurate representation of the found reality. Implicitly, some aspects of the drawing reflect specific aims. For instance, including here only the material from a few rooms is one born of necessity – only so much will fit on a single page at a scale large enough to show some detail. In addition, the lines defining individual stones in the walls are not straight, point-to-point lines; they reflect the shapes of the stones but are not intended to be mathematically accurate. They rely on a relatively small number of surveyed points; the original survey may have included some measured points plus lines drawn on site with the aid of an overlay string grid and/or staked strings along the wall. That field drawing would then have been translated into a record drawing, with the surveyed points retained as reference points and the sense of the individual stones copied from the field drawing, perhaps with the aid of trips to the site and/or photographs. The record drawing might then be copied in ink and at a specified scale to produce the publication drawing shown here. In this multi-step process the surveyed points are critical for maintaining scale throughout the drawing and for retaining the relationships to other parts of the excavation. In making the final drawing, of course, the scale factor is critical because it determines how much detail can be included. (Those familiar with the process of moving from the original drawing to a printed version thereof will understand that there are other issues that must be carefully planned to create a good published drawing.)

These drawings convey some three-dimensional information. Elevations of some points are shown, as explicit elevations (m. above sea level or a datum point) adjacent to icons indicating where the elevations were taken. The stated levels show elevations for the tops of wall stubs or for the floors of rooms where the levels were taken. There is additional 3D information in the elevation drawing. Section views are a similar way to add 3D information. The minimal 3D information, of course, complicates the retrieval of dimensions; any measurement taken from a drawing provides only a plan-view distance between points; the actual 3D distance cannot be measured from a plan. That is, if the two points measured are not at the same elevation, the true distance from one to another will be greater than that indicated by measuring and applying the scale factor.

The absence of 3D views that combine plans and elevations to provide a more realistic view of any excavation or structure is hardly surprising. Not only are such views very difficult to make, they require a great deal more survey information, since more points must be surveyed and every point must have three coordinates, not simply two, as required for a plan or elevation. Thus, both more survey data and more drafting time are required for 3D views. Since such drawings would not necessarily provide more information than is required for understanding the structure, adding the third dimension in drawings may not be worth the time and trouble. Photographs are often be used instead.

Similar drawings may be produced of buildings constructed of cut stone or other more geometrically regular materials, structures that are easier to survey because they exhibit less irregularity. For instance, portions of the Propylaea, the entrance building to the Athenian Acropolis constructed during the classical period, have been surveyed and drawn, and one of those

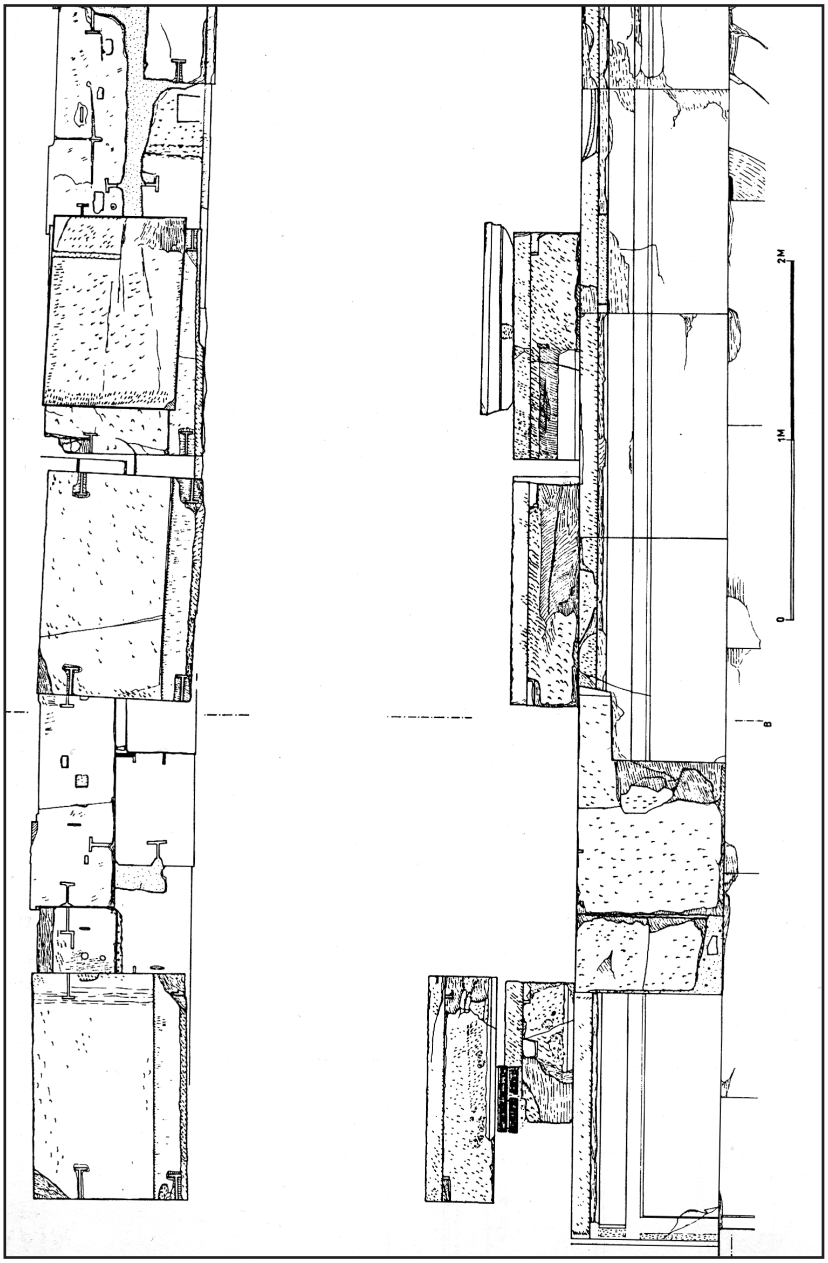


Figure 3

The left side of a drawing from *Study for the Restoration of the Propylaea, Volume I*, T. Tanoulas, M. Ioannidou, and A. Moraitou, Athens, 1994, Fig. 68. The drawing was made by T. Tanoulas. The published drawing was scanned for reproduction here. The publication should be checked for a full appreciation of drawing quality, and this version of the drawing may appear better printed than on-screen.

drawings is shown in figure 3. In this case, there are other artistic effects in the forms of shading and hatching to indicate tool marks and degradation of the building stone. The "reader" must know that the stones have been carefully shaped as ashlar blocks to understand the meaning of the hatching and shading, which represent tool marks and wear in this case; knowing that, however, the drawing is very clear. In addition, the use of the hatching and shading provides here an artistic effect that, as with the Pseira drawing, makes the drawing attractive, not simply informative.

These drawings represent the state of the art toward the end of the twentieth century. Conventions such as the heavy lines on the Pseira drawing and the shading/hatching on the Propylaea drawing have been developed to permit draftsmen to draw what has been surveyed and, at the same time, express a great deal more than simple plan or elevation information.

Drawings like these are informative, attractive, and clear. They are not perfect, though. Since they are drawn to scale, the dimensional or geometric information they record cannot be more precise than the chosen scale permits the draftsperson to display – and the "reader" to retrieve. A drawing made at a 1:100 scale must represent a distance of .1 m. (10 cm.) in the real world as .001 m. (1 mm.) on paper. As a result, a superior 1:100 drawing, measured by a competent reader, can probably yield no dimension more precise than to the nearest 10 cm.[1] That is far less precise than the field measurements from which the drawing descends; those measurements were most likely made to the nearest mm. As the scale of the drawing gets smaller, the precision of measurements taken from the drawing declines further. A natural corollary is that, as the precision with which field measurements are taken rises, the capacity of the drawing to display them cannot keep pace.

In addition, scaled drawings produced for archaeological projects generally lack the third dimension because three-dimensional drawings are simply not expected and because the production of 3D views is much more difficult than the production of plans and elevations. If true three-point perspective drawings are needed, the demands on the draftsman are even greater.

Drawings also lack the fourth dimension, of course: time. A series of drawings of building phases or site development can illustrate change over time, but a single drawing cannot. Occasionally multiple phases can be illustrated in a single drawing through the use of colored, faded, or broken lines for secondary phases, but only rather simple differences can be illustrated in that way.

These problems aside, the quality of archaeological drawings had reached a very high level some decades ago. Few scholars felt the need for better drawings, though published drawings may often have been criticized as either too few or too small for their explanatory roles. The issue there was publication cost, not potential quality.

The discussion so far has concentrated on published drawings, at least in part because only published drawings are available for general examination. However, it is very important to be explicit about the obvious: most of the drawings for any project will remain in the project archives and be seen only by the draftsperson(s) and other project personnel. Those drawings may have all the survey information such as coordinates and dimensions. They should have everything recorded in the field, including notes, comments, and elevations. Some will be at very large scale to include extensive detail; others will be at small scale to include the entire project area. Those drawings are the original record of the work. The published drawings, on the other hand, are distillations from the record drawings; all are made explicitly for publication and are intended to emphasize particular features, structures, finds locations, and so on. As with card files discussed in the database chapter, the record drawings can only be consulted by going to the project archives in person; even then, they may require interpretation and comment from project personnel. Nevertheless, those record drawings will contain a great deal of information that cannot be included in the published drawings. Anyone seeking a full understanding of a project would want to consult them as well as the original file cards.

Producing Drawings for Publication

There are some important aspects of making archaeological drawings for publication that need to be discussed before continuing. Any drawing, no matter how basic, has an intended finished size and an intended use. Sometimes the size is implicit – the size being created – but often the size intended is not the actual size produced; the ultimate product may be an enlargement or reduction created in the printing process for publication. (As a result a draftsman may use not only a magnifying glass but a minifying glass as well to examine drawings as they will appear after enlargement or reduction for publication.) Similarly, the use may

[1] A careful reader could certainly measure to precision finer than the mm. with a micrometer. However, the precision produced by a draftsman working to create lines of measurable width that are then photographically reproduced, combined with the typical reader's measuring potential, is not such that confidence in finer precision can be high.

be implicit, as it generally is, but often multiple drawings of the same material are produced at different scales that suit their different purposes; the drawings at large scale cover a small area but show more detail, and they offer more potential to take measurements than those at smaller scale. The drawings at smaller scale may include a wider area to show more and broader relationships between and among the items/structures shown, but they offer lower measurement precision.

The size of a finished drawing is a remarkably important matter that affects many aspects of the drawing; some of the issues are obvious, at least after they have been pointed out. Others are more unexpected. For instance, the weight of the lines used must be related to the size of the drawing. A very large-scale drawing can accommodate very heavy lines, but a dense, small-scale drawing cannot. On the other hand, very thin lines may disappear altogether if a drawing is reduced in the printing process. Indeed, reduction is especially common, with drawings prepared at a scale convenient for the draftsperson but then reduced, either to a common scale or to a size that can be accommodated by the page size of the publication. In such cases, a beautiful drawing may turn out well or not, depending on the combination of the amount of the reduction and the draftsperson's forethought.

Good Drawings Gone Astray – The Problem of Advancing Technology

One example of the pitfalls that can await good drawings may be of interest. When I prepared drawings for a publication of the results of work I had done in Athens, I made numerous sample drawings, each with slightly different effects to determine how to produce what I hoped would be superior drawings. I was especially conscious of the need to produce good drawings, because I knew that, as someone associated with the use of CAD in archaeology, my drawings would – and should – be carefully scrutinized.

I decided to show both actual and reconstructed material together in most of the drawings and experimented to see how best to distinguish between the actual and the restored (without color, of course, since that is too expensive). Drawings made locally with the best printing service I could find yielded a good solution - thin, black lines for the actual finds and thicker, gray lines for the restored material. Experiments to find the right choice of gray for the thicker lines finally yielded thick and thin lines that seemed equally prominent to the eye. Therefore, each drawing could be "read" as if all lines carried the same importance. Yet the distinctions between actual and restored blocks were very clear.

The drawings were sent off with the text, and, when the bluelines (blue approximations of the finished drawings and photos) were sent back, I objected, noting that the difference between gray and black did not seem apparent. I was assured that the difference would be clear in the final version. To my horror, that was completely wrong. All the lines, thick or thin, were black. The appearance was terrible. I was embarrassed and still find it necessary to apologize for the quality of the drawings.

What had happened? The experimental drawings printed locally had been made on a plastic-coated stock used by the printing shop to produce precise, controlled lines and points. The paper used by the publisher, on the other hand, was standard (clay) coated paper used for high-quality printing. That coated paper absorbs ink much more than the plastic. Each of the small dots on the plastic stock became a larger dot on the coated paper, and those larger dots merged, making the light gray lines turn virtually black.

This was not a mistake in the sense of somebody doing something incorrectly, against normal operating instructions. Rather, it was an accident caused by the kind of misunderstanding that is so easily encountered when people who don't really understand one another's needs must communicate – and, in this case, through a third party who understood neither of the interested parties. The communication problems were exacerbated by the fact that the process involved trying to use old and new technologies together without adequate experience. The lesson, though, should be clear. Important drawings should be carefully crafted, and the people responsible for the printing should understand the intentions of the draftsman – and be responsible for meeting them.

Heavy lines that seem clear and that can be reduced well enough may become so tightly spaced as to be unreadable. Thin lines may drop out; differences in line weight that are obvious at one scale may be negligible at a smaller scale.

Text – notes, labels, or added dimensions – presents similar problems. When the text is large enough to be seen on the original, will it still be the right size on a reduced-size copy? For the draftsperson who is interested in the best quality – as is obviously the case for virtually all – there are also issues of consistency. If some drawings are to be reduced, then all should be reduced by the same percentage. That may not seem important, but line weights will be reduced along with the overall sizes of the drawings, and a good set of drawings should have consistent line weights. The same, of course, can be said for text. Text items should be the same size from one drawing to another.

Filled areas present another problem because of printing processes. Since gray is typically created in printed drawings by using black dots – close together for a darker tone and further apart for a lighter one – a drawing that is to be reduced or enlarged must be very carefully prepared if there is any part of it filled with gray. Various combinations of inks and papers can have unexpected results, with dots sometimes running together to make gray turn into solid black. Similarly, small dots can turn into larger blobs.

These issues may all seem academic, in the pejorative sense, but they are important issues in the production of drawings for publication, and they all relate to the values of CAD as a technology.

Defining CAD

Having avoided a definition of CAD software to this point, I can no longer put off trying to define it, though that can be very difficult because the paper analogs are generally misleading. Therefore, I will try to define CAD without reference to paper analogs at the beginning.

CAD programs provide a way to create a computer model of some physical reality – a model not a drawing. The model exists in the computer in much the way a database does, as stored components with various attributes. The model may be too complex to be seen as a whole and must then be viewed as individual views of parts of the whole, either on screen or as drawings on paper. Neither screen views nor paper drawings can show the full model in most cases; any view is merely one view of a particular portion of the whole, made from a particular point in space, at a given scale, and with specific colors, line types, and line weights. The model itself is a geometrically accurate and complete record of its subject, optionally in three dimensions, and including precise, coordinates in real-world terms, not at scale, for all points in the model. All points in the model are recorded in a Cartesian grid system with x, y, and z coordinates.

Those coordinates can be retrieved by a CAD user at any time. As a result, whatever precision a surveyor or designer has used is retained in the model and available to users, and the computer can directly calculate the distance from any point to any other from the stored coordinates with that level of precision.

As the model is created, the use of a 3D coordinate system as the underlying base means that the complete geometry of the item being modeled is always retained. Furthermore, CAD programs have the capacity to provide a view of the model from any point in space, using the geometric information and the rules of geometry and perspective to generate any desired view on command. The view can be a plan, elevation, axonometric, or perspective view, at the pleasure of the user. Today it is even possible to produce a physical scale model from a computer model. Although the scale factor would limit the precision of any measurements, it would be an accurate reproduction of the geometry.

The child's block shown here is a simple illustration of the power of CAD. The drawing in figure 4 is an engineering drawing that might have been produced

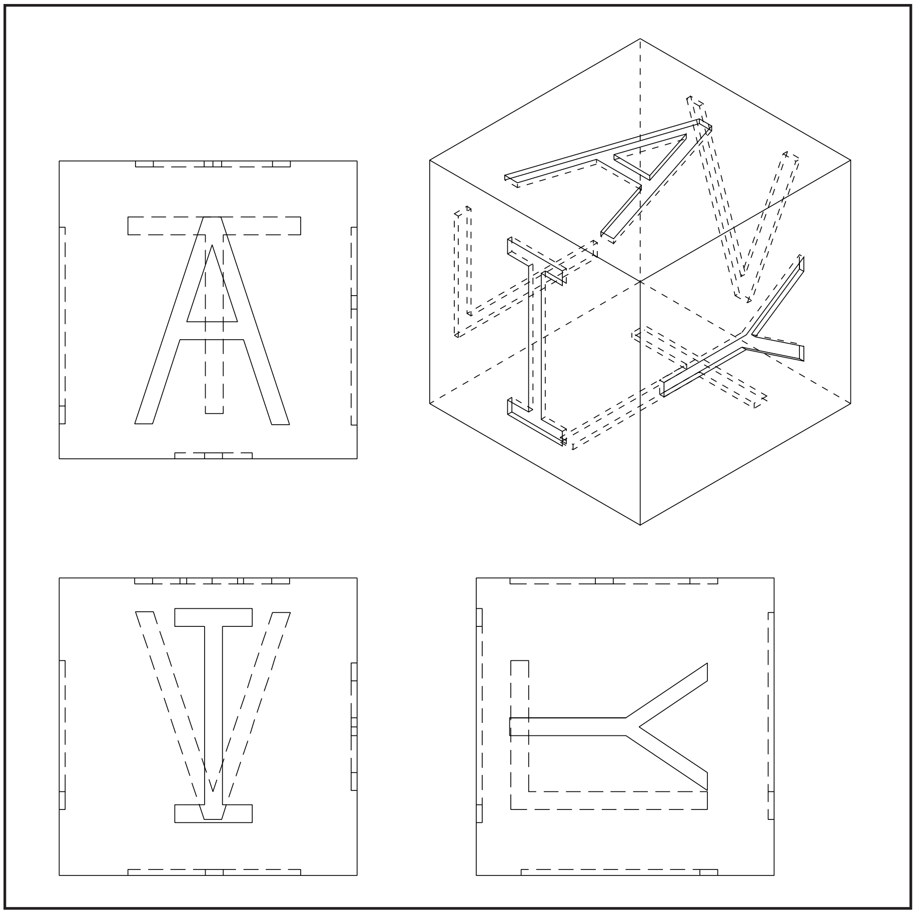


Figure 4

A child's block as drawn in a standard engineering exercise, with a front, top, and right-side view (each using broken lines for hidden parts), plus an isometric view.

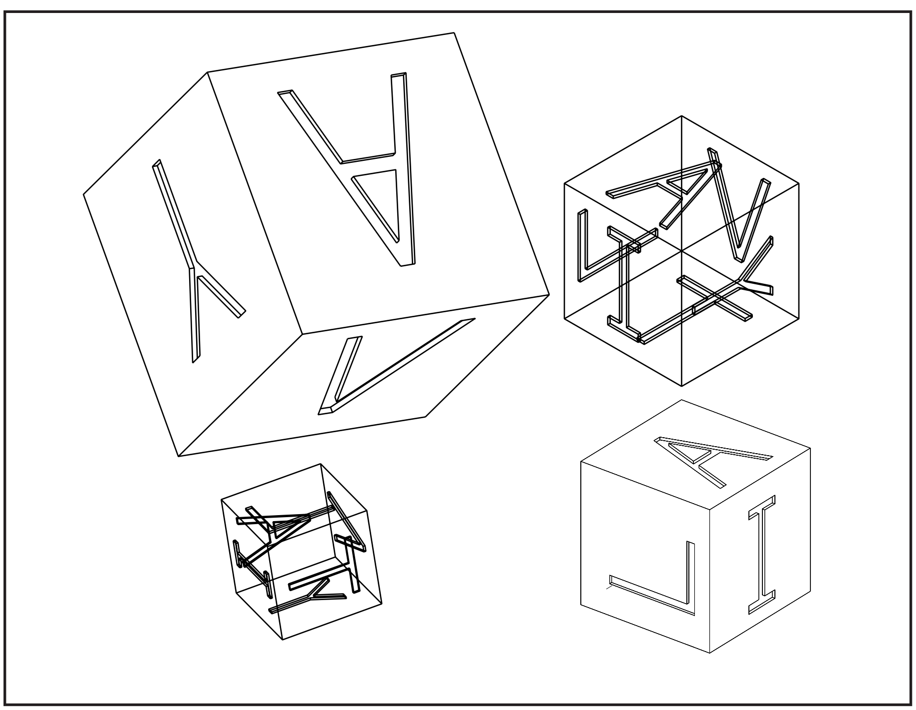


Figure 5

The child's block as a CAD model - seen from various angles, at differing scales, and with or without portions of the hidden sides showing.

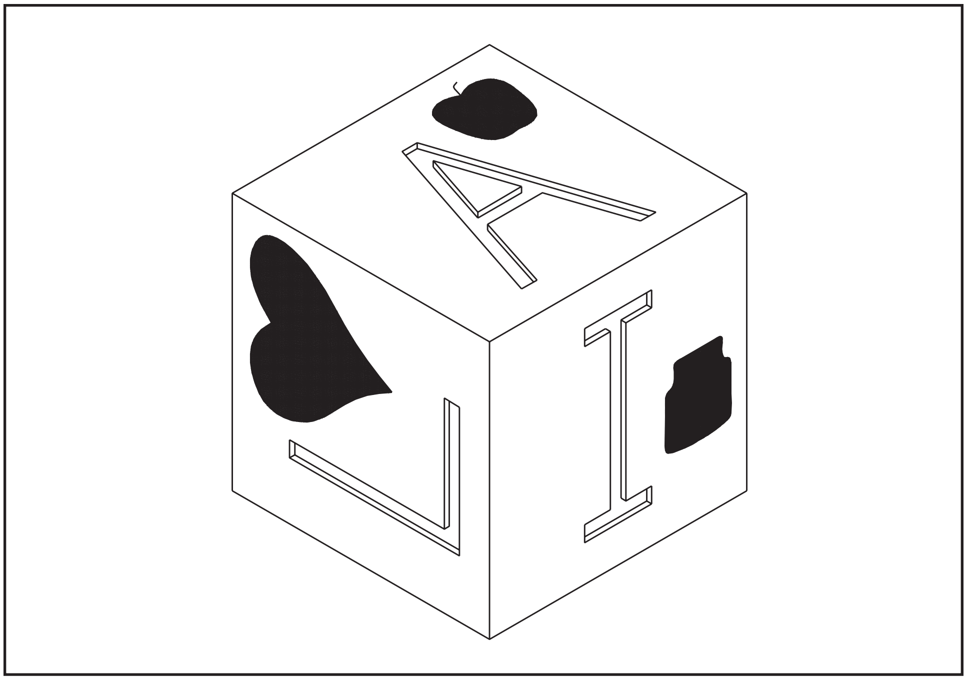


Figure 6
The child's block illustrated in figure 5 but with added decoration.

many years ago. The drawings in figure 5, on the other hand, are all views of a computer model of the block, generated by the computer according to the principles of geometry (and perspective when requested). Each is based upon the accurate and complete geometry of the basic model, and each is completely dependent on the coordinates specifying the locations of the points in the model. Two show the block with all portions indicated (and no broken lines) but at different scales; the others show the block with appropriate portions hidden from view but from different viewpoints and at different scales. Any individual drawing could be enlarged, reduced, or changed in any one of a myriad of ways, from changing line weights or colors to adding textures, but each view is based upon the geometry and, when applicable, the rules of perspective.

CAD models are also constructed of individual segments, and each may be included or excluded in the model maker's view of the material at any time. That is, CAD programs permit users to see all of a model or only specified segments of it. That feature may be illustrated with another view of the block, figure 6, this time with decoration on the block – an apple, a heart, and an ink bottle – shown. Since those decorative items were made in a separate segment of the model, they could be excluded for the views in figure 5, included in figure 6. The viewpoint chosen does not affect the ability to include or exclude any portion of the model.

The use of these model segments permits CAD to be used for the fourth dimension, time. One may move through time phase-by-phase with a CAD model, simply by including for each phase all the segments that are appropriate for that phase and no others. So long as the segments have been properly created, it should be possible to show each phase separately. Of course, different scholars with different views of the phasing may each create a unique sequence.

CAD's Development

CAD is really two different streams of software that merged into a single, more complex program type as computers became powerful enough to support all the functions of both the original streams. On the one hand, architects needed programs to automate drafting processes when dealing with large and complex buildings. Small changes to a plan might require an entire new set of drawings, but each drawing could entail many hours of labor and risk the introduction of new errors, making yet more drawings necessary. Using a computer to automate that process was so obviously desirable that large architectural firms began working to create their own computer-assisted drafting software in the 1970s.

The early architectural CAD software reflected the needs of a specific part of the architectural community, those firms dealing with very large and complex projects. For such projects it was necessary to design far more than the walls, doors, and windows that seem to make up the essentials of a structure. Air conditioning and heating systems had also to be designed and all the duct work planned. Plumbing pipes had to be routed through the structure. Electrical wiring for all the individual spaces in the structure (each with its own interior distribution of outlets, overhead lights, and so on – and possibly its own meter) had to be planned. The elevator shafts and associated mechanical parts also had to be planned. Each of these parts of a structure had to fit with all the others, but putting all of them together on a single drawing was impossible; all the necessary detail could not be contained in one drawing. To accommodate all the detail on paper drawings, architects had been using a system of transparent overlays. While the walls, doors, and windows of a structure were shown on a base drawing, details were on individual overlays – one for the electrical system, one for the heating and air conditioning, and so on – that could be positioned atop the base drawing, correctly oriented, and easily understood as a part of the whole. All the drawings were made on sheets with holes for registration pins so that correct orientation was easy; in the U.S. this system was called pin-bar drafting or pin-bar overlay drafting.

A computer analog to the overlay sheets of pin-bar drafting was critical; so the computer-assisted drafting programs incorporated a simple system for separating the various parts of a drawing while maintaining the correct spatial relationships. Each drawing entity – any line, arc, circle, or point representing any part of the whole – could be drawn in place and then assigned to a specific drawing segment, each of which could be included or excluded for any particular screen image or paper drawing. Drawings could be divided into as many segments as the draftsman wished, each segment being the analog of a transparent overlay. Each drawing entity – regardless of the drawing segment – must be in its proper place in the Cartesian grid so that the relationships between and among entities are correct, but any segment or segments may be included or excluded for any drawing or screen view. Thus, computer-assisted drafting programs permitted everything in a model to be assigned to a specific drawing segment and thereby to be included or excluded in any on-screen view or paper drawing along with all the items in that segment. It is important to note that the segments in such a system are conceptually distinct and may not be physically distinct. That is, a heat register and an electrical outlet may appear to be in the same place on a plan drawing (one being at a different height than the other); so at typical drawing scale it may not be possible to separate them from one another. This is an important point, because the term used for a drawing segment is usually either layer or level, and both imply a physical boundary for the material. But there need be no physical boundary, only a conceptual one: as one layer shows where the heat outlets are located, another shows where the electrical outlets belong, and the locations may be indistinguishable at drawing scale.

The final output of computer-aided drafting software was intended to be physical drawings, drawings equivalent to those that might have been produced by hand without CAD software. Good drawings were a crucial product of these CAD programs. Artistic quality was not important to the users of CAD software. Straight lines and straight-forward, unadorned drawings were the desired results.

The other stream leading to modern CAD programs began with engineers who had two very different needs from those of the architect. They needed three-dimensional information, and they needed real-world coordinates of points that could be entered and retained without scaling so that the original precision would not be lost.

Engineers designing products to be molded from plastics must design both the products and the molds into which molten plastic will be pumped to create the finished product. For small products, the mold is made, in part, by machining two mating metal mold parts that, when fitted together, leave a cavity for molten plastic, a cavity that is exactly the shape of the finished part. The machining of the molds is a slow, time-consuming process that requires great care and very tight tolerances. Applying computer technology to the machining process promised savings in time and personnel costs as well as better, more precise results. However, making a three-dimensional part to very tight tolerances required two important kinds of information. First, it required that the basic computer version of the mold parts be fully three-dimensional. That is, the computer had to treat the mold parts as 3D objects, not a series of 2D drawings thereof, the standard way for engineers to show their ideas. Second, the cutting machines had to be guided along very precise routes, with tolerances that might be incredibly tight. The only way to accomplish the two goals was to store the computer version of the mold in real-world three-dimensional numbers, not scaled dimensions. Thus, all points would be defined in a 3D Cartesian grid, and all distances in the computer version of a mold would be determined by their positions in the grid, not scaled approximations of those positions. Screen views and paper drawings might be scaled, but the computer version of the item must not be.

These 3D computer designs with real-world dimensions are no longer drawings. Drawings can be produced by the computer on screen or on paper, but the underlying computer design is more than a drawing. It is a model of an object, fully three-dimensional and conceived in real-world measurements, not scaled ones. That is important, because it affects both the terminology and the approach to making the model. As to terminology, it is better to use the term *model* (as both verb and noun) rather than *draw* or *drawing*. That is, one models an object; one does not draw it. This is not a trivial distinction. *Model* reflects the process better; the process of making a model is a significantly different process from making a drawing. The result is a model, not a drawing. Again, the distinction is valid and important, since a drawing is a necessarily 2D abstraction of a 3D object. From the model any number of individual drawings can be made, but each is an abstraction from the model, which is more complete and more complex than any individual drawing.

Once a model, not drawings, became the result of the design process, it was a short step to adding 3D viewing capabilities, as shown in figure 7. Viewing the models as if they were real 3D objects

An Unexpected Benefit of CAD's Real-World Scale

A very famous inscription from fourth-century Greece describes in some detail a shed for storing ships' rigging – the so-called Arsenal of Philon, Philon being the architect. The parts of the building are carefully defined in terms of the Attic foot, an ancient measure that scholars believe to be equivalent to .295 m. Were one to attempt to create a CAD model of this arsenal, as some have tried in fact, imagine trying to model something one-and-three-quarters of an Attic foot in length (1.75 x .295 m. = .51625 m.).

A CAD model, however, can be constructed with the Attic foot as the base unit. Making such a model requires only the simplest of measurements, using the Attic foot as the basic unit. One need only convert palms (four palms per foot), and dactyls (four dactyls per palm) into metric equivalents. The same process could be used to model any structure in any unit of measurement. A scale in a modern measurement unit can be added, of course, to permit comparing the structure to a known quantity. Alternatively, the entire structure could be adjusted in size at the end of the process so that the unit of measure corresponded to any desired unit, making all model measurements retrievable in that chosen unit of measure. In any case, modeling in the original unit of measure is both easier and more true to the source information.

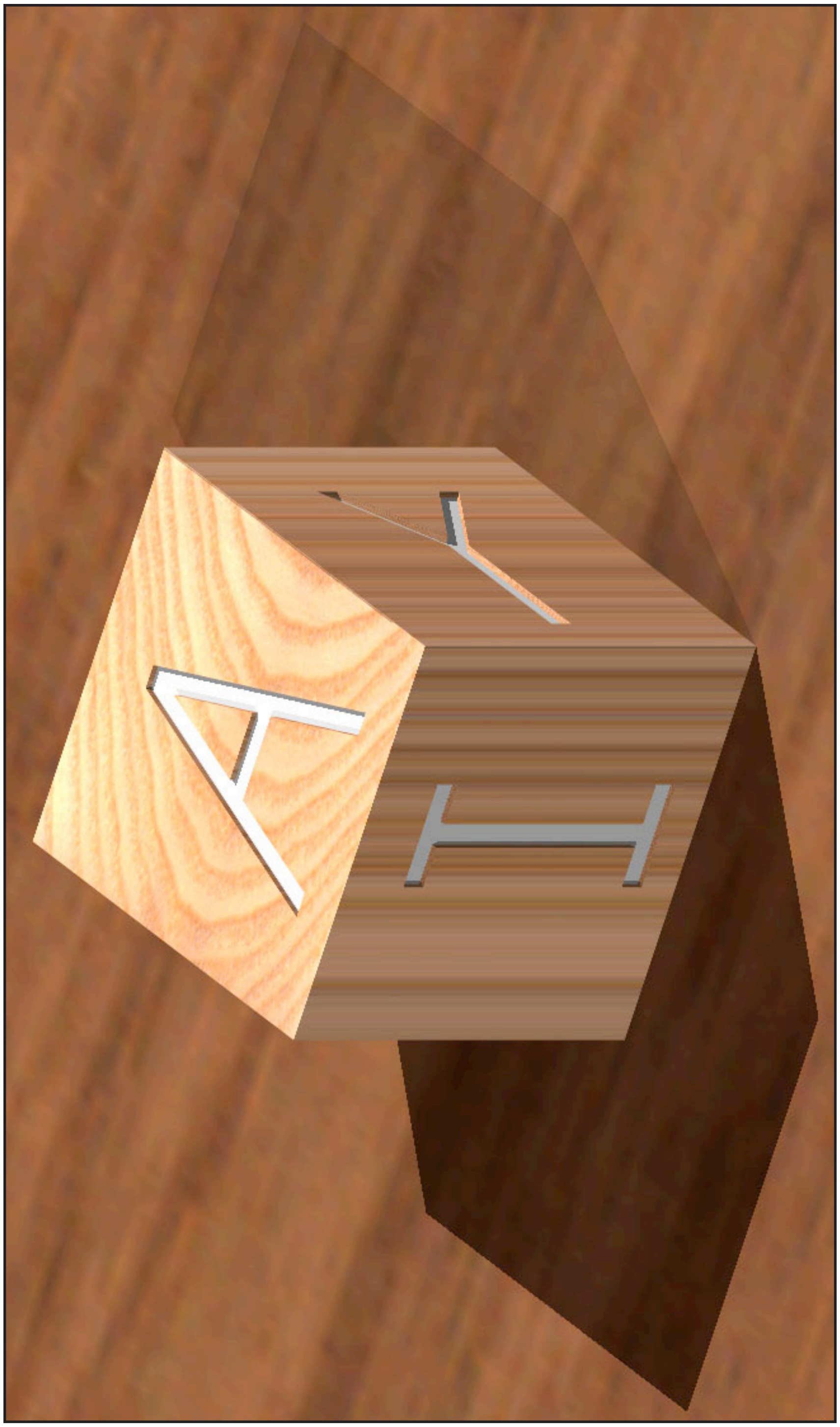


Figure 7
A child's block as a CAD model - with added surface textures, an explicit surface beneath, and two light sources to illuminate the block and the surface on which it rests.

encouraged industrial designers to go one step further and to demand software that could show models with color, texture, and lighting effects. Because they were dealing with mass-market objects that required very large investments of capital to manufacture, they could afford the costs of the computing power more easily than they could afford the costs of unappealing designs.

The separate streams of CAD software – computer-assisted drafting for architects and computer-assisted design for engineers – merged when computers became powerful enough to permit both sets of functions to be used together. Nevertheless, CAD programs are usually aimed at one or another of those markets, architecture or engineering, because the practitioners work in different ways and desire different kinds of automation. Some programs have remained more neutral, relying upon add-on software to provide features and processes for either architects or engineers.

The combined drafting and design programs were for a time called computer-assisted drafting and design (CADD) programs to make clear a distinction from the two original streams and the combined software, but the more accurate name never gained much currency and has not remained in use. CAD applies to all, though some will take that to be an abbreviation of computer-assisted (or computer-aided) design and others of computer-assisted (or computer-aided) drafting. Modern CAD programs include segmenting models into layers and very sophisticated systems for producing paper drawings at various scales, with standard measurement styles, with borders and labels that are set independently, and with multiple views on a single sheet. At the same time, CAD programs now use real-world dimensions as the core of the model, not scaled ones, and they treat all points on the model and the model itself in three dimensions. (Some inexpensive programs allow only two-dimensional drafting.)

What Does CAD Provide to Improve Archaeological Drawings?

The state of the art with archaeological drawings was excellent many years ago, as we have seen, and the problems associated with publishing drawings are reasonably well understood; so why does one need a computer? One could mimic with the computer the drawings used as examples above, but there is little point in doing something with a computer that can be done by hand. The aim should be to add capabilities, not simply replace paper with digits.

When the excavator of Pseira, Phillip Betancourt, prepared to publish the Pseira cemeteries, he chose to use CAD instead of traditional hand drawings to make the preparatory and final drawings. The benefits for the study and publication illustrate the advantages of CAD. A drawing of Tomb 2 is shown here as figure 8, and the entire cemetery is shown in figure 9.

One Data Source

Although two drawings of the Pseira cemetery have been shown, there is, in fact, a single CAD model of the entire Pseira cemetery; each of the drawings is an excerpted piece of the whole, a specific portion at an appropriate scale. Each tomb was modeled separately because each had been drawn separately in the field, but each was modeled at its proper surveyed location in a single large model, as if the paper size were infinite. The large model included, in addition to the individual tombs, the survey grid, the contour lines, and the coast line.

Had the Pseira cemetery been drawn in a paper-based drafting environment, we would have needed to choose between a large scale for modeling the details of individual tombs and a smaller scale that would have made the whole drawing fit on a reasonable-sized piece of drafting paper. In fact, we would have chosen to make a series of drawings rather than a single drawing to deal with the scale/size dilemma. The cemetery as a whole would have been drawn with minimal detail for the individual tombs, just enough to illustrate the relationships between and among them, and

Measurement and Line Weight

Using different line weights to assist with drawing clarity is a hallmark of good draftsmanship. Emphasizing some lines and reducing the importance of others is critical to clarity. However, using a drawing with different line weights has an impact on measurement precision. When anyone needs to obtain a measurement from a paper drawing - whether the drawing has been done with a CAD program or by hand – where does one measure when a line is thick? To one side of the line or the other or to the middle, which will almost certainly be impossible to locate precisely? (In a CAD model, the center of the line marks its true location.)

In a CAD environment the paper drawing is not the best source of measurements. The CAD model – queried with the CAD program – provides dimensional information more accurately and more precisely. As a result, concerns about measurement from a paper drawing and the impact on drafting practices need not trouble the CAD model-maker.

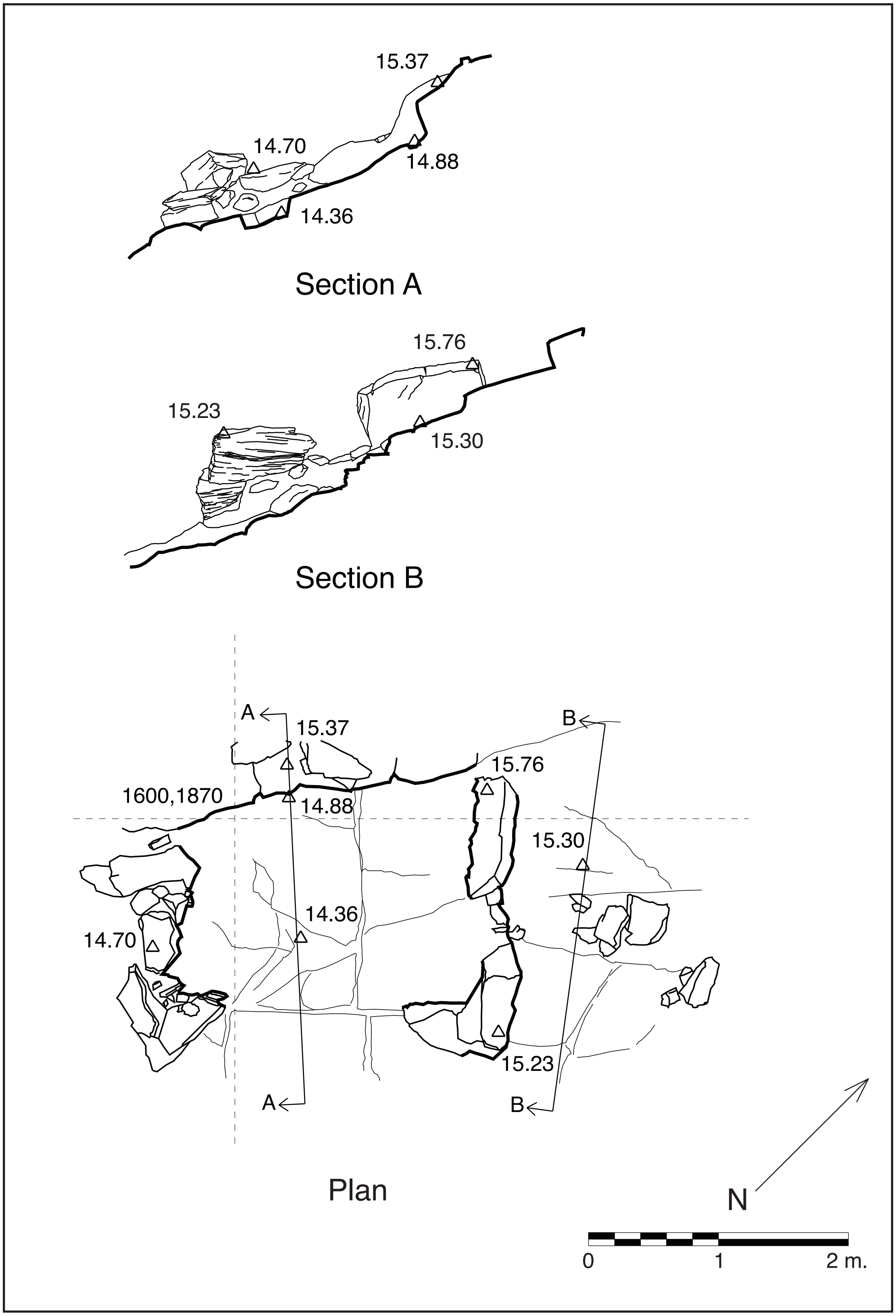


Figure 8

Plan and sections of Tomb 2 from the Pseira cemetery. In this case the drawing is from the CAD model, not scanned from the publication. The drawing here, however, is not directly from AutoCAD; an AutoCAD drawing was modified in Adobe Illustrator. Line weights were adjusted there for printing, and text was added. The broken line showing the excavation grid was also trimmed in Illustrator.

Figure 9

The entire cemetery from Pseira - from the CAD model via Illustrator.

each tomb would have been separately drawn, with all recorded details. The two drawings shown here, figures 8 and 9, in fact, illustrate those two extremes, the one being a detailed drawing of an individual tomb and the other a drawing of the entire cemetery, with less detail about each tomb. As drawings from a CAD model, however, these are just individual views of a single all-inclusive model. Everything in both drawings – and all the details for all the tombs – is included in a single CAD model. Even cross-sections are in the model.

The first advantage of CAD in the Pseira example is thus the inclusion of all the information about the cemetery in a single model. In that model we have details of individual tombs, sections, the excavation grid, contour lines, text, elevation markers, and the coast line. Indeed, the CAD model can encompass a virtually unlimited area and virtually unlimited levels of detail – and the size of the finished drawings need not be considered at all when the CAD model is being constructed. Each finished drawing is produced for its specific purpose and may be tailored

AutoCAD

Different operating procedures are used in various CAD programs, and there are large and small differences between/among CAD programs. In general, however, the differences between and among CAD programs have more to do with operational differences than conceptual ones.

Throughout the remainder of this discussion of CAD, the processes used in AutoCAD will be referred to when a process must be described or illustrated; AutoCAD terminology will also be used. Describing multiple approaches or multiple terms becomes impossibly complex, and AutoCAD is such a widely-used standard, with specific advantages for use in archaeology, that using AutoCAD approaches and terminology appears to be sensible.

AutoCAD seems to be the best CAD program for archaeological use for a variety of reasons, but two are critical and can rather easily be described. (A key disadvantage to AutoCAD is its availability for Windows only. There is no longer a version of AutoCAD for the MAC OS or for UNIX; there has never been a version for Linux. AutoCAD can now be run on newer Intel-based MACs with Windows installed.)

First AutoCAD is one of the few programs that permit a mildly irregular surface – a subtly undulating wall, for instance, or a block that seems to be flat but is slightly warped – to be modeled with all the survey points required and still to appear as a single, undivided surface in a simple line-drawing. Most other programs can deal with such surfaces only as a group of separate, adjoining surfaces that seem to be continuous in a shaded or rendered view but not in a line drawing. Lines separating each part of the larger surface would appear in a line drawing. (A work-around could be used in those programs, to be sure, but it would add considerably to the difficulty of using them.)

Second, AutoCAD permits segments of a model to be named, which is not unique, but it also permits searching all the segments in a model for those whose names contain common characters or combinations of characters. The important matter here is the search possibilities; good search procedures make it possible to use segment names to create database-like access to the segments in an AutoCAD model. This is a feature of enormous importance to the effective use of CAD in archaeology. Absent this database-like access to CAD model segments, it is much more difficult to use a model as an analytic tool. Models may still provide a great deal of information, but that information is much harder to retrieve and to use for analysis.

The capacity to access model segments in groups also aids enormously in the production of printed drawings.

Although there may be other programs that offer the same capabilities, none is as widely sold. (AutoCAD's high cost can be partially offset by the generous discounts available to academics.) None can be learned at so many places either, from community colleges to technical schools to large universities. Finally, there is no other CAD program with such a wealth of third-party books and learning aids.

Figure 10

The AutoCAD model window divided into four windows, each with a different zoom ratio to show more or less of the Pseira cemetery.

to suit that purpose. (As the area covered by a CAD model grows, of course, the problems introduced by the earth's real shape, as opposed to the Cartesian grid on which CAD models depend, limit the coverage of any given model.)

It may seem that the value of the single model – as opposed to many individual drawings – is vitiated by the need to publish individual drawings anyway. However, the excavators can use the model (as could any student with access to the model and the CAD program) to examine the cemetery in ways impossible with a set of drawings, no matter how many individual drawings were made. At any moment a user can enlarge a portion of the model ("zoom in") to see more detail or minimize ("zoom out") to see a wider area – or pan from side-to-side or up-and-down to see parts of the model not in the current view. Indeed, virtually any level of detail can be produced instantly. It is even possible to have multiple views on screen at the same time. Some of the possibilities are shown in figure 10, showing a multi-window view of the Pseira model, each window having a different selection of material at a different scale. (You may notice that all lines have the same weight. On-screen line weights are not generally helpful, especially as one zooms in and out with CAD. Colors are more useful to distinguish different materials.) In effect, the CAD model provides access to a drawing of any portion of the model, at any scale, with any colors for emphasis on command, and anyone using the model has that level of access.

The production of the Pseira model shows how the use of a single model can provide all the data for a site – and all that is needed to generate any number of individual drawings. It is remarkably complete in itself; so the model can be used by scholars to obtain views and combinations of views that would be truly impossible with paper drawings. Indeed, the CAD model should be seen as the real data source; paper drawings generated from the model exist only for publication or other illustrative purposes. Needless to say, the CAD model can become the primary data source only for those who can use the CAD program available for the project in question. Those unfamiliar with or lacking access to the requisite CAD program will be dependent on drawings produced from the model by others.

Measurement Retrieval in a CAD Model

The screen-shots shown above do not have scales in them because the use of CAD makes a scale superfluous. The dimensional information available to a CAD user is far better than anything that a scale – even a pervasive scaling indicator

False Precision in CAD Systems

CAD systems record all points in a coordinate system as precisely as the computer system permits. (To be more precise about the use of double-precision numbers here would be off-point, but a search of the Web for information about double-precision numbers will provide information for those interested in this issue.) Avoiding issues of number storage systems, it is safe to say that, in general, that means a point will be recorded as if precision were far higher than we can measure. As a result, a user who queries a CAD model may receive information that implies false precision, as if we had measured to the nearest hundredth of a millimeter or to even finer tolerances. In AutoCAD, for example, a query may identify a point as having an x-coordinate of 1.10200000 and a y-coordinate 2.00500000, although the actual locations entered by the site architect were 1.102 and 2.005 (survey precision to the mm.). It is one of the failings of CAD for archaeological use that the precision will always be reported with the standard number of apparently significant digits, regardless of the precision of the original data. Programs may permit a user to display precision only to the nearest mm. or tenth of a mm. (as in figure 11). However, any choice for display of precision applies to all coordinates and/or dimensions; it is not possible to attach a precision limit to any point. Nor is there any simple way to indicate that any given point should be treated as less precisely defined than any other point.

This problem with precision means that every model must be carefully documented as to the precision of all measurements used. That documentation cannot be implicitly included by limiting individual survey point precision at time of display or query; so it must be done in external documentation. The documentation must include survey precision for all data in the model so that users know what level of precision to assume in any subsequent analysis. That documentation requirement does not mean that each point has its own attached precision; it does mean that full and complete explanations of precision used throughout the model must be included so that no user is mislead.

such as a grid – can provide. That is so because the core of any CAD model is retained in the computer as a set of points with real-world coordinates. Every screen view as well as every paper drawing is simply a scaled product made from the real-world-scale data. Therefore, the distance from any point in a CAD model to any other point in that model can be calculated instantly by the program from the underlying coordinates. Furthermore, that distance can be calculated with precision regardless of the current display scale, since the calculations depend on the coordinates, not the scale of a particular presentation.

The upper portion of figure 11 shows a CAD screen in the process of a "distance" query (in AutoCAD the command *distance* requests the distance between the two points selected by the user), with the line stretched from the starting point and the system waiting for the selection of the next point near the cross-hair marking the current position of the cursor; the text window at the bottom shows the current state of the command. The lower portion of the figure shows the result of the query in the text window, with the full distance between points as well as the distance on each axis and two angular dimensions. In a CAD environment, scale is not an issue, and the scale of an individual drawing does not restrict the precision with which dimensional information can be retrieved. If original measurements were made to the nearest nanometer, a CAD system could retain that precision. (Display with four decimal points, shown in figure 11, is the AutoCAD default, but the system can be instructed to display precision to the user's specified level of precision. See above, False Precision in CAD Systems.)

Segmenting a CAD Model

The Pseira model could be shown in so many different ways because of the ability to segment the model discussed previously. Any entity in a CAD model – a line, a box, a circle, a triangle used to indicate an elevation measurement point – must be placed in a specific model segment, and there may be hundreds of segments, each of which can be individually included or suppressed in the current display or for a paper drawing. As a result, the CAD modeler can segment a model in any way that seems useful for working with the model, for displaying it, and for analyzing it.

The Pseira model had segments for the grid, the contour lines, the scale, the north arrow, and for each tomb. There were actually several segments for each tomb:

1. general outline of the tomb
2. outlines of individual blocks making the tomb
3. cracks and markings on the individual blocks
4. cracks in the floor of the tomb
5. triangular elevation marker(s) indicating where an elevation had been taken
6. text for elevation marker(s)
7. lines indicating the presence and orientation of section drawings
8. text for the tomb label

There are many reason for breaking the model into all those segments. The first reason has to do with the way the model is used by the excavator or other members of the staff; they need to be able quickly and easily to eliminate the things that are chaff in a particular view – text, section lines, and elevation markers, for instance, so that they can get an uncluttered view of that which is important to them at any given moment.

Any user of a model, in fact, should be able to choose the visible model segments according to any relevant analytic criteria and see or suppress other segments according to changing analytic criteria. The possibilities are rather limited in the case of the Pseira tombs, but they are nevertheless important. A user should be able to select any tomb or group of tombs and, for all tombs selected, see the general outline and/or the interior details. We will return to this issue because it is so critical to the use of CAD in archaeology.

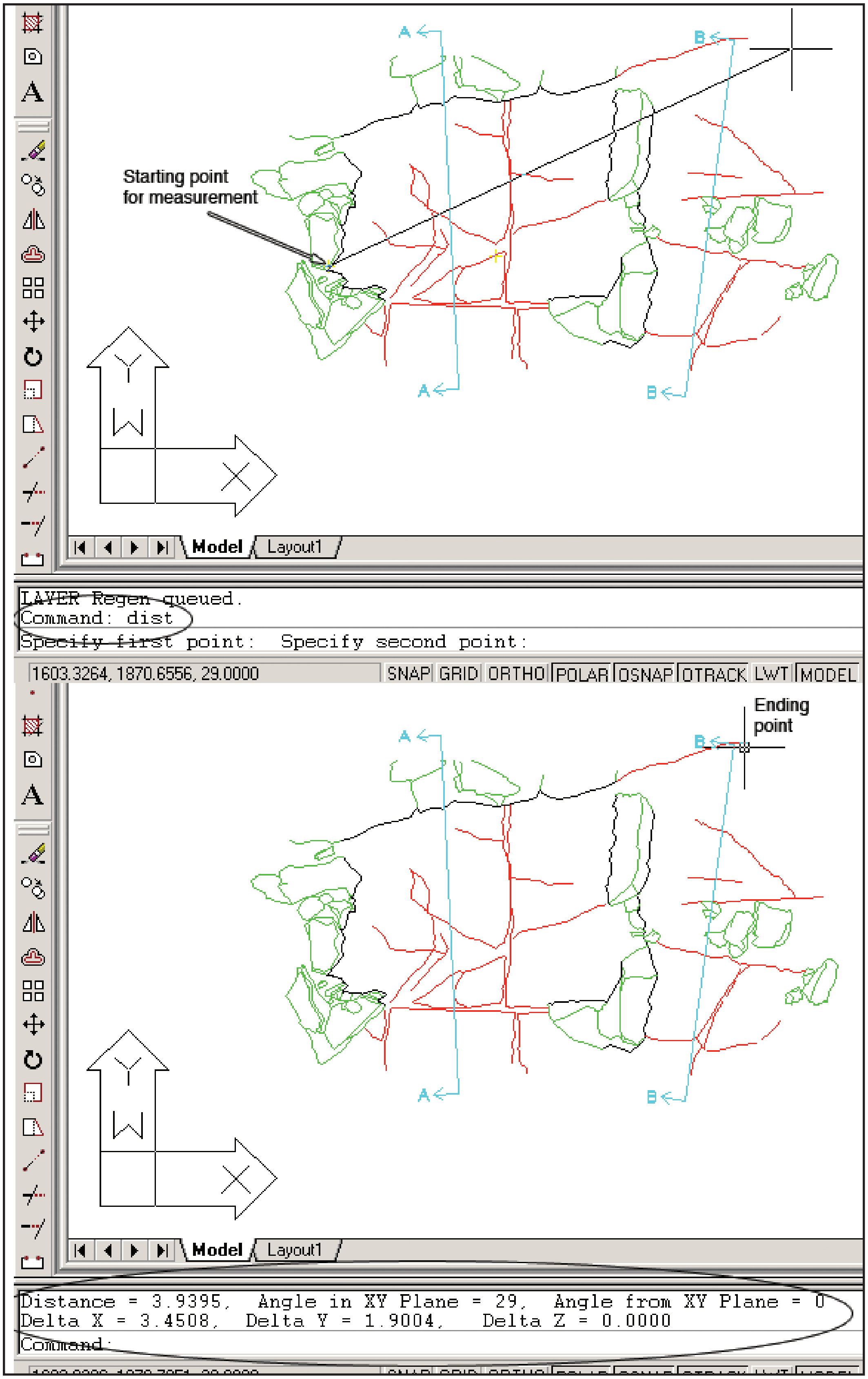


Figure 11

AutoCAD at two stages of a dimension query. In the upper image, the dist (distance) command is in process, with the cursor position showing as a cross-hair. In the lower image the command has been completed and the results show in the lower portion of the image.

The need to see certain portions of the model – and not others – applies to any drawing to be produced for publication or illustration. The drawing above of Tomb 2 (figure 8) was made by including only the segments desired, excluding all others such as the contour lines.

There is yet another reason for segmenting the model into so many discrete units. Each of the segments not only makes analytic sense, but each may be emphasized differently in a paper drawing – and, in fact, differently in one paper drawing as compared to another. Thanks to the drawing segments, it is simple to apply a line thickness (or color if colors are used) to each entity in a data segment in any paper or on-screen drawing. Thus, any given drawing shows lines with varying thicknesses (or colors) so as to make clear the points at issue. For example, the cracks in the floor of a tomb may be drawn with very fine lines while the outlines of whole blocks are drawn in heavier lines and the general outlines of the tomb are in yet heavier lines. The drawings in figures 8 and 9 show such control of line weights, accomplished with the aid of model segmentation.

Segmenting a CAD model is one of the most important things one can do when building such a model – and one of the most difficult. Segmenting the model is difficult, ironically, because it seems so easy. The natural tendency is to call a segment by some obvious name, say tomb1. Eventually, when it is clear that tomb1 should actually consist of several segments – one for tomb 1's elevation markers, another for the text for those markers, another for the tomb outline, and so on – the names get more and more complicated.

As segment names get more complicated, not only is it more and more difficult to create short names, but maintaining consistency becomes a problem, and it gets harder and harder simply to remember the names that have been created. It also gets harder and harder to call up more than one or two segments because of the length and complexity of the names. Pseira provides an excellent example.

First, all of the names for the Pseira segments used abbreviations so that the names would be short. For instance, there was no need to use tomb in the names; T plus the number of the tomb would suffice. However, it was necessary to use two numbers for all tombs, even those numbered one through nine; otherwise, the names of some model segments would have more characters than the names of others. (The importance of this will become apparent shortly.) For each tomb there might be general outlines, block outlines, block cracks and holes, floor cracks and holes, section lines, elevation markers, and a final category for not applicable or no category (the reason for which will also become apparent) – thus, six segments per tomb, like this:

T01B - Tomb 1, individual blocks
T01C - Tomb 1, cracks and holes in blocks
T01E - Tomb 1, elevation markers
T01F - Tomb 1, floor cracks and holes
T01G - Tomb 1, general outline
T01S - Tomb 1, section lines
T01Z - Tomb 1, no category

The same set of segments would exist for each of the tombs; so anyone using the model would know which segments contained which parts of the model. Any segment name with "T," two numbers, and "B" must show individual blocks.

This way of naming segments is especially valuable with AutoCAD – where the segments are called layers, the term I will use from here on. In AutoCAD (and only one or two other CAD programs), it is possible to call up the layers for inclusion or exclusion with simple search specifications that make manipulation of layers both easy and extremely flexible. Effective search specifications are easy to create using just two wildcard characters – ? to indicate any single character in that specific location and * to indicate any number of characters beginning at that location. (There are other wildcard characters that further enhance the search possibilities.)

Considering the Pseira names outlined above, for instance, a search for T01* would find all layers having to do with Tomb 1. (Note that this shows the reason for using two characters for the tomb number. Had the layers for Tomb 1 been labeled T1..., a search for T1* would have found not only Tomb 1 but, in addition, any tomb numbered from 11 to 19. If there were more than 99 tombs, therefore, we would need three characters for each tomb name.)

Well-constructed layer names also permit much more interesting searches. For instance, all the tomb models include both outlines of individual blocks and a general outline intended to show the general shape of the tomb. In a drawing showing all the tombs, one might want to see only the general outline for each tomb so that the drawing would be more clear. A search for ???G (or T??G) would find only those layers with the general outlines and would find those general outline layers for all tombs. A search for ?0?G (or T0?G) would find all general outline layers for tombs one through nine. (Note that the *?* was used, in part, to make certain that the searches were aimed at the correct character location in the layer names.)

Using Layers to Assist in Creating Good Paper Drawings

There is great practical value to separating parts of a model carefully and with an eye to finished drawings. Staying with our Pseira example, consider how finished drawings were made. First, the material had been separated by categories that included important distinctions intended to show in the finished drawings – general outline vs. block outlines vs. cracks and holes in the blocks vs. cracks and holes in the "floor." Second, AutoCAD permitted layers to be given specific colors, and the color assignments could be made by category, using the search tools described in the discussion of Pseira's layer-naming system. Third, when the drawing was exported to an intermediate program the colors were used to assign line weights so that the general outline was drawn with a heavy line, the block outlines with a lighter one, and cracks and holes - whether in the tomb floor or individual blocks - with a very fine line. Similar control of line weights could have been maintained within AutoCAD to make paper drawings directly.

The Pseira layer-naming system described above is actually simpler than the one used. Because the model included section views as well as plans, added layers and another character were needed to permit us to separate plans from sections. The inclusion of text, which might be used for some drawings but not others, also required more layers and another character in the name.

The final scheme included two new characters at the beginning of the name, one to indicate plan or section, another to indicate text or drawing.

The new layer-naming scheme resulted in many more layers, with some of them being the following:

PDT01G - plan, drawing (as opposed to text), Tomb 1, general outline
PTT01E - plan, text, Tomb 1, elevation marker
PTT01G - plan, text, Tomb 1, general outline
PTT01Z - plan, text, Tomb 1, no category
SDT01G - section, drawing, Tomb 1, general outline
STT01G - section, text, Tomb 1 general outline
STT01Z - section, text, Tomb 1, no category

Note in this list of layer names that each character takes on meaning in combination with the others. Thus, the text in the layer named PDT01G should apply to the general outline of the tomb, but the text in layer PTT01Z applies to what? In this case, we use the no category indicator to indicate that the text applies to any or all plan elements for Tomb 01 (but not to section elements which are labeled with text in layer STT01Z). Similarly, layer PTT01E contains text from the elevation markers only (the actual elevation associated with each marker).

The layer names for Pseira actually became one step more complex because some tombs had more than one section. To maintain a distinction between the sections, another character was added, after the S for section, to indicate section A, B, or C (or not applicable). So the final scheme was this:

PZDT01G - plan, section letter not applicable, drawing, Tomb 1, general outline
PZTT01G - plan, section letter not applicable, text, Tomb 1, general outline
PZTT01Z - plan, section letter not applicable, text, Tomb 1, no category
SADT01G - section, section A, drawing, Tomb 1, general outline
SATT01G - section, section A, text, Tomb 1 general outline
SATT01Z - section, section A, text, Tomb 1, no category

The names above seem to be complete, but they do not include any of the layers that might be called background layers: the survey grid, the contour lines, the coastline, the north arrow, the scale, or the text for either contour lines or the grid. It is not necessary to make the system more complex to include those items. Instead, we simply gave them one-letter names that used letters different from those permitted in the base system's first character. Thus, layer G contained the survey grid, layer C the contour lines, and so on.

As should be evident, this way of naming layers enables access to layers much as one might access data in a database. Different parts of the model can be called up or suppressed according to the analytical and practical content of the layers. Indeed, naming layers in this way – in order to permit access by analytic or practical criteria – is one of CAD's most beneficial features for scholars. It aids in the analytic process as well as in the process of producing good, effective drawings.

Note that not all of these layers shown above would actually be needed; there would be no text layer for general outline, for instance, since text referring to the plan of a tomb would be placed on the layer PZT##Z. In fact, one of the hidden virtues of the layer-naming system is that one may search for material by layer name to learn whether any such material exists. If there is no layer with the appropriate name, there should be no material fitting the category or categories defined. For instance, a search of the Pseira layers for ??T2* would produce no layers, showing that there are no tombs numbered from 20 to 29.[2]

The layer names used for Pseira could have been very differently constructed while remaining equally effective. The point is to exploit the capabilities of the software, in this case AutoCAD, and to provide both analytic and practical aid to the scholars using the model. (As noted previously, very few programs provide the same flexibility for naming and searching layers. That is one of the primary reasons to use AutoCAD.) One temptation you will face is that of using one character to convey two different meanings. In the Pseira scheme, for instance, we could have combined some of the characters, but we would have lost the easy searching possibilities and made the use of the system both more difficult and less flexible.

As is so often the case with computer technology, very careful advance planning is required – in this instance to construct layer names that provide the maximum benefit. Unfortunately, that planning will rarely be sufficient in anticipating all the needs of the system, and it may be necessary to modify the system as the project develops. Layers can easily be added and renamed, and drawing entities can be moved from one layer to another; nevertheless, it is far preferable to design the system well in the first instance. Changes can be time-consuming, and they tend to introduce error; so it is best to need as few changes as possible. When changes are made, it is critical to create copies of the model before the changes

[2] AutoCAD, like most CAD programs, cannot accept a command that leaves no active layer; so searches for layers that may not exist might not work as expected. For that reason and to provide for generally simpler layer manipulation, I always include a layer Z in any scheme. That layer is kept empty and is used only to provide a kind of default layer that should always be available and empty.

Programs like AutoCAD have a layer 0 (the number zero, not the capital O) that is the default layer, and it should not be used for model entities since its name cannot be fitted into the layer-naming system. It could be used like my proposed layer Z, but I use layer 0 for a copyright statement that always displays when the CAD model is opened. Since layer 0 cannot be eliminated, holding the copyright statement seems a better use for it, and simply turning off that layer removes the text in the copyright statement from view.

and to document carefully the changes and the mechanisms for making them. Of course, the layer-naming system must be documented, and the records must show that system at all stages in its development.

An important note about layers. In AutoCAD at least, layers can be locked, protected from any change. That permits users to safeguard the analytic structure of the model so that use of the model need not put it at risk.

Paper Drawings

CAD makes it possible to produce better paper drawings. That is a bold statement, and some would dispute it, arguing that the artistic touches that make the best of archaeological drawings so good cannot be obtained with CAD. That is true in the sense that CAD programs alone may not produce superior artistic effects; however, using CAD output in conjunction with a drawing program such as Adobe® Illustrator®, CorelDraw®, or Canvas® can yield those effects. In fact, for published drawings, as opposed to drawings to be used by project personnel, I recommend the use of a drawing program. The basic drawing may not need the assistance of the illustration programs, but for text and other effects, they are preferable. More on that below.

One of the reasons CAD drawings can be better than hand-drawn ones is that it is so easy to remake a drawing – again and again if necessary – to get it just right. The process starts at the most basic level with the selection of layers to be included in the drawing – the kinds of material to be included – and the area to be included. The area to be included obviously affects scale, and scale should be established at the outset so that a group of drawings can be produced at the same scale if desired or different drawings can use related scales. The chosen scale should also be selected with the finished size in mind, preferably so that no reduction in size is needed for publication. If reduction will be required, it is safer to plan so that all drawings are reduced by the same percentage (for the sake of line weights). There must also be decisions about line weights to be applied to different material (again using the layer-naming system to advantage by assigning line weights to layers, singly or in groups). Finally, text and added features such as a scale may be needed. They might be added to layers reserved for printed output or put on layers intended for labels, as the CAD specialist prefers. Finally, a print (often called a plot in CAD programs) can then be made to check the results. Because making additional drawings is so easy, adjustments can be made until the end results is precisely what is desired. Line weights, scale, and inclusion/exclusion of data segments can be adjusted again and again, with minimal labor expended.

The tomb drawings from the Pseira material (figure 8 and similar ones for the other tombs) each showed different selections of material, each being limited to the information about a single tomb plus the excavation grid (only enough to show location) and the scale and north arrow. However, all used the same line weights (and scale). Making the first drawing took some time and several experiments to get the effects desired. Making the others took only a few minutes.

Another advantage of using CAD for drawings is the fact that using CAD can eliminate the need for a draftsperson to produce a drawing at any size other than the intended publication size, making the draftsperson's control over the finished product much tighter. If there is no need to plan for enlarging or reducing paper drawings for publication, all the issues about line weight, fill, hatching, text, and so on can be decided from the beginning by the draftsperson. So long as the drawings have been produced correctly, they should be printed correctly.[3] Of course, the ability of the CAD technician to produce drawings to specified size depends upon good communication between the technician and those in control of the actual

3 The real world does not work so well, see box on p. 154. The number of possible glitches between the creation of a good drawing and its printed descendant is nearly infinite. The situation is improving, but problems will doubtless remain so long as there must be changes in media from first production to final publication.

publication processes, something the technology cannot guarantee.

There may be file-format issues with CAD drawing files for publication, as opposed to paper drawings, as publishers move to more completely digital processes. When a publisher needs to use all digital sources, the drawings must be computer files, not paper drawings. CAD programs will permit the user to create a file rather than a paper drawing when required; in such cases, however, the file format will be of considerable importance, and the CAD technician must be certain to provide files that can be used effectively. (Generally speaking, PostScript® is the format of choice for publishers.)

There are many instances when direct CAD output is not the most convenient or efficient drawing form. For instance, if multiple views from a drawing are to be combined, it may be much easier to import CAD output into a drawing program such as Adobe Illustrator, CorelDraw, or Canvas and finish the drawing there. The Pseira Tomb drawing in figure 8, for example, includes both plan and section views. While it is theoretically possible to produce something very similar in AutoCAD, Adobe Illustrator served as an intermediary. It was easier and quicker. When a great deal of text must be used or artistic effects added, it is also preferable to use one of the drawing programs as an aid.

When using an intermediate program, a CAD file can be used as the starting point (often in more than one format, depending on the situation). In the case of the Pseira drawings for publication, for example, an Illustrator base file was created with bounding box, scale, and general label. Into that drawing was inserted a CAD output file (PostScript format) that could be scaled, moved, and oriented to fit. Line weights were easy to adjust either in the CAD file or in the Illustrator file.

It must be clear that the production of paper drawings may serve either of two functions – publication or internal use to illustrate, explain, amplify. If a drawing has been prepared for publication only, it need not become part of the project archives, and the file may be deleted from the archives. If, on the other hand, a drawing is intended for use by project personnel as an aid to understanding, it should become part of the archives, in which case its creator, presence, use, and history must be documented (see Chapter VI, p. 211). An alternate preservation process would require documenting the specifications for a drawing rather than saving the drawing itself – layers included, scale, colors chosen, etc.

There is one serious difficulty with paper drawings produced via CAD, whether with or without drawing programs. None of these programs can easily produce the truly irregular effects sometimes desired. Stippling or shading or indicators of texture may be more easily drawn by hand then on a computer. It is not that the computer cannot be used, but computers cannot make rough and irregular lines easily. It may therefore be preferable to use the computer to produce a drawing that will, after completion, be modified with additional hand-drawn stippling, shading, or other irregular effects. The choice in such cases will have more to do with the draftsman's familiarity with computer tools and hand-drawing techniques than the actual capacity of either approach.

To recapitulate, the use of CAD for the Pseira tombs brought the following benefits:

1. The entire cemetery, including both plans and sections of all tombs, was modeled in a single computer file or model and accessed as required.

2. Geometric/survey information could be retrieved from the model, with accuracy and precision, rather than from scaled drawings. (Accuracy and precision are, of course, always limited by the modeling process, which, in turn, depends upon the survey systems used.)

3. A user of the model, at a computer with AutoCAD, could obtain virtually any view of the whole cemetery or individual tomb(s) conceivable.

4. A user of the model could call up material for viewing (or suppress material) according to a variety of analytic and practical criteria.

5. Paper drawings could be produced easily and quickly, and the draftsperson could change line weights, colors, and other effects easily and quickly.

6. As with other digital data, the model, portions of it, or drawing files from it could be sent to anyone via the Internet at no cost or on a disk for the negligible cost of the disk and postage.

Using CAD Programs

We will not try to discuss actual modeling procedures here, because each CAD program is different. However, using any CAD program requires a change in approach for anyone who has done manual drafting. All manual drafting is a matter of making lines, whether straight lines, arcs or circles, or less regular curves. CAD modeling is different in that the operator must self-consciously define the kind of line to be drawn before starting to make it: straight lines, individual segments of multi-segment lines, arcs and circles, or free-flowing curved lines. Any program requires that the drawing entity be specified prior to modeling it.

All drawing programs also require an explicit Cartesian grid or coordinate system, as noted previously – or impose an implicit one. Working in 2D, of course, requires only x and y axes (and the 0,0 point). In 3D settings, the z axis must be added.

As may be obvious to some, working in a coordinate system means that no modeling work begins, as a paper drawing may, at an undefined point, with all subsequent points being related to it but none having a defined relationship to the real world. Instead, every point will be defined in terms of the coordinate system in use; even picking a seemingly random point on a computer screen with the mouse defines a point with x, y, and z values. Thus, the first point in a model is as well and explicitly defined as the last. The coordinates of all points are – implicitly or explicitly – defined.

Using Colors or Styles for Meaning

It would seem that it makes sense to use colors and/or line weights and/or line types to confer meaning in a model – rather than or in addition to using layers. Red might indicate hypothetical additions to a model, for instance. Broken lines might also be used for restored material. These are, indeed, good ways to show differences between and among parts of a model with drawing choices. However, using them to categorize portions of the model instead of layers is not recommended.

The most important reasons is simply that searching the model for items with specific characteristics requires using the layer system, not line types or colors. Since the database-style access provided by the layer names is so valuable, this is a critical advantage.

In addition, it is generally easier to change colors and line types for a temporary reason, e.g., for a specific paper drawing, and then to be unable to find and undo those changes. Changes not undone would alter the information stored if the model were saved in its new condition.

Changing the layer on which objects have been modeled, on the other hand, can be a more complex process, though recent programs have made that easier. More important, it is never necessary to change the layer on which an item has been placed to produce a needed drawing or on-screen view, Furthermore, layers can be locked to prevent changes. As a result, using layers to convey meaning is safer and to be recommended.

Colors or line styles may and should be used to assist with making individual drawings or screen images clearer, but they should not be used to convey meaning within the model itself. If the color, weight, or type of a line implies no meaning, changing any of them for a given drawing carries no risks.

There is one unusual model entity common to most CAD programs that has no analog in paper drawings and should be understood. In AutoCAD it is called a polyline. It is simply a group of line segments treated as a single entity. The advantage of using the polyline or its equivalent lies in the ability to modify, move, rotate, copy, or scale the entire entity as a unit, not line-by-line. For instance, one command will change the layer for a polyline; selecting each individual line is not necessary. In addition, a closed polyline is equivalent to the GIS closed polygon in that all line segments meet at vertices, and the figure closes on itself.

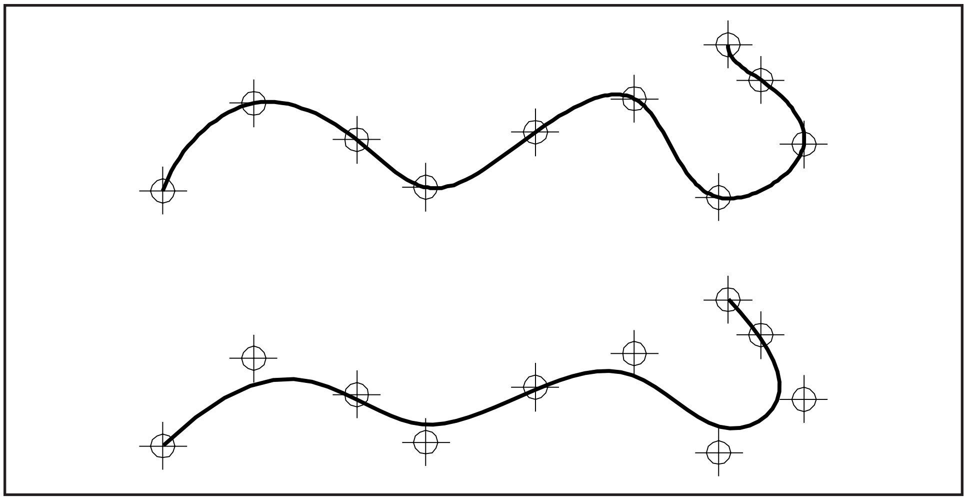


Figure 12

Two continuous curves from AutoCAD. The upper curve actually passes through all the points (the cross-hairs); it is a NURBS curve. The lower curve, a spline, actually passes through only the first and last point.

Continuous, irregular curves are also entered somewhat differently in a CAD system. The most common need for continuous curves involves a series of known points through which an undulating line must pass. There are different kinds of curves, as illustrated in figure 12, and one must be sure what conditions are to be met before determining which kind to use. NURBS (non-rational, uniform B-splines) will pass through all points, but quadratic and cubic splines will generally not, being pulled toward the points but not necessarily through them. (Older versions of AutoCAD permitted users to draw quadratic or cubic splines; NURBS were added some years ago. The latest version I have used, AutoCAD 2007, now draws a NURBS curve with the *spline* command. An undefined spline – quadratic or cubic is not stated – can be created by drawing a polyline and editing it with the *spline* sub-command. Other CAD programs will have their own ways for dealing with curves.)

All model entities will have specific, identifiable, unambiguous coordinates for all defined points – whether or not the model maker thinks about those coordinates. Simply making a mouse click, for instance, to begin a line will provide just such a point with coordinates, though the coordinates have not been chosen with precision.

The lines that will appear on screen or on paper – whether straight or curved – can be given colors dependent upon the model layer on which they are placed or individually specified colors. That is, any layer has an associated color that is the default for entities on that layer, but each entity can be assigned its own color. Line weights on paper can be similarly applied to specific lines or to groups or layers; screen line weights are sometimes adjustable.[4]

Lines may also be given types, which is to say that they may be broken lines, one of a number of specific sequences of long and short broken lines to indicate specific shapes (as used in standard engineering drawings), or otherwise made distinctive. Such line types are hold-overs from manual drafting, and they are very rarely needed by archaeologists – or, for that matter, anyone modeling in a 3D environment. A wide variety of line types is supplied with most CAD programs so

[4] The ways programs deal with line weights and colors are not standard; so the possibilities vary. With AutoCAD, for instance, the easiest way to set line weights for a drawing is to relate a given weight to a color, making the use of color and line weight cooperative. Line weights can be directly applied to the on-screen drawings independent of color, but, as noted above, it is hard to distinguish between and among them unless the differences are gross; so it seems preferable to use colors on screen to specify line weights for printing.

Coordinate Systems

Cartesian grids or coordinate systems are familiar to most people from early geometry classes. Formulae are often plotted on x- and y-axes to show how the variables change in tandem. Of course, many small maps use Cartesian coordinate systems as well.

In most CAD systems the Cartesian coordinate system can be three-dimensional, having x-, y-, and z-axes, and such a grid can be used to define unambiguously the precise location of any point; in the absence of the third dimension, the assumption is that all elevations are at zero because they have not been specified. Whether operating in two dimensions or three, any CAD system opens even a new document with an implied coordinate system - a 0,0 point and two axes or a 0,0,0 point and three axes. The particular coordinate system used for a given project, however, can be quite arbitrary. That is, one can use a 0,0,0 point and an orientation of the axes that are convenient for the project - related only to the site, building, or datum point for instance - but unrelated to any real-world grid or to the cardinal directions. The entities modeled can later be moved and rotated to be put into different locations so as to fit a different grid or coordinate system. The geometric relationships between and among the parts of the model would not change as a result of such movement. (Individual entities or groups of entities can also be scaled; their sizes can be adjusted by a ratio or an absolute scale factor. The internal geometric relationships between/among the objects scaled are unchanged by scaling.)

One of the most difficult aspects of CAD for many users is the generation of alternate coordinate systems in a model for specific purposes. If the position and orientation of a coordinate system may be arbitrary, they can also be changed, even temporarily. That is, the 0,0,0 point can be moved and/or the axes re-oriented. The effect is equivalent to moving and rotating the entire model with all its entities, but the basic coordinate system is retained, to be reactivated when desired. That is, when modeling some material the coordinate system can be temporarily changed without affecting the geometry or the base coordinate system.

The idea of an alternate coordinate system may seem both strange and useless, but it can be remarkably useful to define additional coordinate systems for use with a specific part of a model. For instance, an archaeologist working with a cut-stone building block with a raised portion on one face (a lifting boss or a decorative band) would likely measure from the edges of the face to the relevant points, treating the individual surface as a basic horizontal surface regardless of its orientation. Those measurements might be inadequate when trying to relate them to the standard grid. In such a case a coordinate system based on the face – 0,0 at the lower, left corner, the x-axis along the bottom, and the y-axis along the left edge – would permit the model maker to translate dimensions easily and correctly to the model. The plan view of the face of the block would be the same as the plan view of the new coordinate system. Once the face of the block has been completed, the model maker can return to the "normal" coordinate system, and all the material included on the face of the block will be correctly defined in terms of that "normal" coordinate system.

Alternate coordinate systems are easier to use than to explain. Making and using an alternate coordinate system must actually be done to be understood. Every time the process gets easier. Each time it is used it seems a more valuable and intuitive tool.

that printing them is easy. As with colors, line types may be assigned individually or by layer.

Areas within models may also be hatched – with a variety of supplied designs. The hatch boundary can either be created directly for that purpose alone or selected from existing lines in the model.

A simple 2D model can be created with just a few commands. Those that generate lines, multi-segment lines (polylines), circles, arcs, curves, and hatching are the only ones needed for most 2D models. In addition, one must know how to add text as necessary. That is straight-forward and simple. Fonts can be selected as well as various colors and effects such as bold-facing and Italics. Text can also be aligned, rotated, centered, etc.

There is one problem with text: its size. Text is normally given a value in real-world terms (e.g., .5 m.) to determine the height of each letter. The problem is that the appropriate size for text depends on the scale of the individual drawing produced with that text. While the size can be adjusted, it is time-consuming to adjust size for many different pieces of text intended to appear in a given drawing. (AutoCAD permits the user to assign text styles in such a way that changing many text entries at once is possible; other programs have similar features.) As a result, best practice is often to leave text out of the model altogether, either adding it only for a particular drawing or adding it with secondary programs used only to produce paper drawings.

The drawing of the tomb from the site of Pseira (above, figure 8) shows what can be produced from so simple a set of CAD commands. Only lines, polylines, and text were used to produce this drawing.

If the modeling process for the Pseira tombs was simple in terms of the number of commands required, getting the proper points into the model was not. Indeed, making sure that data have been entered correctly and to the highest level of precision possible is one of the most difficult parts of making a CAD model.

Entering Field Data Into a CAD Model

In the case of the Pseira tombs, each tomb had been drawn in the field or lab from on-site sketches and survey data. The drawings were to scale, but they were individual drawings of individual tombs, each related to the underlying grid but not to one another. They needed to be copied and inked for publication; more important, they needed to be put into an overall context. Each drawing was related to the survey grid; so all could be placed correctly in that common grid simply by creating the grid, placing each tomb in the proper location within the grid, and using a layer-naming system to keep track of everything.

Since field drawings had been made, the easiest way to make the CAD model was to trace the field drawings. This is relatively easy to do with a digitizing tablet, a kind of electronic drafting table connected to a computer. The surface of the tablet can used to trace a drawing – using the known drawing scale and position. Once scale and position have been established, every point on the tablet is a specific point in the model – x and y coordinates only. Of course, no z value can be implied by a point on a two-dimensional surface.

Tracing with a digitizing tablet has an impact on precision, however. First, the use of scaled drawings as a base reduces the recoverable precision; the basic data have already been scaled and have therefore lost some precision (though surveyed points entered as coordinates in the model retain their precision, presuming they have not been traced but typed as coordinates). Second, tracing a drawing by its very nature reduces precision as the draftsman selects one side or another of a line, changes angles of view, or otherwise makes slight adjustments. The result is certainly less precise than it would be with direct entry of survey coordinates, but, in this case, the precision obtained is more than adequate for the material. These roughly built tombs cannot be precisely measured in the first place. Pretending to achieve high levels of precision would be just that, pretending.

The paper drawings could also have been scanned and then brought into a CAD model as independent drawings. To integrate the information from the drawing, however, each drawing must be traced into the CAD model on-screen. Otherwise it will remain as raster data in a vector environment (see Chapter II, pp. 44 ff. and throughout the GIS chapter, Chapter IV); in addition, the CAD system will treat the entire drawing as a single entity, making editing and assignment of specific entities to different layers impossible. In most cases, tracing on screen is more difficult and less well-controlled than tracing on a digitizer. Precision of on-screen tracing can be affected by the resolution and quality of the scan; lower-resolution scans will produce less precision at the outset, and a poor scanner may distort one axis relative to the other. Using scans, though, may save money by eliminating the need to buy a digitizing table (assuming a scanner is available). If large drawings need to be scanned, though, readily available scanners will not be adequate.

If survey coordinates are available, simply typing them is probably the most common way to enter data. Typed coordinates normally require 3 coordinates – x, y, and z – but if a specific operation assumes a 2D process, the elevation must be stated, even if that simply means that the elevation is defined as zero. A point location chosen with a mouse or digitizer must also use a stated z or zero.

Especially in larger projects, input from a total station (electronic surveying instrument) is one of the easiest ways to get points into a CAD model. It is also one of the data entry systems least subject to error, since nothing needs to be typed or re-typed; the numbers are simply transferred from the data recorder of the total station to the CAD model, either directly or via an intermediate file. Some total stations can be used to draw directly in the CAD model through the total station, though the drawing commands are generally limited to simple 2D commands.

If only points are taken from a total station, they can be connected to one another and/or existing model points to create model entities in the entry process. (There can be a precision problem with total station input. The data may imply precision to several decimal places, but such precision is not warranted. The data transfer process should therefore be adjusted to round off the numbers transferred so that all digits beyond those deemed significant are zeroes.)

Total station data can also be transferred via intervening programs. The actual data points can be placed into a spreadsheet or text file and then manipulated to yield a sequence of commands that some CAD systems will process directly (and quickly and easily). For some examples, see *CSA Newsletter* articles at csanet.org/newsletter/nlxref.html#CAD and in particular the article at www.csanet.org/newsletter/spring04/nls0404.html, "From Field Data to CAD Model: Modeling the NW wing of the Propylaea," by Harrision Eiteljorg, II, *CSA Newsletter*, Vol. XVII, No. 1: Spring, 2004.

It is possible to work with some CAD systems via a specific computer language and to use that language to generate a routine that will insert data points from a file directly into the model. In general these input processes – whether via a spreadsheet, a text file, or a routine in the applicable computer language – are most valuable when they include modeling commands so that the data points are not only being brought into the model but are being used to create model entities – lines, arcs, circles, surfaces, and so on.

New 3D scanners may also be used to obtain data. Like total stations, they will provide 3D coordinates for points. Unlike total stations, the points are not selected by a surveyor but are all points within the area of view and falling on the grid intersections used by the scanner. Additional information about all these technologies will be found in the section on surveying later in this chapter.

There are three ways to use photographs for data entry. One, digital photos can be inserted into the CAD model and traced on screen (with the same process used for scanned drawings). The resolution of the photos, of course, limits the precision of the resulting traced lines (which must obviously be two-dimensional), as does the problem of relating the material photographed to the coordinate system. (This problem means that, in general, photographs can be used this way only if they are of flat objects such as a mosaic and they can be oriented so as to match the position in the model of the object.) Two, photos can be traced on a digitizer. The possibilities for using photos on a digitizer are considerably more interesting than using on-screen tracing because there are more ways to link digitizer locations to the CAD coordinate system. In particular, one may tape down a photo on a digitizing tablet, pick some known points in the photo, enter their coordinates (two-dimensional ones only), and then let the CAD system determine scale and orientation, even if there is perspective distortion in the photo. So long as the photograph is of a flat plane, the CAD program can convert points on a photo taken from virtually any angle into data points on the appropriate plane in the CAD model. This use of a digitizing tablet is not intuitive and requires considerable experience with the CAD program and the use of alternate coordinate systems, but very effective use of photos can be the result. (I cannot vouch for the possibility of tracing from photographs in programs other than AutoCAD.)

Three, photographs can be used with an intermediate photogrammetry program or a photogrammetry system. Photogrammetry programs are available for personal computers, and they are capable of providing excellent data. These programs depend upon good photographs, of course, and the requirements are more severe as the need for precision rises. Photogrammetry systems require either identifiable points that can be clearly seen in multiple photographs (close-range or convergent photogrammetry) or stereo photographs with expensive ancillary equipment (stereo photogrammetry). Close-range photogrammetry can be carried out by scholars, but stereo photogrammetry normally requires an outside contractor who has access to both the photographic equipment and the necessary computer equipment. Accuracy and precision from either system can be very high.

Transfer of actual model elements from one model to another is another way to work. Since there are few good intermediate file formats for transfer between models made with different software, this is not a very practical system when different programs are in use. The DXF format, which is an AutoCAD format but one made public and therefore at least somewhat less proprietary, is the most widely used for sharing models or model parts from different programs Photogrammetry programs may transfer drawing entities in this way.

Finally, manual-drawing approaches can also be used when areas have been measured with tapes and levels. For instance, it is quite simple in AutoCAD to indicate that a line starts at a given point and ends at a point 3.001 m. to the left (positive-x direction) , 1.001 m. up (in plan view, positive-y), and .003 m. higher (positive-z) The number of alternate data entry schemes that permit manual-drafting approaches is impressively large.

However one enters data into a CAD model – and few models will be constructed with only one approach – data input procedures must be documented to inform future users regarding available precision. The data entry procedures used have a significant impact on precision. Anyone using a CAD model should know as much as possible about those data entry procedures so that he/she can evaluate the model correctly.

Adding the Third Dimension

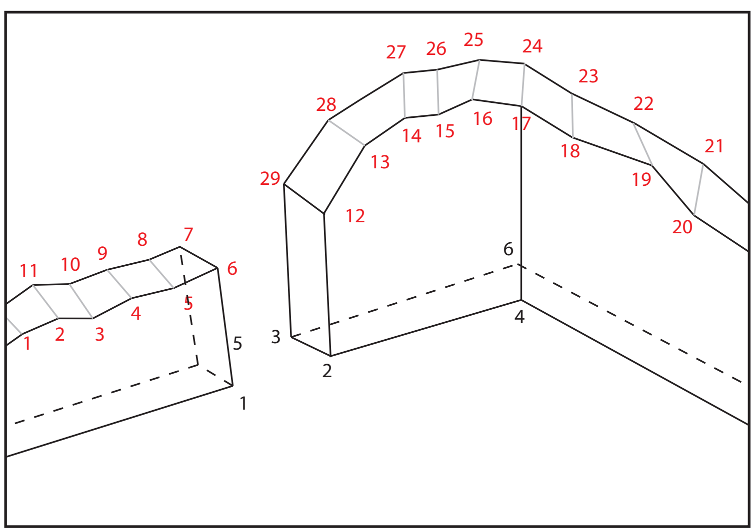


Figure 13

A comparison of the number of survey points required for a 2D model and a 3D model of the same wall.

The Pseira example has been an excellent one so far because of its complexity –and because archaeologists will very often use only two-dimensional drawings. However, one of the important virtues of CAD is its capacity to model real-world objects in three dimensions. Adding the third dimension changes the nature of the model significantly, enabling more lifelike views and far more complete and complex models. On the other hand, making 3D models requires far more survey data and generally makes it more difficult to produce drawings with an artistic flair. Adding the third dimension also assumes that doing so makes the model significantly more valuable to scholars; otherwise the costs in time and effort are too high.

The additional demands placed on surveyors by working in 3D are not easily understood until one looks at a typical archaeological setting and thinks about the requirements. Figure 13 provides an example showing how many more points would be required to make a 3D model of a simple room. In this drawing the points that might have been surveyed for a standard plan drawing are numbered 1 through 6 (in black). The additional points required to make a 3D model are numbered 1 through 29 (in red). Note that the number of required points for a plan only is 6 (including the out-of-sight corners numbered 5 and 6), but the number of survey points added for the upper portions of the walls is 29 because there was so much irregularity in the tops of the walls. Furthermore, each of those points must be a three-dimensional point. That is, the surveying must provide x, y, and z coordinate information; whereas traditional survey techniques often ignore elevation changes and rely upon occasional measurements of elevations. Were more points taken at ground level – a distinct possibility here and a certainty with irregular walls – the ratio of the number of points taken for a plan view to the number of points taken for a 3D view would be less dramatic, but the added burden would still be significant.

Despite the added demands of surveying for 3D models, the results may often be worth the time and trouble. A new example will show us some of the benefits of 3D models. It is the old entrance to the Athenian Acropolis (mostly but not completely demolished to make way for the famous building that stands there now, the Propylaea). Working in 2D only, we might have the plan and elevations in figures 14 and 15 available for study. Since the "structure" (too strong a word; it was just two dressed-up courtyards in front of a fortification wall and the steps connecting the courtyards) went through various phases, we might have the same set of drawings for each of the phases, using layers to separate the materials from various phases. Before going any further, please study these drawings carefully and make an attempt mentally to reconstruct the portion of the entrance area remaining from the phase represented here (the last one). It might even be worth making a simple sketch of the area to see how well the drawings have done their jobs.

The next illustrations, figures 16 and 17, are 3D views of the area. Both views were made from a 3D CAD model, and the two views manage to show the whole preserved area of the last phase of the entrance, though, as you would expect, a great many different views could be added. It is not likely that every reader will have imagined the area as shown here from the 2D drawings. The ability to conceive of 3D objects from 2D drawings is not widely spread in the population.

These drawings all come from the same model; they simply present the material from different points of view – plans, elevations, and axonometric views (lacking perspective). In this case, at least two different axonometric drawings from different points of view were required because the cross-wall interrupts any view and obscures different parts of the area in each view.

A 3D model also provides more geometric information to a user. Since all points are properly located in a 3D grid, point-to-point distances can be accurately calculated – in three dimensions. In addition, the coordinates of each point – including elevation – can be obtained on request.

Modeling in 3D has another important benefit. It requires geometric accuracy and precision of all reconstructed parts. That is, when adding hypothetical material to a 3D model the scholar must fit fully-defined 3D geometry to the model elements that are already well-defined 3D geometric items. If the added material is not accurately modeled, it may look correct from one point of view, but the errors will be clear in others. As any scholar knows, it is very easy to make a reconstructed drawing of some partially-preserved structure or site, but such drawings

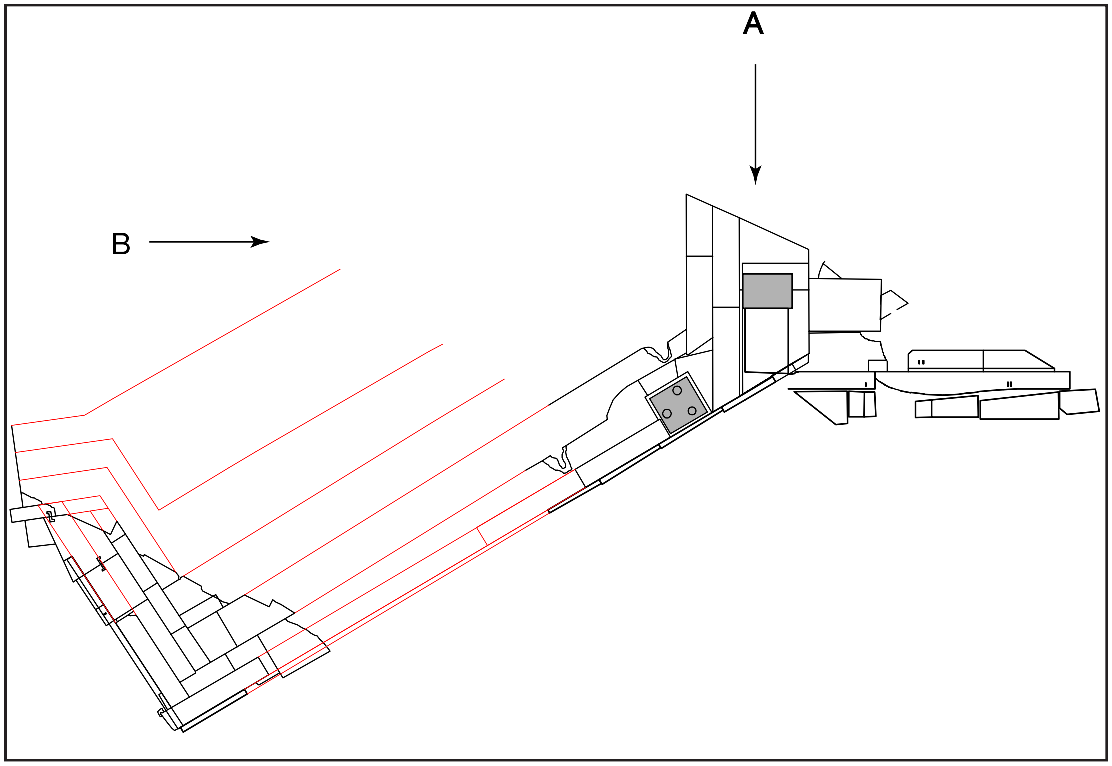


Figure 14
Plan view of the older propylon on the Athenian Acropolis (slightly simplified – red portions restored). Gray anta and tripod base for orientation.

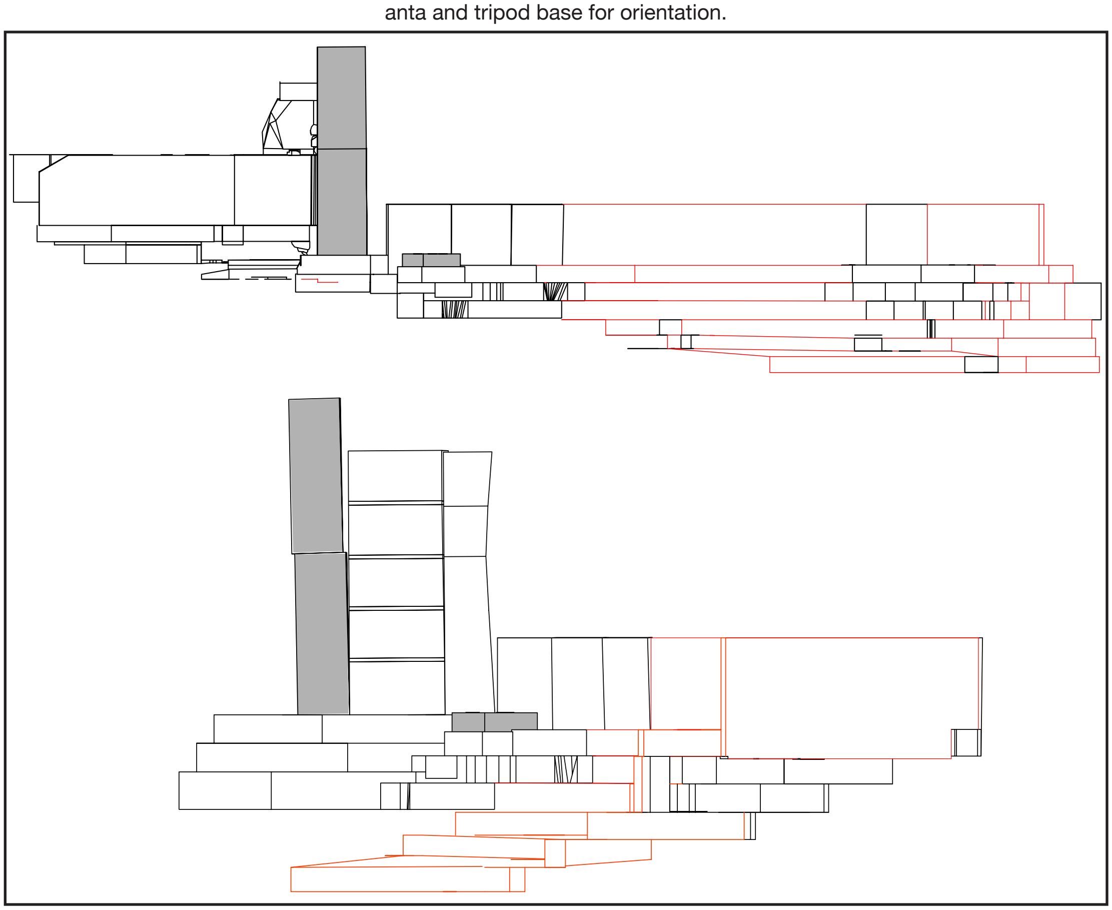


Figure 15
Elevation views of the older propylon - A above and B below (slightly simplified – red portions restored).

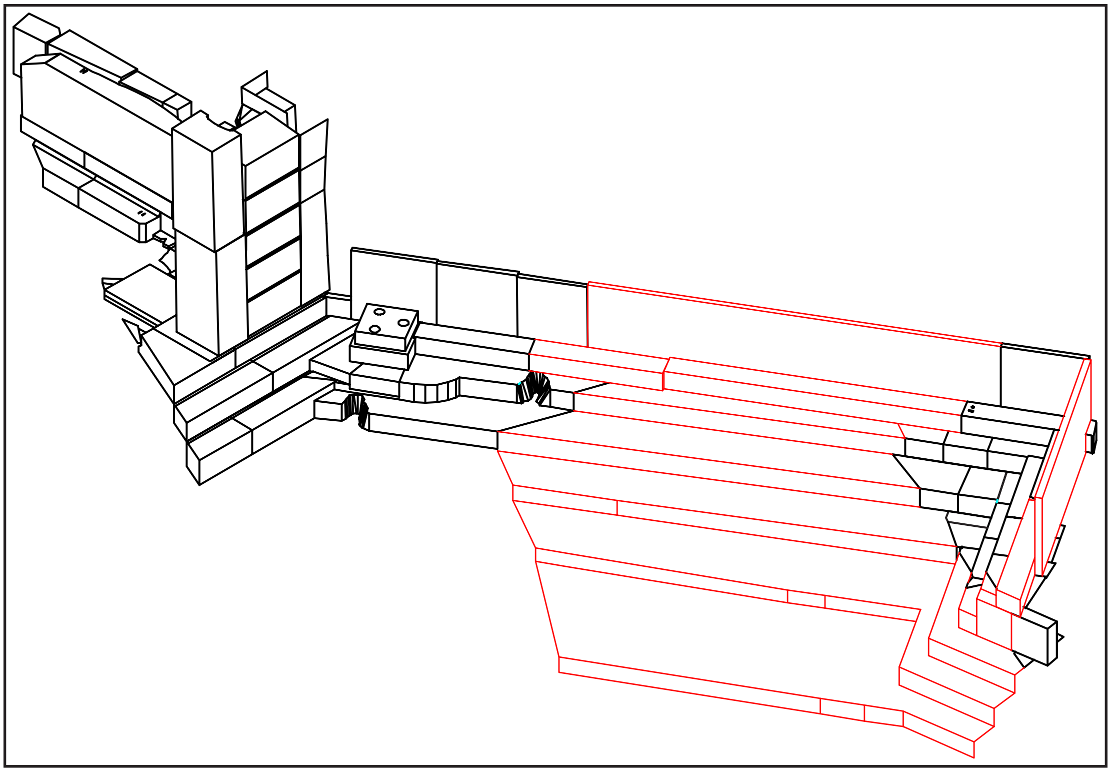


Figure 16
Isometric view of older propylon from northwest (slightly simplified – red portions restored).

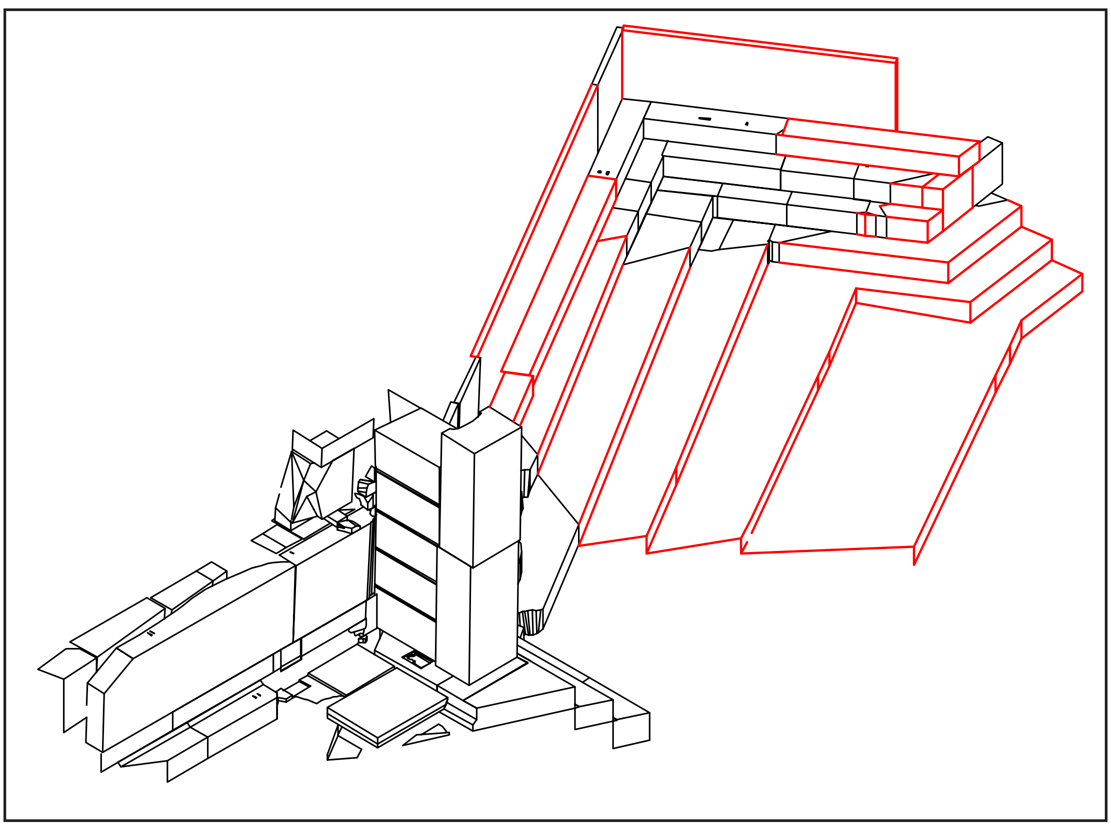


Figure 17
Isometric view of older propylon from northeast (slightly simplified – red portions restored).

are often based on imagination alone. In a CAD environment, they must be based on geometry as well as a good imagination.

3D Modeling

Models in a 2D environment consist of lines in space; but three-dimensional objects must be more than lines marking the edges of objects or surfaces. Consider the difference, for example, between a wire coat hanger and a piece of cardboard of the same shape. The coat hangar is effectively lines in space while the cardboard is an object that occupies space, interrupting lines of sight that are not interrupted by the coat hangar. A CAD system must have a way to understand the difference between the lines that represent the wires of the coat hangar and those that represent the boundaries of the cardboard, an object that occupies space. Thus, three-dimensional objects must somehow be defined as entities that occupy space; in a CAD environment, objects must be explicitly declared as surfaces or solids – not lines or circles or arcs making closed figures – in order to be treated correctly. That is, the CAD system must understand that certain entities are surfaces or solids in order to construct 3D views that mimic the real world, with some things hidden by others. (Compare the views of the child's block in figure 5.)

In a CAD environment an object can be modeled as a solid object or as a collection of surfaces. Both versions of the CAD model may appear the same, since CAD programs recognize that both solid objects and surfaces interrupt lines of sight to hide other objects or surfaces. It may seem that modeling the solid object is so much to be preferred that no program would bother to create surface-modeling tools. However, more powerful computers are required to model solid objects; so early CAD programs often dealt only with surfaces. As a result, there is a strong tradition of dealing with surfaces in some situations, solid objects (referred to simply as solids) in others. The dichotomy is a blessing for archaeologists because we so rarely have all of an object to model (for instance, a wall composed of individual blocks has some faces hidden from view – and from measurement – so long as the wall remains standing); so it is often impossible fully to model the items in a structure or excavation area that has not been completely dismantled or excavated. In fact, archaeologists generally make 3D CAD models of only the visible surfaces of real objects, not the entire objects. The surfaces, while neither as complex nor as complete as solid objects, are still very different from simple lines in space because one cannot see through a surface.

Any 3D CAD model must have surfaces/solids defined explicitly to know that the entity is more than simple lines or complex shapes and that it will obscure other objects in certain views. Compare figures 19 and 20, both drawings of the older propylon. If the objects had not been explicitly defined as surfaces, objects in the background would have been visible through the foreground objects as shown in figure 19. Figure 20, on the other hand, shows the more lifelike view that results from using surfaces.

The use of lines – and curves, points, and arcs – without surfaces is often called wire-frame modeling. That is, the model maker is creating just the lines that bound the objects being modeled. Modeling surfaces but not full objects is generally called, not surprisingly, surface modeling. Modeling complete objects is called solid modeling. As stated above, solids are not

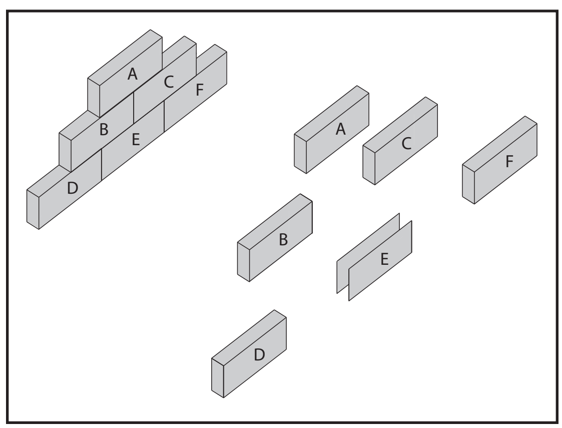


Figure 18

Blocks modeled as solids or surfaces. All the blocks In the upper-left group, seen as part of a wall, appear to have been modeled in the same way. As shown exploded, though, block E consists of only two surfaces. In fact, all of the other blocks could be either three surfaces (the only visible surfaces from this point of view) or solids. Only by querying the model could one determine which. (None of these blocks could have been made only of lines since all hide parts of themselves at the least.)

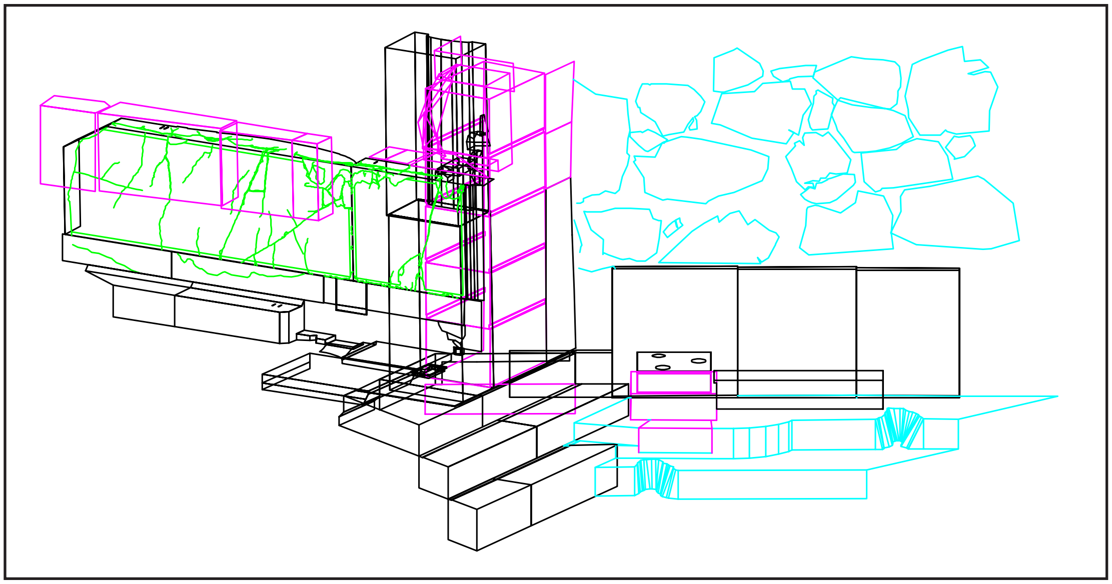


Figure 19

The older propylon seen as it would appear if modeled only with lines in space rather than surfaces. Lines that should not be visible from this vantage point are nevertheless shown.

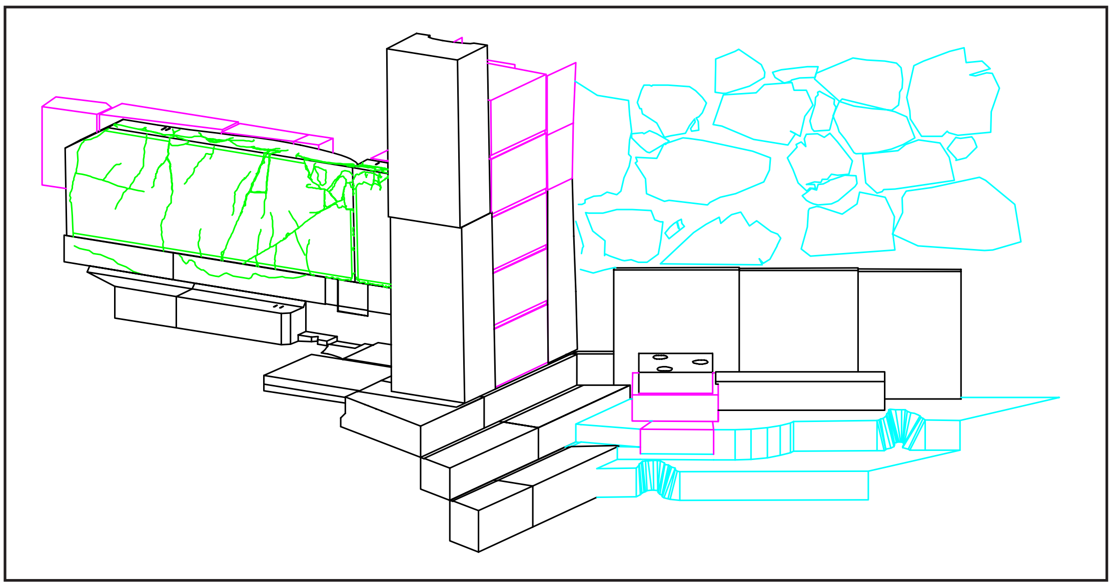


Figure 20

The older propylon as a surface model. Since there are explicitly defined surfaces in this case, the lines that should not be visible from this vantage point do not show.

generally modeled in archaeological field work because excavators so rarely have access to the full object with all its detail. Fortunately, most CAD programs permit simple lines, arcs, and circles to exist in models with surfaces and solids as well. The user need not model exclusively solids or surfaces and may readily place lines on a surface, as was done with the cracks in the wall block in figure 20.

Even though 3D models are considerably more complex than 2D models, making them requires using only a few more commands. Users will generally know and use more commands for looking at the model (pan and zoom, for instance, with sub-commands), changing point of view, selecting layers for display, and querying objects in the model than for making the model itself.

Even with that small number of modeling commands, using CAD in the field can be very different from using traditional drafting tools. The extent of the differences depends on surveying procedures, processes chosen by the draftsperson, and the intended end result. The fact that all measurements should be based upon point locations in a 3D Cartesian grid system has a considerable impact by itself. The use of specified coordinates means that the model-maker is often typing coordinates rather than "drawing." Even when making a new line of known distance and direction and avoiding explicit coordinates, the keyboard is usually more important than the drafting pen.

Although some draftsmen use CAD on portable computers carried right into the trenches, that is still rather rare, and batteries are not really up to such use yet, though promises of longer-lasting batteries are made regularly. As a result, the CAD model is generally constructed from sketches and measurements made in the field and brought back to the lab for later processing. This can create significant problems since it is often difficult to be certain that all data points needed have been fully surveyed. There is no simple rule of thumb for operating procedures to guide a CAD technician in such cases, but there are three good guidelines. One, do not save data for processing after leaving the field for the season. Even if all required data are available, it can be surprisingly difficult to model complex

Modeling in AutoCAD with the Keyboard

Ironically, one of the great virtues of AutoCAD descends from its age. At the time AutoCAD was developed, the user interacted with the program (as with most contemporary programs) by typing commands, not by selecting icons. Although icons are now commonly used in AutoCAD as well as so much other software, one may still enter commands by typing at the "command line," the area at the bottom of the AutoCAD window where two-way communication with the program is carried on. (See figure 11.)

It is now possible to paste text into the command line rather then typing it (or to save the text in a file with the .SCR extension and then call up the file). As a result, anyone using AutoCAD can model by creating text in any program (word processors and spreadsheets are commonly used by this author), pasting the text into the command line, checking to be sure the result is correct, modifying the text to repeat the process as necessary, and moving on to the next task. This is a remarkably efficient way to model when the coordinates are already in text form or when a long string of coordinates must be entered, making typing errors likely. One need only create text with all the characters that would be entered at the command line, including spaces and/or carriage returns; copy the text; and paste it at the command line. (It is critical to paste the text at the command line, not in the model. If the cursor is in the wrong spot – anywhere in the model – the process will simply place the text into the model at that location.)

Starting with data from a total station, for instance, a model maker might download a series of points (x, y, and z for each), remove any spaces so that all coordinates are expressed as strings of numbers and commas, add the AutoCAD command "3dpoly" and a return at the beginning, put a return after each set of coordinates, and a final return at the end to complete the command. Pasting that text into the command line would create a three-dimensional , as opposed to planar, polyline (AutoCAD's multi-segment line) starting at the first point, ending at the last, and with a vertex at each of the other points recorded by the total station.

Changing the command to "spline" and leaving all the coordinates in place would create a continuous curve through the same points instead of the line segments. Thus, this process permits a model maker to produce two versions of the same line, one consisting of connected line segments and the other being a continuous curve, for comparison; having created the command to make one figure, altering a single term permits the other to be made.

Spreadsheets can be used just as well as word processors to generate commands for modeling. Indeed, spreadsheets can often be used more creatively, but users must be careful to paste text as plain text – or to use a word processor as an intermediate program. (See *CSA Newsletter*, "A Spreadsheet as a CAD Aid - Again," by Harrison Eiteljorg, II, Fall, 2001, XIV, 2, at csanet.org/newsletter/fall01/nlf0105.html and prior articles referenced there.)

real-world items that cannot be seen. Two, finish modeling anything that is to be dismantled while it is still standing – for the same reason. Three, make certain that your field notes are complete and explicit; leave nothing half-explained.

Real Geometry vs. CAD Geometry – Data Density

Simple geometric shapes – planar surfaces, cylinders, cones, truncated cones, pyramids, and so on – are generally not very difficult to model, but there are usually problems with modeling even slightly irregular shapes in CAD. At the most basic level, a surface that seems to be a simple plane but is not truly planar can be very difficult for some programs to model – and virtually no real-world surfaces are truly planar. Even the carefully finished blocks of classical Greek marble buildings are not simple planes. Indeed, a machined block from a modern structure would not seem to have a truly planar face if surveyed with a total station.

Since computers must treat all terms in very literal ways, it is not so surprising that they have trouble with surfaces that are not truly planes. It is simple to specify three points that are the corners of a triangular plane; the resulting triangular surface is not a problem for any CAD program to model (although the actual surface may be irregular internally and therefore not truly flat). The fourth point, making the surface a quadrilateral, creates the problem. Such surfaces may be planes in a design, but in the real world – especially the battered and worn parts of the real world archaeologists deal with – a surface defined by four surveyed corners is never a plane. The work-around is obvious – make two triangular surfaces that join on a diagonal and treat them as one. The problem is the joint. If the CAD program cannot suppress the line indicating that joint, proper line drawings cannot be made. (Renderings often can make those joints effectively disappear, but line drawings are the more common form of presentation.) AutoCAD can model surfaces that are not planar – and will automatically do so when a user invokes the “3dface” command to model a quadrilateral surface, ignoring out-of-plane problems for the fourth point. Many other programs cannot manage that. That is one of AutoCAD’s critical advantages for use in archaeology.

In fact, that “3dface” command is the one used most often when modeling in three dimensions with AutoCAD. With that command (and the sub-command that suppresses joint lines) virtually any surface can be modeled save those based on curved edges. Modeling with the “3dface” command is remarkably flexible, but also remarkably tedious to use on surfaces with more than four or five edges.

The majority of the surfaces in an archaeological CAD model will be made with the “3dface” command. Even very irregular blocks can be modeled as a collection of quadrilateral and triangular surfaces. Doing so, however, involves necessary sacrifices; replicating the real world is practically impossible because that real world is so irregular. Consider the simple step shown in figure 22, for instance. Its top consists of two quadrilateral surfaces (1 & 6) and four triangular ones, and the cutting consists of several more surfaces. Even this many individual surfaces, though, cannot fully express the irregularities on the block. The edges of the various surfaces are straight, and the surfaces are effectively flat.

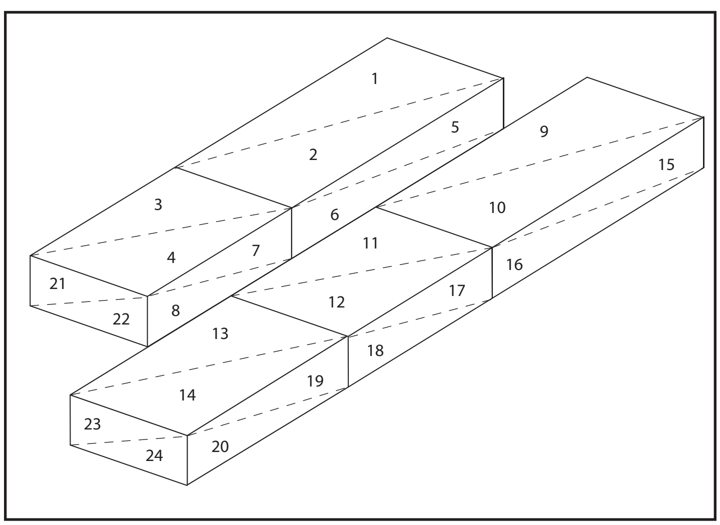


Figure 21
The faces of five steps showing with broken lines the triangular surfaces that make up each rectangular surface. A total of 24 triangles compose the 12 rectangular surfaces.

However, there will inevitably be deviations from straight and from flat in any real block. In the end, if the modeler is willing to make enough separate surfaces, any block can be modeled; however, the more precise and accurate the model, the more time and effort required – and the more survey points.

Modeling any block requires three kinds of simplification that reduce accuracy and precision. First, the surveyor must decide that some finite number of points is sufficient to define the perimeter and precisely survey those points that are critical to defining it. The surveyor is deciding that the block can be generalized from that selection of points. A decision has been made – often implicitly only – that any additional information gained by surveying more points is superfluous.

Second, the surveyor must decide whether or not to survey points lying within the perimeter, on the surfaces that make up the face of the stone. If such points are to be surveyed, of course, the surveyor must also decide how many points to survey and which ones.

Third, the CAD technician must decide which points go together to make the interior surfaces. That may not seem a problem, but the two different models of the block in figure 23 are both based upon the same 3D locations of the points making up the block's perimeter; neither uses points from the interior of the face of the block. Nevertheless, the two version are quite different because they combined points to make the constituent triangular surfaces differently. Had interior points also been surveyed, the number of different versions of the block would have grown, but the chances of modeling it accurately might not.

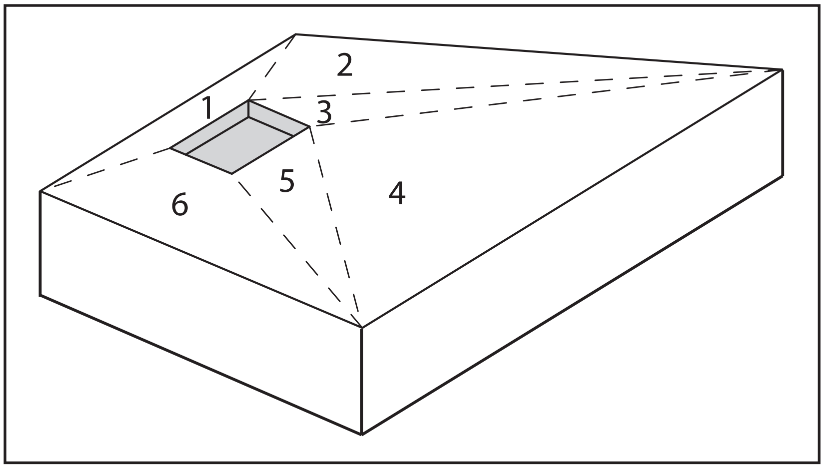


Figure 22

This block (a simplified version of the topmost step of the older propylon) has a cutting in it; so it had to be modeled of many individual surfaces that would define both the major surfaces of the block and the surfaces of the cutting. Specifically, this block consists of 6 surfaces that combine to make up the top surface of the block (numbered in the drawing). The broken lines would not appear in the model, but AutoCAD makes it possible to show those interior edges on command to assist with the modeling process.

Although neither of the models is particularly accurate as a representation of the surfaces of the original block, it is not clear that the inaccuracy is of any importance in this context. The position of the block is accurate. The surfaces making up its face were neither necessary nor intentional for the ancient builders of the wall, and a photograph can be used to illustrate the character of the wall. Is there any reason to worry about modeling the block more carefully? In some instances there may be; so these decisions about what points – and how many – to survey are important.

A CAD model of those blocks – or any large real-world physical object – cannot completely replicate reality. As previously stated, a model cannot contain enough data points to mimic reality faithfully. If data points were surveyed in a fine grid – measuring, for instance, only 1 cm. on a side or even a finer grid at 1 mm. on a side – would all necessary points not be found? It depends, of course, on what one means by necessary. But even a 1 cm. grid requires a number of survey points well beyond the needs of any practical survey plan, and it would surely miss some important points that do not conveniently occur at a grid intersection.

No paper drawing provides better results, of course, but we are accustomed to the limitations of paper drawings, and there is a natural tendency to assume that CAD models can magically overcome all the limitations of paper drawings. Quite the contrary, the CAD environment – especially if working in 3D – makes it necessary to be more self-conscious about the need for survey data – and the limits imposed by practicality, time, and money.

How many data points are required to define a rectangular stone? A roughly finished block? An adobe wall? A mosaic floor? The baulk of a trench? There is no answer to those questions. They cannot be answered by some abstract, theoretical choice. Rather, the archaeologist in charge, the surveyor, and the CAD technician must decide what level of data density is required for the work at hand and is also reasonable and cost-effective to obtain. It is obviously better to err on the side of more rather than fewer points, but practicality demands that reasonable limits on data density be found. It is imperative that the archaeologists determining those limits make the choices for good archaeological reasons, not for reasons having to do with the technology. The difficult question is thus the simplest question. How many data points are required (and where should they be)?

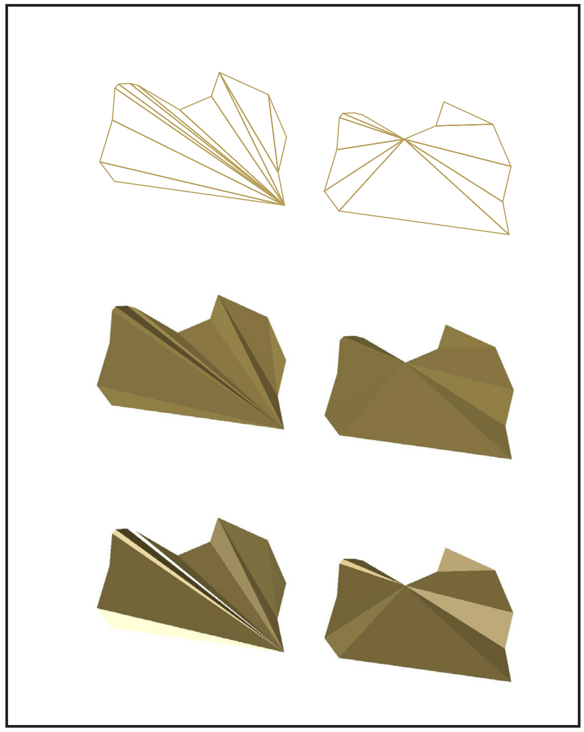


Figure 23
A single block of the fortification wall seen in figure 20, shown (top) as modeled in two ways, (middle) as rendered from the two different models, and (bottom) as shaded from the two different models.

Look back at figures 19 and 20 and examine the way the irregular blocks of the old fortification wall (upper right, blocks in blue) have been shown in these two drawings. There were many data points specified around the perimeter of each block, and the resulting drawings show a block with straight lines between those points. At the sizes of the two drawings, the stones look as we would hope. Figure 24, however, shows one of the blocks from that wall greatly enlarged, and the edges of the block there seem to be relatively long, straight lines. That is not the shape of the block; it is a compromise between the shape, which is far more irregular, and a model that could have been produced from even fewer survey points. It is an adequate representation – at smaller scales, more than adequate – but not a fully precise and accurate one. Was the number of points chosen adequate? Too many? Too few? Every scholar might have a different answer, and every answer would depend upon the aims, needs, and intentions of the individual project.

Figure 24
A single block of the fortification wall seen in figure 20, enlarged to show its consituent edges.

Objects with curved edges – whether as regular as a column or as irregular as an adobe wall – present similar difficulties. Any model is a necessary abstraction requiring compromises and forcing the model maker to decide what tolerances are appropriate. Whether for simpler blocks or more complex objects, however, few field archaeologists want to engage in a philosophical discussion of the data density issue or questions of permissible tolerances. They expect the site architect to decide such questions as part of the job, just as the site architect would naturally have decided how many points to take along the base of an undulating wall for a standard paper drawing fifty years ago.

Consider the drawing from Pseira shown at the beginning of this chapter. Relatively few survey points inform such a drawing; much is interpolated. No user of such a drawing would expect to take careful measurements of an individual stone or otherwise to use the details for some critical dimensional analysis. However, such is the nature of CAD that any point in a model can be queried for its coordinates. Therefore, the CAD technician must document the model so that any user knows which points were based upon survey data and which were interpolated.

Unfortunately, no standard exists for this issue, no agreed-upon notion of data density for structures or sites where 3D modeling is most likely to be needed, and the questions are very important for determining the nature of the CAD models that will serve as records for future scholars.

Some examples to make this discussion more real: a monument like Sacsayhuaman in Peru with large, shaped stones using no mortar – stones shaped with hammer stones, not cut to regular shapes; a cut-stone monument (also no mortar) like the Parthenon on the Athenian Acropolis; a mud-brick structure from the American southwest (whether ancient or brand new); a Roman concrete wall such as those built in Pompeii (some still covered with a veneer and some not).

Starting with the Inca structure from Peru, the problems and questions are very clear. A simple plan of this structure would be based upon the points where walls change directions. A 3D model could be made with nothing more than the heights of the walls at those corners, with photographs to be used to provide some detail. Would that be sufficient? Such a model could supply no information about the way the walls bond as they meet at corners, but, again, photos might do that. What about the stairs? Would it be sufficient to model them with corner points, even though the corners, like those of the walls, are not so easy to define?

Rendering and Virtual Reality Programs

Scholars will generally be satisfied with the images produced by CAD software, perhaps with assistance from illustration programs. Many people, however, want to see more realistic representations of structures, cities, or excavations. There are two different kinds of software designed to provide those more realistic views: rendering programs and virtual reality programs.

Rendering programs can provide extremely realistic views of a 3D model. The best of them can provide the realism of a photograph, complete with a variety of textures, reflections, shadows, and even vegetation. Such representations, however, must include information simply impossible to verify, assuming, for the sake of argument, the possibility of completely accurate geometry. For instance, the appearance of materials at the time of the completion of an ancient structure cannot be fully known; nor can the local vegetation or the full nature of the surrounding area. Indeed, one of the most difficult aspects of producing good renderings is determining what to do about the area surrounding a reconstructed ancient building. If added buildings are included, must they all be well studied and accurately modeled? If not, how does the rendering show the unknown or the partially known?

Virtual reality programs attempt to take rendering software a step further by permitting users to navigate through a fully three-dimensional world consisting of structures and surrounding terrain presented with nearly the same level of realism as a good rendering. In order to permit the user to navigate in real time (to move in the computer model without any apparent lag in time from instruction to movement) some realistic effects of renderings must be sacrificed, but the more powerful the computer the closer the views can be to the quality of a good rendering. The impact can be very strong. As with renderings, however, the virtual worlds suffer from the problem of the unknown. How does the system treat the unknown or partially known?

Regardless of the problems and benefits of using these kinds of software, good models for renderings or virtual worlds must begin with CAD solid or surface models, and, since our aim is documentation, CAD is our subject. We will not be concerned with rendering or virtual reality here, though their value for presenting archaeological information to the general public should not be underestimated.

It is easier to decide how to approach the Parthenon, but it is also much harder to survey it correctly. Because that building was constructed with unusual subtlety, a great deal of information is needed. In fact, the shape of each block should be surveyed. That requires an enormous amount of detail, but it does not fully satisfy. What about the shapes of the decorative pieces? Must each triglyph be modeled individually, or is it sufficient to know the basic exterior dimensions of each block, filling in the details with some "typical" arrangements of the parts? How are the columns to be treated? Should each drum be modeled separately? If so, with how much detail? The flutes are all potentially different, but how much needs to be determined about each?

Figure 25
The Inca remains at Sacsayhuaman in Peru.

Adobe or mud-brick structures may provide the most difficult subjects for CAD because, even when new, they are not simply irregular but intentionally so. The technique (or at least the style inspired by the use of adobe) is still used in Santa Fe and other areas of the American southwest, and new structures built of or to mimic mud-brick are common. The walls are not intended to be vertical but to sweep gracefully from their wide bases to narrower tops, often with window openings showing clearly the diminution of wall thickness. Wall tops are similarly straight but in a loose, non-mechanical way, and corners are graceful curves, not

Figure 26
The west facade of the Parthenon in Athens, Greece.

hard angular changes of direction. When such a building begins to decay, surveying becomes more and more problematic because it is impossible to reconstruct now-missing corners and lines as one might with a regular, geometric structure.

To survey new adobe walls would be very difficult. A few survey points would suffice, in terms of getting the important information, but the result would be a sterile and mechanical presentation of a decidedly non-sterile, non-mechanical structure. Surveying such a building in the process of decay would present even greater challenges.

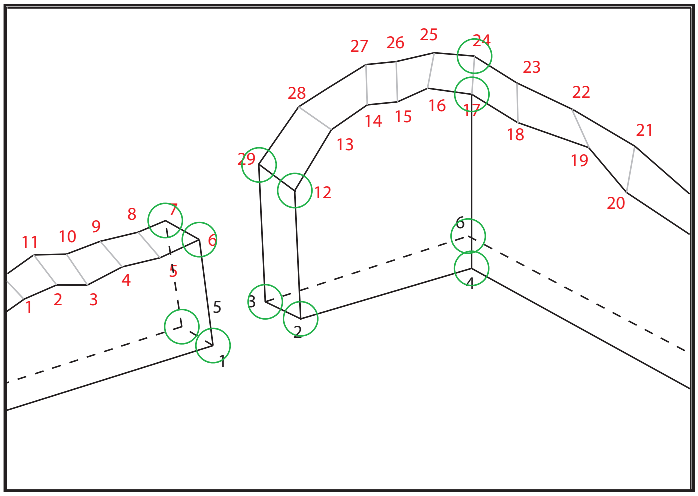


Figure 27

The wall shown in figure 13, p. 178. Were this an adobe wall, all the corners (circled in green) would have been intentionally curved, requiring much additional work for the surveyor.

Finally, there are the Roman concrete structures of Pompeii. Those with veneer remaining on the walls are somewhat easier to deal with. They were intended to be geometrically regular, with right-angled corners and vertical walls. It may therefore be reasonable to take relatively few survey points. The structures that have lost their veneer, however, are quite different. There the scholar is often left with shapes and surfaces that were never meant to be seen and may therefore be far more irregular than the finished walls. So what points should be surveyed?

In each of these cases the surveying and modeling questions are not easily answered, but there is one over-riding issue that can help determine approaches to the questions if not specific answers. The CAD model is the scholarly record of the structure. As such, the dimensions that can be retrieved from the model are those a scholar deems important for the record, not for the sake of visual appearance. Photographs can always augment the model – in addition to other forms of documentation – but nothing can provide information not recorded in the field and in the model as the final expression of a scholarly study. So the CAD model should provide proper 3D coordinates for all those points the scholar deems significant, and points not actually surveyed should not end up in the model unless the manner of their inclusion is made clear and explicit in the documentation so that they will not be mistaken as surveyed points.

Figure 28

A wall in Pompeii showing both the underlying

The foregoing examples were intentionally all structures. The general approach there – and the important differences from one building type to another – can equally well be applied to excavation models. Excavation trenches can be regularized without falsification if users understand the manner in which they were regularized, though it

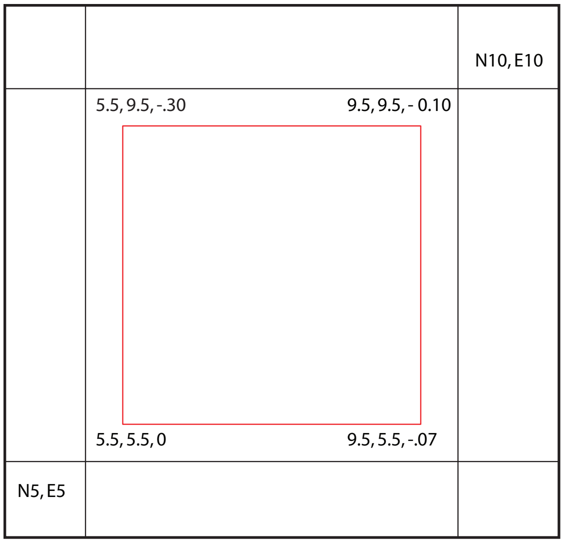


Figure 29

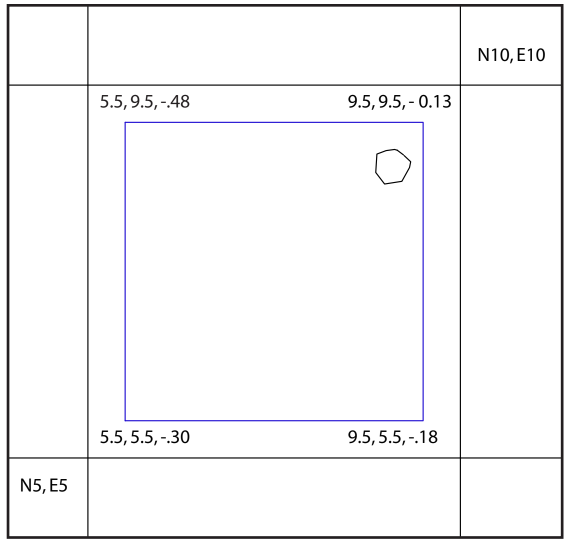


Figure 30

can be very difficult to deal with excavated lenses and other such complex shapes. The points actually surveyed should be clear. Objects can be represented by icons in the surveyed findspots – and the important record can be complete and well-preserved. Details should be included whenever their presence/location/size is an issue. The advantage of having three-dimensional representations of the trenches (and building parts) is worth the added effort. Photographs and documentation can provide otherwise missing information.

The illustrations in figures 29 through 43 show how an excavation might be documented. (Since they are discussed thoroughly in the text, there are only figure numbers for captions.) Figure 29 shows the basic grid for our hypothetical excavation. The corners of the grid have been labeled, as have the corners of the first excavation unit, the plow zone (all with x, y, and z relative to the site datum). The outline of the trench has been color-coded red. This is the base-line drawing since it shows the excavation area before digging has commenced. Choices for color, text size and such basics as the presence or absence of the grid lines are entirely those of the draftsman in consultation with the person for whom the drawing is prepared. The model entities lie on layers that permit the draftsman to pick and choose the information to be shown in each individual drawing.

Figure 30 shows the bottom of the plow zone. Once again the coordinates of the corners of the excavation unit are shown so that comparative elevations may be seen. In addition, the irregular outline in the NW corner is the outline of a pit that was identified only at the base of the plow zone. No coordinates for its outline are shown, but the coordinates for surveyed points could be included in the drawing if desired. The color for the plow zone is blue.

Figure 31 shows the same excavation unit, the plow zone, but from the southwest in a 3D view, actually two 3D views, one with the trench treated as a box, with all sides exposed and the other as it might be seen in the real world, with

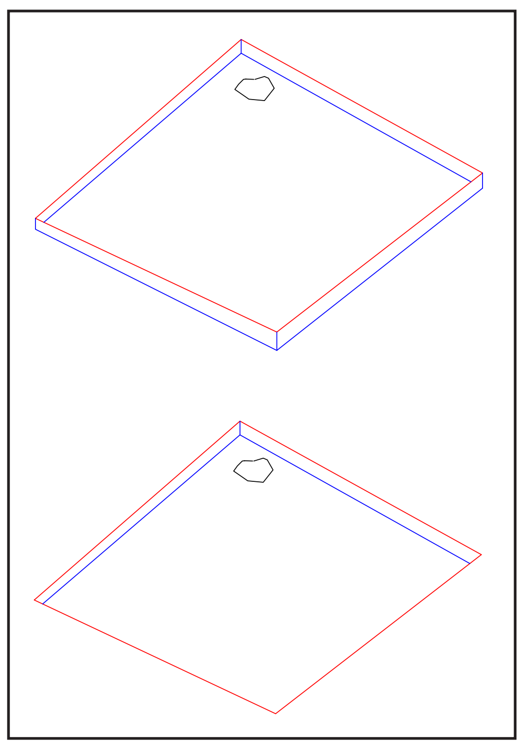


Figure 31

some outside surfaces hidden by the ground. There is no text in these drawings, but text could easily be added as desired.

Figure 32 shows the next excavation unit, the first level below the plow zone. It is color-coded light brown. This time the coordinates of the corners have been omitted, and only the grid coordinates are shown. However, a fire ring is shown, with each block outlined. In addition, the outline of the intrusive pit – at the floor level not the outline where the pit first appeared – is shown, as is an icon for an object that was found in the pit. The shape of the icon (and its layer name) indicate the type of artifact. The position is conventional, with the upper left corner marking the actual location as recorded by survey. (The point located by survey would be shown by an arrow included in the photograph of the object *in situ,* and the photograph and its data would be connected via information recorded on the appropriate layer.)

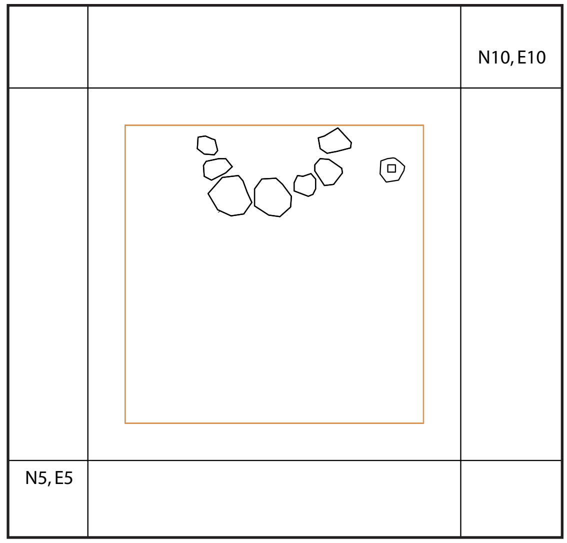


Figure 32

Figure 33 shows the same excavation unit with the rocks of the fire ring faded to gray and markers for survey points to define the bottom of the excavation unit shown in black. The outline of the pit and the object icon have been omitted in this drawing. Since this is a plan view, there is no indication that the floor reflects the more complex shape defined not only by the corners but by the points surveyed on the floor as well.

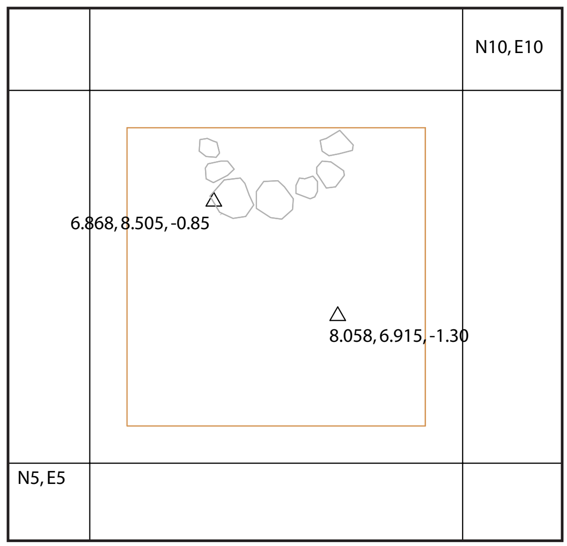


Figure 33

Figure 34 shows this excavation unit in 3D. There are multiple versions of this drawing, one showing the plow zone and the unit below as if they composed a box and two in a more traditional view with the hidden sides of the trench invisible. The uppermost view shows the fire ring, the bottom of the intrusive pit, and the location of the artifact (and, by the shape of the icon, the nature of the artifact) found in the pit. Since the artifact was found mid-way down the pit; it seems to float in 3D space.

The middle view omits the pit and artifact icon, and the lower view shows the pit as a 3D shape, without the plow zone – and without the artifact icon that is hidden by the surfaces that make up the sides of the pit.

Note that the stones of the fire ring are shown as extruded right, rectangular prisms. The plan-view shape is accurate, but there has been no attempt to record properly the various faces of the blocks. To do so would require more time and expense than justified. (In addition, the 3D views originally showed the blocks without lines at the bottoms of their vertical faces. Those lines should be there to mark the edges of the vertical faces where they met the ground. Since AutoCAD does not produce a line to show where one surface is hidden by another, the missing lines had to be added manually and only as drawn, not measured, elements.)

The stones of the fire ring illustrate one of those needs to determine what is important and what is not. The tops of the blocks have been surveyed; so all are at the correct elevation, but, of course, each survey point located a single point on the top of a block, not a flat surface. The shapes, in plan view, are accurate. The 3D representations are less appealing and less accurate. Nevertheless, the ability to obtain a 3D view and to understand better the complex geometric relations of the stones and the pit are enhanced by the 3D elements. Similarly, the pit has been represented in ways that clearly generalize the actual shape, but the result is useful.

The excavation unit below the one shown in figure 34 has fill only, but beneath it is the unit shown in figures 35 and 36. This is an improbable unit to say the least since it contains both a rough-stone wall and a mudbrick wall, but it allows better illustrations of difficult forms. Figure 35 omits the grid for the first time in a plan view, making it possible for the drawing to be a bit larger. It also shows the plan view without the complications of the 3D surfaces that combine to approximate the face of the mudbrick wall. The contours of the wall at each end have been surveyed and used in combination with the surveyed paths along each side of the wall to make the wall's 3D surfaces, and those surfaces are shown in figure 36, as are the grid and text noting the coordinates of the grid squares.

Figure 37 is a rendering created within AutoCAD and then cleaned up with PhotoShop; it shows the mudbrick wall as it should be seen (and from a different angle – from the northwest). In this case the sides of the upper excavation units are not shown so that more of the wall can be seen.

Figure 38 shows two different 3D views of this last excavation unit. The upper view shows the unit with baulks of the units above. As a result, less is visible. The lower view shows the excavation unit without the baulks of the upper units so that more can be seen. In addition, that view shows the unit as a wire-frame drawing

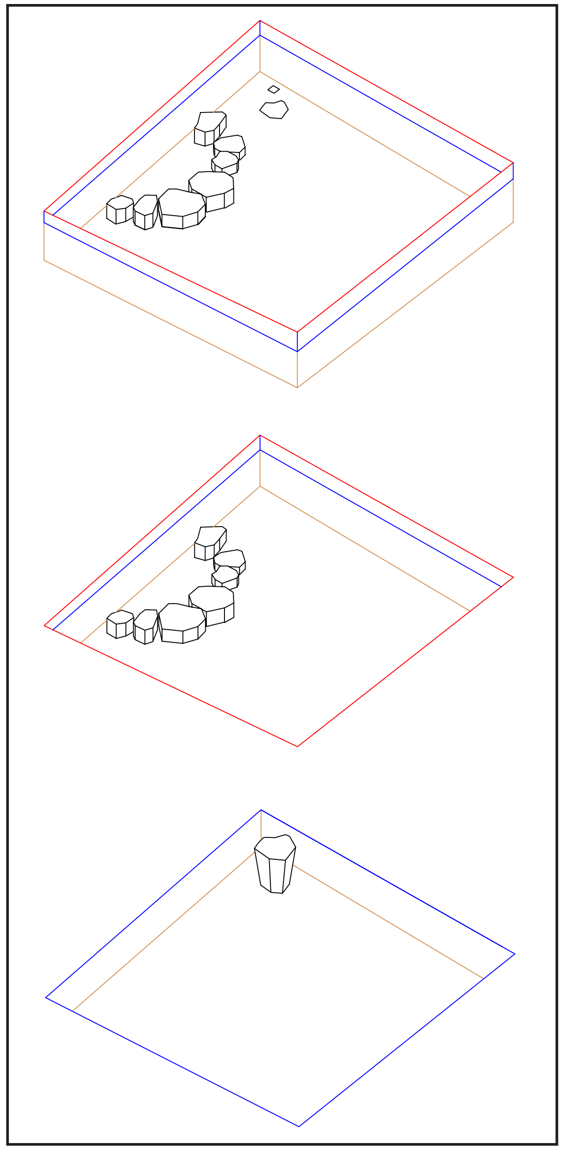


Figure 34

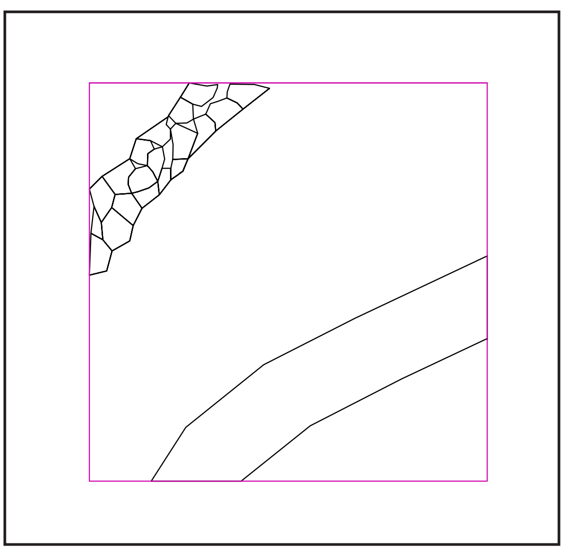


Figure 35

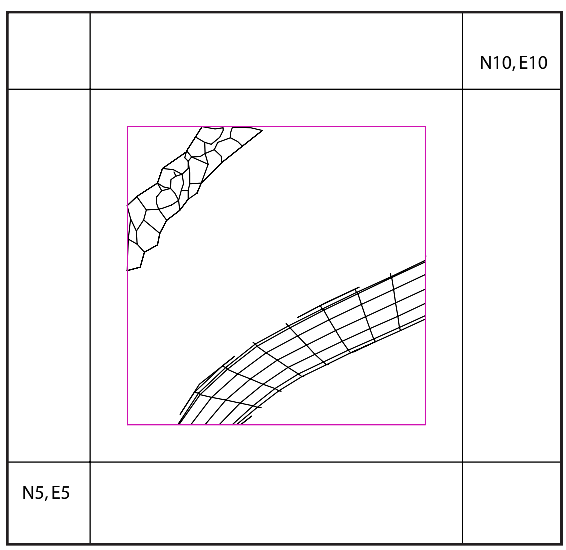


Figure 36

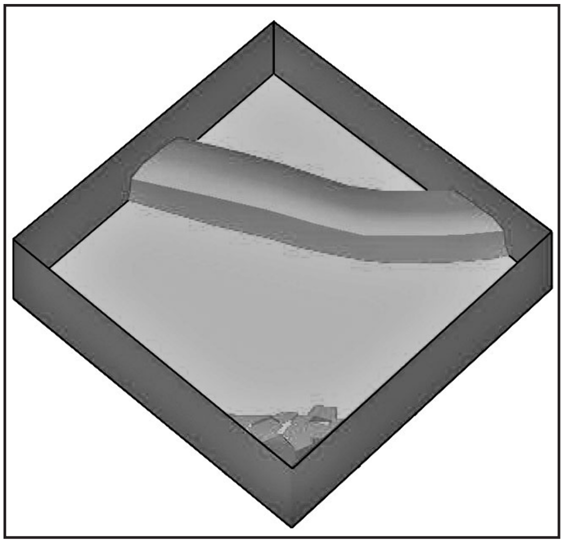


Figure 37

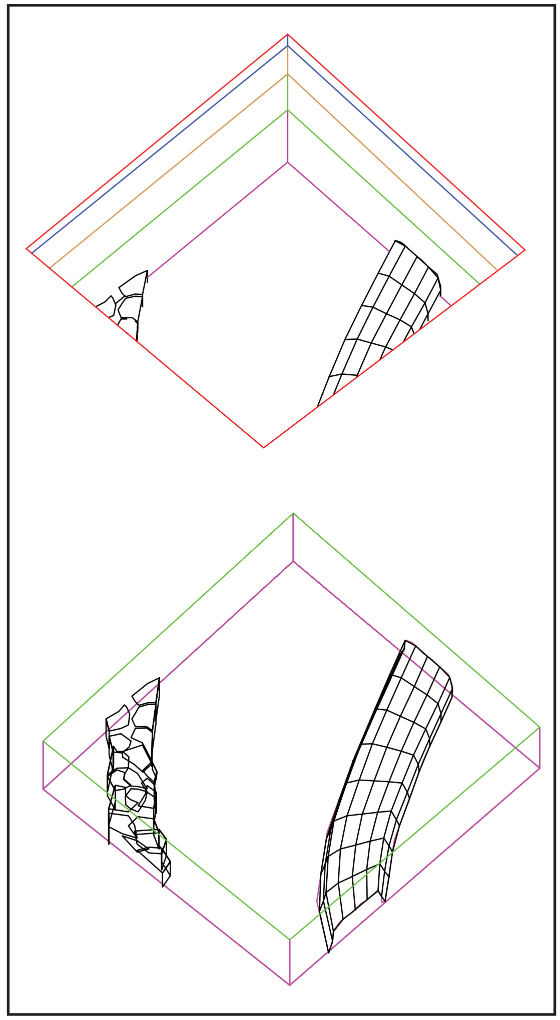


Figure 38

rather than a hidden-line view. As a result, even more of the excavation unit can be seen.

The mud-brick wall in this excavation unit fares well in the rendering, less so in the 3D views that show the edges of the various surfaces created to indicate the shape of the wall. What is not apparent from these drawings is the difficulty of handling the stone wall in 3D. The top of the wall is faithfully drawn (see figure 35); indeed, the drawing is virtually identical in terms of techniques used for the hand-drawn Pseira wall in figure 1. There is one critical difference: each block is drawn at a different elevation, the elevation of that point on the block for which an elevation was noted.

The face of the wall can also be drawn very well, again much like the Pseira walls in figure 2. Such an elevation view of the wall is shown in figure 39. In this case, the blocks' faces are all vertical, but they have been positioned in line with the edges of the topmost surviving blocks. That makes the model of the faces roughly like that of the tops of the blocks. Because the tops of the blocks have been modeled as horizontal and the faces as vertical, neither can particularly accurate in a 3D view. Nor do they work effectively together in a 3D view.

Figure 40 shows both the wall top and the face of the wall in one 3D view. At small size, the drawing looks better; so it is produced at somewhat larger size here so that warts and all are visible. As is apparent, the blocks of the wall top seem to have gaps between them because they are at different elevations, making

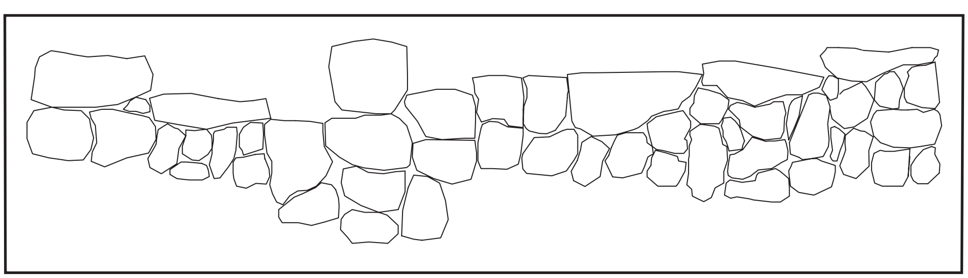


Figure 39

gaps show in this axonometric view. In addition, the edges of the top and face surfaces do not fully match, as they cannot with these techniques. The resulting gap at the joint between top and face is unsatisfactory. As a result, it may be better to treat any rough stone wall as we did the fire ring above – as right rectangular prisms – as shown in figure 41 for this wall. Doing so permits the wall top to be accurately drawn (as in figure 35) but does not pretend to treat the whole wall as a 3D entity. The drawing of the wall face can be made and retained in its own layer (figure 39), but the 3D views could use the simplified version of reality shown in figure 41. This approach permits both wall top and wall face to be included and also permits a 3D view to be created when required. Whichever approach is chosen, as always, the technology must serve the scholar and the project, not the other way around. There are no hard-and-fast rules here. (Both versions could also be produced and stored on separate layers.)

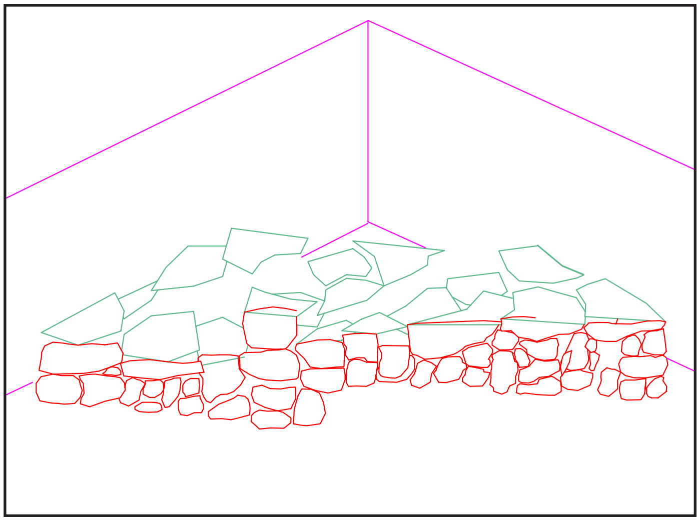


Figure 40

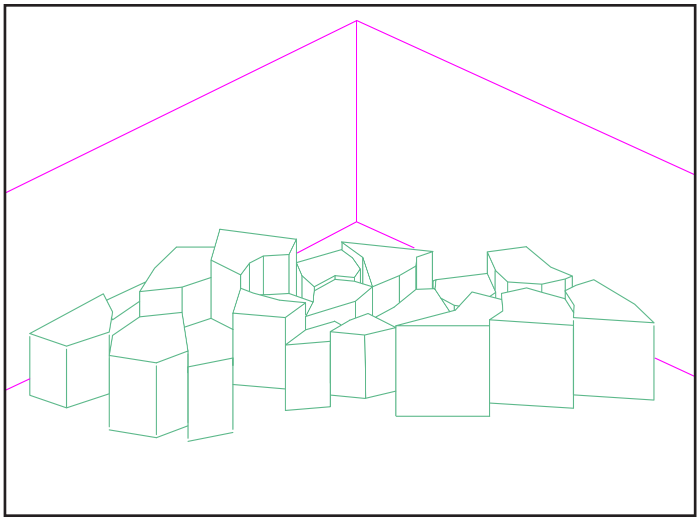


Figure 41

The baulks for these excavation units can be modeled as a part of the modeling process with the kind of attention to detail that is the norm for such drawings. For these units I have made drawings of the east and north baulks. They are shown in figures 42 and 43. Figure 42 shows the north baulk, and figure 43 the east one. Both were drawn from specific point locations along the floor of each excavation unit at the baulk; there were more such points used for the baulk drawings than for defining the surfaces making up the floors of the excavation units. The difference between the number of points used for the floors and for the

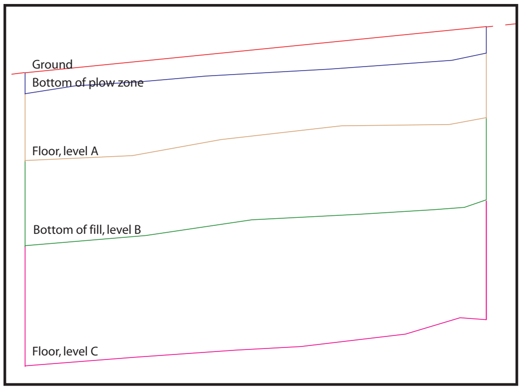


Figure 42

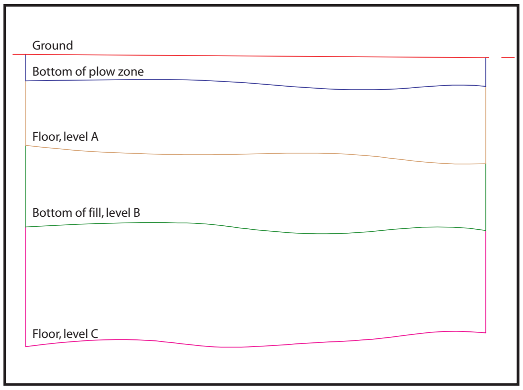


Figure 43

baulks reflects one of those choices regarding tolerances and approximations. All the points could have been used to model the floors, but they would have added little of value. The impact of those points on the baulks, though, is more important to our understanding. So more points were taken for the baulks, and all those points (any number of such points) could be used in the drawings.

The two drawings of the baulks differ in that the points marking the floor levels on the east baulk (figure 42) were made with a simple polyline that was not smoothed. Those of the east baulk were connected using a spline instead. The lines in the drawing of the north baulk seem more realistic by virtue of being smooth and showing less abrupt changes of direction. The coordinates of the surveyed points could be obtained from either model, but the locations of those points are more clear in the version made with a less realistic polyline.

More on Data Density

This general question of data density deserves a bit more. It is tempting to consider a CAD model – or, for that matter, a paper drawing – as a simple expression of fact, a positivist document and to obsess over the need to compromise when modeling in 3D. But neither the CAD model nor the paper drawing can ever be free of interpretation. At the simplest level, the choice of points to survey is an interpretive act. Therefore, to the extent possible, the documentation accompanying a CAD model should include explicit discussions of all issues involving survey choices as well as those concerning the nature of the CAD model. Such documentation serves two purposes. It provides information for users of the model, and it forces the scholars in charge to confront the questions of data density and approximation head-on.

Data input procedures must also be documented to assist with the matter of precision. The different ways to input data – using absolute coordinates, tracing, picking points from a grid, and so on – make possible vastly different levels of precision within a single model. Users must know what precision to expect from different parts of a model.

Layers Again – Still More Complexity

The importance of CAD layers in archaeology has been stated before, but a complex 3D model provides some additional complications – and additional possibilities. For instance, layer names can be constructed to permit keeping plans separate from 3D layers, as was done with the hypothetical excavation model just examined. Of course that requires adding to the layer naming system. In a more complex model it may also be necessary to distinguish material by phase, date, building material, or even the scholar responsible for a given reconstruction. As the information in the layer names becomes richer, the names can get very long – the system for the model of the older propylon in Athens now has 18 characters.[5] However, the possibilities for searching the model grow commensurately. Fully 3D models of portions reconstructed by a specific scholar, made of a

---

5 The layer names consist of one character to indicate the kind of layer (2D plan, 3D model, text, etc.), one for "in-situ-ness" (or, when appropriate, that the material is hypothetical, in which case the scholar responsible is indicated), two for the area in question (a general indicator such as military, religious, public and a more particular one for fortification, entry, courtyard), two for architectural purpose (again a general indicator for wall, stair, column and a more particular one for block, step, drum), two characters for material (a general indicator for rough stone, cut stone, bedrock and a more particular one for limestone, Pentelic marble, poros), and ten characters for date (five for beginning date and 5 for ending date). Thus: the name mirewbbp-0489-0478 indicates a modeled layer with undisturbed *in situ* material in a religious entry area, containing wall block(s) of cut-stone Pentelic marble, erected in 489 B.C.E. and removed in 478 B.C.E., while cipcwvbp-0489-0437 indicates cracks (by implication modeled material) on cut-stone Pentelic marble blocks, *in situ*, that serve as a wall veneer in a public courtyard, erected in 489 B.C.E. and removed in 437 B.C.E..

specified material, and with a certain beginning and ending date can be located and displayed.

The names of the layers seem at first glance to be ridiculously long, but the system is logical and relatively easy to use because it depends on clear-cut categories, not acronyms or hard-to-remember individual layer names. For instance, the first letter in this system identifies the layer as being one with plan information only, 3D entities (exclusively surfaces in this model), cracks (3D data implicitly but considered to have become visible only after the original installation of the blocks), holes (also 3D), or labels. The next letter identifies the layer as one of a number of possible levels of "*in-situ*-ness." An object may, at the simplest level, be *in situ* without qualification, but it may also be *in situ* but in a position of secondary use. Objects might also be *in situ* but not precisely so, having been moved by natural or man-made causes (two separate categories). Of course, objects might also be completely hypothetical. To make matters more complex, those hypothetical objects might by ones about which scholars who have studied the area agree, but each might also be associated with a particular scholar. Note that these distinctions are often conceptual, having little or nothing to do with position or other physical distinctions. Note also that, as with databases, CAD models can and should honor scholarly differences of opinion by permitting multiple, competing reconstructions to exist in the same model, each identified with the scholar responsible for it.

Not all of the remaining characters need to be discussed here, but the last ten do. They provide two important dates – the beginning date for the objects on the layer in question (five characters) and the ending date for those objects (also five characters). In this model, the earliest date is roughly 1200 B.C.E.; so five characters were allowed for both beginning and ending dates – four numbers plus a minus sign to indicate B.C.E. Thus, a layer ending with "-0489-0478" should contain objects constructed in 489 B.C.E. and removed in 478 B.C.E.

In this model the dates provide an interesting illustration of the limits of AutoCAD's naming system. I chose to name the layers as if I could use search requests that would find layers with dates greater or less than a specified number (hence the use of the minus sign for dates B.C.E.). But AutoCAD's search system will not do that yet. As the system actually exists, I could just as easily have used phase numbers. That is, I used a generic date for the late Bronze Age material, a well-accepted historic date for other material, and dates for the three remaining phases that were estimates. All the dates indicate a level of precision for our knowledge – accuracy to the year – that is not possible, but they provide a scheme that I can use well to segment the model. (To learn more about the details of a fully-functioning layer-naming system – and to see how excavation information may be added to the system – see "CSA Layer Naming Convention," by Harrison Eiteljorg, II, in at csanet.org/inftech/csalnc.html.)

3D Drawings

Paper drawings in 3D present some different problems for CAD. In short, it is more difficult to make 3D drawings look hand-drawn, to have an artistic flair. That should have been apparent from some of the drawings here. That is, in part, because complex objects must be somewhat simplified to be modeled in 3D. As a result, there can be too many simple, straight lines that have a mechanical look; much depends upon the scale of a drawing, as the comparison between figure 20 and figure 24 showed. More artistic effects can be obtained by using Illustrator or a similar drawing program to modify a CAD product. (Renderings are also possible, but their use is beyond the scope of this book.)

Given the added difficulties of 3D – added survey work, complex questions of data density, and the problems that can arise when producing 3D drawings – it is fair to ask when one needs to work in 3D. Consider the example of the Pseira rooms in figures 1 and 2 above. As the drawings show, the wall tops are very irregular. So are the wall faces, and, in fact, even the ground and floors are far from regular. Modeling such irregular surfaces is more than difficult; it requires much more data, more time in the modeling process, and, most important, considerable simplification of reality, as discussed and as shown in our excavation example.

For a 2D model, the absence of the third dimension makes it clear that the real world has been simplified. In addition, much of what is modeled in a 2D setting – as in the case of the Pseira tombs – comes into the model via a process more akin to drawing than drafting. For example, tracing hand drawings on a digitizer or tracing a scan of such drawings on screen are common ways in bringing data into a model. A 3D model, on the other hand, requires 3D data points, not drawing in any literal sense. The model depends upon surveyed points with 3 coordinates: x, y, and z values. The simplification, as discussed above, must be self-consciously done, but it must also make it possible to locate enough individual points to produce a model that is both adequate for scholars and adequate for constructing a reasonable 3D model. Too few data points will yield a model that simply cannot produce a good model. Too many will make extra work for everyone.

Does that mean 3D will only work for structures and more regular material? The answer is both "yes" and "no." It may mean that something like the walls at Sacsayhuaman should be treated as both a 2D and a 3D object, with plans showing the path of the wall well but with only occasional changes in the wall heights so that the 3D model is closer to what architects call a "massing model" – a model intended to give a sense of the overall mass of the structure rather than to show its details. Combining a good plan with photographs and such a 3D model might be an ideal solution. Using a 3D scanner on such a subject might seem a better choice; a far better approximation of the shapes of the blocks can be obtained – but at a significant cost in time as well as money.

What about an excavation? Can one really make a 3D model of an entire site? The answer this time is a qualified "yes." There will be compromises, as we have seen in the example, but the result is nonetheless valuable. The three-dimensional complexity of an archaeological site is such that a 3D model can be extremely helpful to all. In such a case, the more carefully planned the work processes, the more valuable the results.

Surveying

Surveying, especially when working in 3D, is both a practical and a theoretical issue. As a practical matter, the surveying procedures present a host of problems when working in 3D – most having to do with finding ways to survey points that are difficult to reach. In a 3D environment, as noted above, the number of points required escalates dramatically; unfortunately, those new points to be surveyed are in harder-to-reach places.

Traditional stereo photogrammetry can come to the aid of the scholar in some cases, saving the survey team from needing to reach survey points. However, the material under study must be appropriate for the use of photogrammetry, meaning that the points to be surveyed must be unambiguous and clearly identifiable in photographs. A fieldstone building façade with mortar joints is a good subject for photogrammetry because the transition from mortar to stone is clear, and the relatively irregular stones will provide unambiguous points in photographs. On the other hand, worn blocks without mortar at the outer surface may make photo interpretation very difficult. If the corners are not sharp and the mortar-to-stone transition points are not clear, finding unambiguous points can be very difficult.

In close-range photogrammetry, where multiple photos are used, the problem of photo interpretation is greater, because it is necessary to find the same point on multiple photos. Stereo photogrammetry, on the other hand, starts with a stereoscopic view and only a single point to identify (though that point is actually identified separately on each of two photographs).

New 3D scanners also promise survey information without needing to reach the survey points. At the moment, however, such scanners are extremely expensive either to purchase or to hire for short-term use. The analysis of the resulting data is also rather demanding, requiring both experience and training. In addition, the precision achievable is somewhat lower, with point locations specified within half a cm. but probably not closer. That may be the case today, but this new technology is developing rapidly; precision and general utility can be expected to advance.

Surveying with a total station – electronic theodolite and coupled distance measuring device – is both the most precise system (though stereo photogrammetry may match the precision if the photos are taken from rather close to the survey targets) and the one that provides the most certain point identification. Such instruments work very well and are probably the least expensive alternate in the long run. One person positions a target while another operates the instrument. This means that a team member is making a careful and conscious choice about the point to be surveyed. That person is obliged to examine closely the area in question and then to choose the points to be surveyed. This is ideal – if and only if the points to be surveyed can be reached with relative ease and security. Modern total stations no longer require the use of a target; they can take a direct reading from nearly any surface. This can reduce the surveying "crew" to a single person. Note the trade-off, however, when using a total station without a target. The operator is obliged to pick points to be surveyed by looking through the instrument telescope from some distance. Very fine discriminations simply cannot be made from ten or twenty meters away, even with the aid of a good telescope.

Total stations that do not require reflective targets will not provide good results when points to be surveyed are ambiguous, as when photogrammetry is more difficult to use. For instance, a scholar would want to survey carefully-shaped masonry blocks by locating the corners of the blocks. The corners of blocks in place for centuries, however, will have been eroded and therefore not available for direct survey. A reflective target at the proper point will be required – at the cost of requiring someone to climb, stand, sit, or lie nearby while holding a target.[6]

Omitted in the foregoing was any reference to measuring with tapes, line levels, and the like. The complex equipment and procedures described are not always necessary, but it should be obvious that measuring with tape measures, line levels, plumb bobs, carpenters' squares, and similar tools is problematic in a 3D environment. One must either make a great many assumptions about things that seem to be level or vertical or at right angles – or attempt to make a good many very fine measurements with tools that are not really meant for such precision. Anything larger than a few meters square should be surveyed with more modern equipment if 3D data points are desired.

The survey equipment and processes are critical for obtaining good survey information, and they require some training and experience. Equally important, however, is the philosophical question that is necessarily involved when we begin to survey for 3D models. Again it is necessary to emphasize that the difficult question is the simplest one. How many data points are required – and where should they be?

Linked Database Information

Most modern CAD programs include some mechanism for linking database tables to the model. Some programs permit external databases to be linked to the model, while others supply their own internal database management. The advantages are obvious. An icon representing an object could be connected to the data

[6] The reflective target can, with some total stations, be as simple as the kind of reflective tape used for bicycles and trucks to reflect automobile headlights, although some total stations cannot read reflections from such tape. With a target as small and light as tape, there are circumstances that permit sighting with the total station before positioning the target and then putting tape in the necessary position with some aid (I have even used a long fishing pole) so that nobody actually needs to be near the survey point. Such targets must be clearly identifiable from afar.

table about that object; the walls of a trench could be connected to the database information about the trench. The examples could go on and on. Nevertheless, it is problematic to use attached databases, and the practice is most emphatically not recommended.

The reason for that lies in the scholar's need for data that will survive not simply years into the future but decades. For that survival it is far better to separate the CAD model from the data so that each form of data may be treated separately. Further discussion of this important topic will follow in the chapter on archival preservation of data.

If data linkage supplied by CAD programs is not to be used, that does not mean there can be no links. It does mean that the links must be created in ways that do not interfere with the transfer of CAD data through various file formats over time – and the similar transfer of database tables through various file formats.

The best way to connect a CAD model to external data is to use a standard DBMS program for holding the data and to put into the CAD model (on appropriately named layers – the layer-naming system is again critical, of course) explicit indicators of links to the database (foreign keys). The indicator can be an icon pointing to the object(s) in question, but it may also be so well-defined by the layer name that its subject is clear. The icon must also include a number, letter, or term that is the explicit link to the data table, the foreign key. This system of explicit links between icons and data is not a fully-defined and articulated one. Users can make many individual choices, and the system can even accommodate lengthy notes and more cryptic database entries simply by using different icons or by using a letter-number combination to indicate the nature of the connected data and the identification number.

If the data attached to the model are relatively simple or if there are few data items, a simple text file could be used rather than a database file. (For a longer discussion of systems to link data to CAD models, see Harrison Eiteljorg, II, "Linking Data to CAD Models," *CSA Newsletter*, Vol. XIV, no. 3; Winter, 2002 at http://www.csanet.org/newsletter/winter02/nlw0201.html.)

Blocks

CAD programs often make it possible to design and insert stock entities into a model so that one need model the same item over and over again. Inserting them in specific places and at specific scales and rotations in a model is much simpler than drawing them again and again. In AutoCAD, these pre-designed entities are called blocks, and they can be very helpful. They can also cause significant problems because they behave in unexpected ways. Specifically, a block designed on any layer save layer 0 (zero) will only be displayed if that layer is displayed – as well as the layer on which the block was actually inserted. The inserted block will also have the color and line type associated with the layer on which it was designed rather than the layer on which it was inserted, again unless it was designed on layer 0. (A word processing file can be used to store text that will model a standard entity and put it on any layer without the problems associated with AutoCAD blocks.)

Archaeologists are unlikely to need to use blocks in modeling the real world, but they may be used as icons or as common parts of reconstructions, e.g., individual roof tiles. Because of their seemingly unpredictable behavior, users should be wary of them.

Cross-Referenced Files

When CAD files become too large, they become hard to open and to use. Everything slows to a glacial pace, and work becomes progressively more difficult because of the pace. The simplest solution to that problem is to keep as many layers out of the current selection set as possible. At least with some programs unnecessary layers are not brought into the computer's memory and therefore do not slow the machine. When that work-around fails, it is possible to use multiple files to hold various parts of the whole, much as one might use layers. When doing that, one file (file A) may include a reference to another (file B) so that opening the file A also calls up the external file B. Of course, when the set of cross-referenced files is large, the system still works very slowly. Putting the same number of entities into multiple files does not make the computer work faster.

The advantage of using cross-referenced files comes from the ability to work on the portion of the model in each cross-referenced file alone. That is, file B can be opened and modified alone without file A – and the other files related to it – or while file A is being modified by someone else. In a large project, therefore, multiple CAD technicians may work on different portions of the whole at the same time, while still permitting all to be viewed together when desired.

When cross-referenced files become a necessity – and they should only be used when truly necessary – significant care and planning are required. First, the cross-referencing should never permit a file that is referenced by another to itself reference a third file. That is, the referencing should be one-file-deep only. Second, the basic file (file A) to which others are linked, should be empty, serving as the empty vessel into which the linked others are placed. The base file should contain only the 0 (zero) layer, a layer with a scale, a layer with a north arrow, and layers named *...refs* (or a suitable name based upon the layer-naming system) for the external files. Since each external file is attached to a specific layer, each *...refs* layer will include one and only one external file; the layer name should help to identify the related file. The base file should contain no entities save those required for the scale and north arrow.

Treating the base file in this way – as the empty vessel meant simply to hold external files – permits the CAD technician to use a number of different base files, each including a specific sub-set of the whole project and each answering the needs of specific scholars or research questions. (In this way any individual file may be used in multiple sub-sets of the whole.) At the same time, each referenced file can be opened independently for editing.

Documenting the Model

A CAD model, like a database, must be documented if it is to be used effectively by others. The following is a list of those issues that should be included in the documentation for CAD files. It should go without saying that the documentation must begin with a description of the aims and general procedures of the CAD work on the project.

The names of all CAD files and for each file:

1. Software used – including a thorough history of the versions used and any upgrade procedures or problems.

2. File formats – both the final and all previous formats as well as a history of the processes used to migrate from one format to another.

3. Cross-referenced files. This may need considerable explanation to make certain a user understands what each base file contains and what each external file contains.

4. Layer-naming system – with a full history of its development and a complete list of the layers used, including full bibliographic references to all sources of restorations included in the model.

5. System of linkage to external database or text files, if any – and the database files must be separately documented, of course, according to the requirements outlined in Chapter III.

6. Alternate coordinate systems, if any, with their purposes.

7. Blocks used and information about them.

8. Stored views, if any, and their uses.

9. Macros or scripts, if any, and their functions.

10. Significance of colors or line types, if any (a practice discouraged here).

11. Measurement units employed.

12. Data gathering methods and precision, probably including a lengthy discussion of both practical fieldwork issues and the impact of those matters on achievable precision. This should include a thorough discussion of modeling procedures, especially those used for 3D entities.

13. Review processes, if any.

14. Instructions or other text documentation.

15. Personnel involved, including as much detail as possible about who was responsible for which portions of the model.

16. Dates of CAD work, especially the work that takes place at times when fieldwork is not active.

CAD with GIS

GIS programs often have rather primitive mechanisms for drawing. As a result, it is common to use CAD programs to create maps for vector-based GIS systems. When doing so, as noted in the GIS chapter, it is necessary to exercise care so that the resulting drawings are appropriate for use in a GIS system.

In general, a 2D CAD program will suffice for producing GIS maps. Since the third dimension is normally a data attribute in GIS data sets, 3D CAD is not required. This use of CAD is driven by the needs of the GIS system not by CAD requirements; so anyone using CAD in such a setting should be directed by those needs and need not be so concerned about proper CAD practices except insofar as they aid in the production of good GIS data.

Conclusion

CAD is one of the core technologies for archaeologists because it can be used so well to document archaeological materials. Using CAD programs may require some skills not so necessary for those using databases. Being able to visualize in 3D, for instance, can be very important for someone building a complex CAD model – though some would argue that similar visualization skills help with the organizing of data tables.

CAD also requires a good understanding of survey techniques – a topic not thoroughly discussed here – and of the limits imposed by the techniques chosen. Otherwise, the limits on accuracy and precision cannot be well understood and communicated to other users of the model.

Using CAD programs wisely, though, requires an approach that is equally important for users of database systems and GIS programs. For all these technologies – in part because all are still quite new to archaeologists – a self-conscious approach is required. One must not only work to build the model or the database or the GIS data set; one must also think about the ways the work is being carried out, the ways other people will use the resulting data, and the ways to prevent misuse. Perhaps the situation will change in twenty years, but, for now at least, care is not simply avoiding mistakes, it is thinking carefully about all the planning steps and never granting the "accepted wisdom" too much status.

Selected Further Resources:

There are few good and up-to-date resources for the use of CAD in archaeology. The Archaeology Data Service's *CAD: A Guide to Good Practice* (see ads.ahds.ac.uk/project/goodguides/cad) is one, but the principle author was yours truly (with significant contributions by Kate Fernie, Jeremy Huggett, Damian Robinson, and Bernard Thomason); so the difference between that resource and this is not great. CSA maintains a resource similar to that ADS publication at www.csanet.org/inftech/cadgd/cadgd.html.

English Heritage has published a valuable aid, especially for 2D recording: *The Presentation of Historic Building Survey in CAD* (undated). Readers should be forewarned, however, that the layer-naming system employed is ineffective. The CSA web site, csanet.org, has many other helpful items, including a very basic tutorial for learning to use AutoCAD, and the *CSA Newsletter* (csanet.org/newsletter) has many articles related to the use of CAD in archaeology.

For those readers with access to a computer running AutoCAD or some other CAD program, experimentation with the program, especially if the user really understands the geometry of the subject of the experiment, should be helpful after the basics have been mastered. A clear, full, three-dimensional understanding of the geometry of the subject of any experiment is critical. Otherwise, the user may make a great many simplifications without even realizing it. With a good understanding of the geometry, however, it is possible to appreciate the problems of modeling the geometry, even if the actual processes are beyond one’s level of competence or experience.

VI

Miscellany: Digital Images, Audio Recordings, Videos, and Text

Introduction

The subjects of this chapter are varied, to say the least. They are other forms of digital data gathered in the course of archaeological work, and nearly all have one thing in common: they are digital fruits of technologies that have such good non-digital analogs that the actual creation of the material is not the issue under discussion here. Instead, we will concern ourselves with the ways the digital versions of these efforts must be treated to be useful and to serve their real scholarly functions. In addition, we will discuss some of the practical considerations that are involved with digital versions of materials we are all accustomed to using in older forms.

There is also some discussion of vocabulary control and the related issue of the data tagging system called XML.

Digital Images

Photographic images

Digital images are not inherently different from silver-based photographic images in terms of techniques used or the utility of the results. There are differences, to be sure, but they are not critical to the value of the images. Nevertheless, there are special problems involved in the use of digital images.

One of those special problems is the ease of adjusting digital images. They can be lightened or darkened, made to have more or less contrast, turned from color to grayscale, from positive to negative, cropped, and altered in any number of additional ways. The temptation to make such changes – to improve the image – is irresistible. It is not necessary to resist the temptation, but it is necessary to save the photo as taken, before the adjustments, so that anyone can return to that original image to see what the camera saw, what the image was like before it was "improved." The original image should be considered an archival one (assuming the image is to be retained at all), even if a modified version is the one to be used and is also archived.

Consider this comparison of two photographs

Figure 1
Above, the necropolis in Orvieto excavated in the nineteenth century and below the portion excavated in the twentieth century.

of the necropolis of Crocefisso del Tufo in Orvieto, Italy. The image above is of the nineteenth-century excavation; below is part of the twentieth-century excavation. The archaeologists working in the nineteenth century thought they knew what the cemetery looked like; so they re-arranged the finds to "put it right." Archaeologists of the modern era do not make such presumptions. A corrected photograph is like the nineteenth-century archaeologists' final version of the cemetery – prettier than reality and therefore potentially misleading.

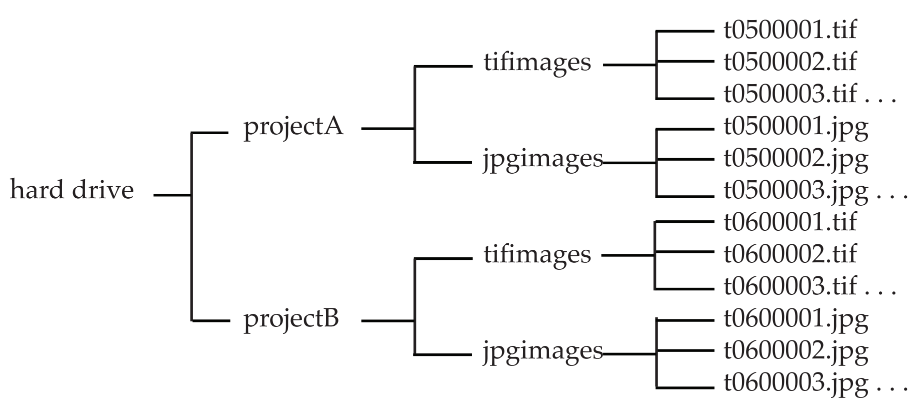
Another special problem of digital imagery is its price. Digital images are virtually free. After the equipment has been purchased, the cost of any individual photograph is essentially nil. Although it may seem counter-intuitive, this is not necessarily a good thing. Since each new photograph has virtually no cost, there is no real impediment to taking photographs, more and more photographs, and the number of images taken for a project can rise to an unmanageable level. Therefore, it is critical that project personnel cull the images on a regular basis so that duplicates, poor-quality images, and images that offer no useful information can be deleted before they overwhelm the project's storage capacity. The remainder of the images must be thoroughly catalogued to be useful, and spending that most precious of resources, time, on the process of cataloguing useless photographs is a terrible waste.

A small, final issue brought into the discussion by the transition to digital imagery is that of color. Archaeologists became accustomed to using only black-and-white images for everything but slides for presentations because of the cost of color publication. Digital imagery on the web, however, can be in color as easily – and as inexpensively – as in black-and-white. In addition, digital cameras can be used to take images in color, translating to black-and-white (gray-scale more properly in a digital environment) when needed. As a result, color images are now the norm; black-and-white or gray-scale images are vanishingly rare. Unfortunately, having used color little over the preceding decades, archaeologists got out of the habit of including color charts in color images of objects. Such charts provide a sure and simple method for identifying color accurately; they should be used.

Cataloguing photographs was referenced briefly in the database chapter, but there was virtually no discussion of the kind of information that is required. For each photograph there should be the following data items: an unambiguous id, date, subject, camera and lens (and f/stop and zoom setting when appropriate and known), camera position and orientation (for site photos), photographer, original file format. Object identifiers or lot numbers should be recorded as well. In the case of site photographs – as opposed to object photographs – there should be added data based on the terms and organization used in the project database. Indeed, the official site photographs should be catalogued as part of the project database. When people are in photographs, their names should be recorded (once in the database tables, a foreign key should lead to the names).

The more difficult problem is to determine how to store the images and how to manage the storage locations and information about them in the project database. It goes without saying that all photographs should be stored on CDs or DVDs for the long term, and the files are likely to be so large and so numerous that CDs and DVDs will be required in large numbers. The CDs and DVDs, though, should not be the media for regular use. The regularly accessed image files should be on the a hard drive with the project database for quick access – if that is possible, given the volume of material. Whether on removable media or on a local hard drive, though, specifying the storage location of the images is not a trivial problem. Many CDs or DVDs will be required, and some attempt to organize images on a hard drive is also required. Therefore, the storage locations should be organized in a way that assists users, and the data table(s) for images should include information about storage locations. The image name/id should be a column in the database row for the image, of course, whether that name is a simple number or an attempt at something more meaningful, but the name alone will not locate the file. Wherever the image is stored, that location must be stated in the database explicitly or by formula, and the manner of determining the location must be clear and consistently enforced. (I would recommend that each image file name begin with a key letter, no matter what else is in the name, with the key letter indicating the kind of photograph – object, ceramic lot, lithics lot, locus, trench, personnel, etc.) The actual storage location can be stated in the same row of the image data table, placed in a linked table, or determined by some calculation. Whatever the storage location, the name should ideally suffice for both the local computer location and the CD/DVD location, assuming the image will be stored in both places. For instance, a storage location such as harddisk/projectname/images/p1/p0000001.tif could be used to indicate that the image p0000001.tif is in the directory p1 in the directory images . . . on the basic hard disk and could also indicate that the image is on a CD or DVD called p1. Thus, each directory in the images directory would be limited to the number of images that would fit on a CD or DVD. As an alternate, all images could be in the image directory, with a calculation to indicate the CD or DVD number, e.g., images p0000001.tif through p0000010.tif must be on the CD/DVD labeled p1. There are many ways this issue of storage location can be handled, including storing all images on their own hard disk, either a second hard disk in the project computer or an external hard disk. In such a case, there may be no need for a database entry for the file location if all files are in a single directory (not recommended for a project of any size since the number of large files would be substantial, sooner or later exceeding the capacity of the disk). The important point is simply that the storage system needs to be clear and explicit from the beginning, and any changes must also be clear and explicit. The system may depend either on explicit file locations or explicit rules. Regardless of the system, it must be designed and enforced early.

There is another issue that effects image file storage. What kind of image do you need to have available at any given moment? That is, if you take and store all photos in RAW (camera native) or TIFF format to make sure that you get maximum detail, you may still want to have available only smaller, compressed JPEG images for regular use. In that case, you may need to have a dual-storage system. (There is server software that can deliver images at any resolution, on command, but that is not yet available on PCs or MACs, though browsers can now re-size images to fit in the browser window automatically, assuming the images in question are stored in a format the browser can read.) Doing so is actually rather simple if, once again, you use a calculation or direct name in an explicitly defined system. For instance, the same file name used above, with all the directories including the images directory (harddisk/projectname/images/p1/p0000001.tiff) could simply be altered by a formula to make a new sequence with jpgimages as the directory name in place of images, the result being a pointer to all JPEG files (harddisk/projectname/jpgimages/p1/p0000001.jpg). The remainder of the directory structure would then be identical; so the disk directories might look like this:



The storage system described here is simple, and it permits the database management system to locate any image simply and directly. It is also very flexible. If, for instance, there are multiple iterations of images, with the unaltered original being one and various alterations being the others, a similar directory structure to the one suggested for JPEG images could be constructed easily, making every image readily accessible – and making it possible for the user of the system to request any or all of those images at any time.

It should go without saying that, if there are film-based images, the individual images must be catalogued so they can be located. Of course, it may be more desirable to scan those film-based images in order to add them to the "born digital" images and have a single, all digital photo repository. Further discussion of the issues introduced by film may be found in the following chapter; that chapter deals with digitizing material from old projects, and film-based images are an important issue. In that chapter there is also a discussion (that could as easily be included here) of the problems that arise when deciding which images should be treated as a part of the project's archives. (See Chapter VII: p. 231 ff., 238 ff., and especially the Side Bar, "How Many Photographs and Whose Photographs?" p. 238.)

Non-photographic images

Among the images may be drawings from illustration programs – line art, charts, graphs, maps, etc. There will also be drawings from the CAD model, either direct CAD output or CAD drawings that have been modified with illustration software, and there may be similar maps and drawings from GIS work. Though not photographs, any such images must also be preserved as part of the archives, and the existence, creator, creation date, function, software, and location of the images should be clear in the database.

As is so often the case, the issue here is really advance planning. Because the number of images on a project of almost any size is likely to be overwhelming, it is critical to make plans early. It is also critical to make certain that all understand the importance of the storage system – and the importance of preserving unaltered original images.

Audio and Video Files

Audio and video files may be treated in much the same way as images, and like images, they must be culled to eliminate the chaff. The number and size of such files may otherwise be as overwhelming as the number and size of image files. (Included in the generic term video files may be such things as QuickTime VR files or any similar file type.) A table for each such file type must exist in the database, and each table must contain the file names and directory structure leading to the files as well as the information necessary to serve as a catalog – unique id, creators, dates of creation, file formats, subject, and so on. (It could be argued that all file types from relatively rarely used digital technologies could be catalogued in a single data table, but it makes more sense to put all similar files in a single table so that one may examine all files of a given type easily. The multi-table design also makes it easier to keep the data tables normalized. Nevertheless, one might easily put many kinds of files in a single table without pangs of guilt.)

As with images, there will doubtless be a need to store audio and video files on CDs or DVDs because they will be large, but a single computer may hold all the files for regular use. Once again, the directory structure is important, not in the sense that a given structure is required but in the sense that a well-designed structure must be in place from the beginning.

As with photographs, original audio and video files should be retained as part of the archive, regardless of the editing processes applied. A scholar in the future should be able to return to the original files to see and hear the original content.

Added information will be required, such as the software used to process audio and video information. The original file formats should also be noted; there are not as many well-settled and industry-wide standards in this area.

A special issue arises for audio and video files. Before starting to make such files, the project directors should carefully consider what they want and why. It will be extremely time-consuming simply to review the materials one time, much less to edit them to make useful resources from the raw material. After all, for one hour of audio, someone must spend an hour listening simply to hear it; more time is required to make something useful of the recording – much more. At least one excavation of which I am aware started with grand plans for video without considering the implied time commitment. The result has not been pretty.

Virtual reality files, if created for the project, should be less difficult to review and catalog. They deserve their own data table.

Text

Text files should be a simple subject. All scholars are accustomed to using text files for a variety of purposes, and there seems little mystery in such files. However, there are substantial difficulties in making text files part of a project data set – and in trying to make the digital archives complete.

First – and not particularly difficult – is the need to keep copies of all correspondence, grant requests, permit requests, permits, and so on. In some cases, though, digital copies without signatures or stamps from government authorities may not be adequate. In others, incoming mail or relevant documents may be in paper form, not digital form. While circumstances may mean that the physical copies should be kept, it does not mean that digital versions should not be made and retained. Consider, for example, an excavation permit – perhaps with a seal or signature of importance – received by mail. If there is a copy of the document in some word-processing format or as a scanned image, it is much more convenient (and less risky in terms of potential loss) to keep that at hand rather than the paper version. Scanning the official copy of such important documents and storing the scan on the computer is an excellent way to make sure that the digital archive is complete – even in years to come when the paper copy is no longer needed. (When possible a digital copy in word-processing format should also be stored for any document requiring text searching.) In general, all business-type documents should be stored using this multi-pronged approach. Whenever a physical copy is required because of signatures, notes, etc., it should be scanned as well as safely stored as a physical object.

All these documents need a storage system so that they can be found, something businesses have long since realized. They have document naming standards that assist. While that might be ideal for an archaeological project, it is not a likely procedure for scholars. Some simple form of directory structure should do the job, but, ideally, there should be a database table for documents, as for images, so that all can be found easily.

A far more difficult question arises when the text in question is text created as part of the project – excavation day books, articles and other presentations, publications, notes made as part of an analytic process. All those documents should be part of the record, in an ideal world. But this is not an ideal world; so . . .

Day books are the most interesting of these text documents to me. Day books can be truly critical items for scholars looking back at an important site. Despite their importance, day books are contemporaneous documents that are unlikely to have been made in digital format. Though some projects will use a laptop computer or smaller hand-held devices directly in the field, most will stay with pen-and-paper day books. Should those day books be transcribed, scanned, or . . . ? This is one of those perplexing questions that is both practical and theoretical at the same time. Given the importance of day books, it would be good to scan them and transcribe them, but many would question the expenditure of time required for the latter. Project personnel are usually too busy to do the transcribing day-by-day; so the extra burden is likely to be very significant.

A compromise is to scan the day books initially and then ask that anyone who uses them share with the project any transcriptions made in the process. That may not result in the transcription of the majority of the day books, much less all of them, but it will be of some help. (This suggests one of the things I have long believed – that transcriptions of hand-written text from any archaeological project should be shared with the project and made part of the archives, to be used – and altered or augmented – by anyone interested in the products of the project.)

What if the day books are created in digital form or transcribed? Is it sufficient to let each day book be a single, season-long text file (in a non-proprietary format such as PDF or RTF, of course!)? Should there be links to the project database – either in the data tables or in the text files? The answers to those questions depend on the organization of the project and the database. That may seem to be begging the questions, but, if the discussion of databases has been as informative as I hope, it should be clear that this kind of data organization is best determined on a case-by-case basis. However, some use of a data table(s) to catalog the day books is critical to assist anyone wanting to find contemporaneous references to objects, features, or contexts. (Day books are also discussed in Chapter VII, pp. 225-226.)

Publications of all sort are also parts of the project archives, whether they are short articles or multi-volume site reports. Articles in scholarly or general-audience journals may have been submitted in a word-processing format, and they can be made a part of the archives in an appropriate non-proprietary text format (along with any illustrations, each in its own appropriate format). Scans or PDF files of the resulting publications are also appropriate for archival storage.

Book-length publications, though, are unlikely to be in simple word-processing formats. It is more likely that they will have been created with desktop publishing software such as Quark® or InDesign®. Retaining the files is, as always, important. But, realistically, the files will be of little use for most project personnel unless they have been trained in the use of the underlying programs. More useful will be PDF files exported from the desktop publishing software, like the ones that have been put on the web for the publication you are now reading. Those files can be read with free software and, an important advantage, the files are hard to modify, even intentionally; so they can be made relatively tamper-proof. Thus, while the desktop publication files should be retained, the PDF files are the ones that will be more important for the project archives.

Not surprisingly, these publications present yet another database complication. A table – perhaps more than one – must exist to allow any user of the system to know of all publications produced as a product of the work and to know how to find them, both via bibliographic entries and within the project archives. The same or related tables should be used to record related publications about the project or project data. Ideally, of course, publications from related projects should also be recorded in a data table.

Miscellaneous Files

Other files – spreadsheets, email correspondence, and God-only-knows-what-else – deserve the same care. Data tables with appropriate information about the nature of the files, the formats, the subject matter, the persons responsible, and so on should exist for any file type used. Even if there are only a few such files, perhaps only one or two of a given type, a simple data table with a great deal of the information in note columns will prevent the frustration of knowing that there are – somewhere but who knows precisely where? – results from some analysis or correspondence on a given subject.

One other file type should be mentioned explicitly. If a project does its own bookkeeping and decides to retain bookkeeping records as part of the total project archive, it would probably be prudent to put the bookkeeping records into either a standard database or a spreadsheet. That will make it far easier to keep the files in useful form for the future.

Paper Records

Not every excavation or survey project will have all records in digital form. Whether the paper records are photographs, permits, grant applications, or any other kind of record, the greatest problems are knowing that they exist and finding them. As a result, even if the records are not to be digitized, there should be information about them in the project database. It should be possible to learn what exists – whether digital or not – and to find out where any information is – again, whether digital or not.

Controlled Vocabularies

In archaeology generally the vocabulary used has never been precisely and carefully controlled, though there are specialties within the discipline that have developed very well-controlled vocabularies. Virtually all ceramic styles have such vocabularies, though the range of their use can be very narrow and unanimity achieved mostly in the breach. Certain categories of faunal remains, chipped-stone tools, and other artifact types also have well-understood, limited vocabularies. Even in those fields, the control is often implicit and assumed rather than required by any explicit set of acceptable terms and definitions, and the meanings of even well-defined terms are not necessarily static. In the specialties that exhibit less consistent vocabulary, getting agreement is very difficult – often not because agreement is so distant a goal but because few scholars are willing to spend the time and effort required to come to agreement. In particular, archaeologists who excavate are not eager to spend their limited time on terminology issues when their own work is not necessarily made easier by the effort. Of course, at an even more practical level, there is little credit in the discipline or in the academy generally for work on such issues. A young scholar is not likely to gain renown – or tenure – for working on controlled vocabularies.

The absence of controlled terminology in the field in general yields substantial ambiguity. As a result, terminology remains an area of great difficulty for the discipline. International groups do exist to deal with terminology issues, but they generate little enthusiasm and are often thought to devolve into sterile arguments over precision. Nevertheless, properly controlled vocabularies are the only avenue to unambiguous descriptions and recording, hence the only route to sharing of archaeological data widely. Therefore, it is incumbent upon all who record data to do their best to find and use well-considered vocabularies and to participate in efforts to generate more and better controls for terminology in the discipline.

It is also necessary to document the vocabularies used – all of them. This includes local terms and more widely-used ones. In most cases it will be sufficient to name the resource used for a set of terms, but there are likely to be some purely local terms as well. There should be no doubt about the meanings of any terms in the data set.

The need for controlled vocabularies will be more important in the future as scholars can excavate less and consequently must work more and more with data from past projects. To the extent that the data require time and effort simply to reconcile the information, understanding will be made more difficult.

XML

XML is the acronym for eXtensible Markup Language, a subset of SGML (Structured Generalized Markup Language). SGML is also the parent of HTML, the language of the Web. XML provides a convenient and precise way to mark text so as to make the content absolutely clear. To oversimplify, XML is a system for tagging text to specify meaning, originally as a way to transmit data from a data table but now used by some in place of data tables. Thus, <last.name>Smith</last.name> makes explicit the fact that "Smith" is a person's last name. The simple text passage "<last.name>Smith</last.name> <first.name>Robert</first.name>

<middle.name>James</middle.name> can be used to store or transmit data in an unambiguous form just as if the three data items were in a table.

XML depends upon very carefully formatted statements that define the data categories to be used and the hierarchical relationships that will apply, not to mention the order in which the data are kept. (The relationships are hierarchical by virtue of the nature of XML, though it is possible to express more relational-database-like arrangements with some effort.) The definitions may be within the document being defined or in external documents that apply to a whole host of other files. In either case, XML depends upon well-defined terminology to function. The terms may be defined in the XML documents, but they must first have been defined in the discipline at large to be useful in XML.

XML has been seen by many as a technology that brings huge benefits to archaeology. By forcing the specifications of terminology, it is seen as promising to bring terminological consistency to the field. In my view, however, terminological consistency cannot be effectively imposed by a technology – XML or any other – but must be arrived at via agreement among the practitioners of the discipline. Until and unless archaeologists generally have agreed to define their terms tightly and then to map those definitions to some list of XML specifications, XML is extraneous. The controlled vocabulary is the first problem; archaeologists need to deal with the vocabulary issues before worrying about a technology that requires a controlled vocabulary. (The reader should be forewarned that the foregoing is my view and may be a minority one.) This is a technology that has won many advocates in the past few years; so it may well thrive. Its use in the business community has become widespread for data transmission.

Copyright

This is a topic that belongs in no particular place and, at the same time, everywhere. Regardless of how much scholars may wish it to be otherwise, access to information can be prevented via the assertion of copyright. This is a more difficult problem with documents, no matter the kind of document, than with simple facts. That is, one cannot easily copyright the fact that a given artifact came from a specified locus, but one can copyright the data table containing that information. Or a photograph of the artifact. Or a video of someone explaining the link between artifact and context. Or the lengthy blog with a discussion of the matter.

As a result of the potential for problems with copyright, anyone involved with project materials, digital or otherwise, must be sure to understand and attend to matters of copyright. For instance, anything considered the product of the work should be clearly identified as belonging to the project, even if some of those products may be shared openly with individual team members or other scholars or the general public. That would apply not only to photographs taken by anyone working on the project but even to a blog operated from the project work area – or from a university office in the off season if the topic is the project.

Even if a photographer from an magazine or a cameraman from a local TV outlet or a sound technician from a radio station should visit the work site or the offices of the project and produce something about the project, it should be clear who has what rights to that person's output. This is not to say that the project needs or wants to control an outside photographer's photographs, for instance, but it is to say that the project has a natural interest in the product. It should at least be clear what rights the project may have to any materials produced so that, should those materials prove to have real value to the project, there is no doubt as their availability.

Because any discussion of copyright seems to imply an attempt to prevent access to something, this is not an area to be treated casually. It should be clear that the copyright control sought by a project will not be used to deny access to others but only to ensure access for the project, and copyright language to that effect can be drafted for the project.

Copyright should also be used to ensure that materials obtained from the project directly are the only versions in general circulation. That is, by asserting copyright, the project can and should see to it that accurate copies of anything in the public domain are available through the project and that copies from other sources are understood to lack that stamp of authority. This is not to create problems for those who need access to project data. Quite the opposite. The intent is to make certain that there is a single, *bona fide* version of a data file or a photograph or a CAD model that may be relied upon by any user to represent accurately what the project has learned and documented. Such a procedure does not prevent adding new information or changing the files; it does prevent doing so and calling the files by their original names or otherwise passing them of as the product of the project itself rather than an updated or altered version thereof. This is critical for a discipline so dependent as archaeology on the sum total of its factual base; the distinction between what an excavation or survey learned and what was subsequently thought must be maintained.

VII

Digitizing Data from an Extant Project

Introduction

Up to this point, we have been concerned with digitizing data as those data are gathered in the process of excavation or survey. In dealing with this "born-digital" data, it has been possible to explore one digital technology at a time. In reality, however, a great many projects come into the digital age in mid-stream or after all data have been gathered, and such projects must be treated differently. Data not born digital must be re-cast for their digital expressions, sometimes with the need to add new, yet-to-be-gathered data directly in digital form (when projects are still ongoing) and sometimes not. But in either case the process of building a digital data set from an existing set of paper-based data is a very different one, requiring different approaches and different processes in order to achieve different ends. Each such project will also be expensive, and digitizing completed projects may also be difficult to fund because there will be many older, completed projects needing to be digitized and competing for scarce resources.

Every digitizing project will be very different from every other, ranging from a simple data table constructed of a particular object type from a particular excavation to a more-or-less complete digitizing of an entire data set so that the data can be gathered, unified, and made available on the web. As a result, the following may seem full of generalities. It is simply not possible to include many detailed examples – and using real-world examples would often be unfair, not to mention impolitic in the extreme when the examples are not positive. So where this discussion is vague, I can only apologize in advance and urge you to read with your own real-world examples in mind. Indeed, it would be a good idea for any reader to give some thought to an old project for which the reader would like a digital data set and then jot down expected issues of importance and anticipated difficulties.

Defining the Scope

The first and probably most crucial difference between preparing the way for born-digital information and digitizing paper-based information is the need to define the scope of a project to digitize existing data that were recorded on paper. Whereas digitizing incoming data in a project generally means digitizing all data; that is not the case when dealing with projects that have amassed data in paper form over some period of time. In its simplest form, digitizing from paper records may be nothing more complex than making a single data table from published information about a particular project to enable an individual scholar to use the data more effectively in his/her related work – a table with data about a particular object type, for instance, or perhaps a table of chronological horizons and markers. In its most complex form, though, digitizing old data may involve an attempt to create a complete data set as complex and sophisticated as any devised for born-digital information.

Personal Projects Are Never Simply Personal

The smallest and simplest of digitizing projects will involve putting a small portion of the data from a long-completed project into a data table or perhaps simply a spreadsheet. The data in such an instance will probably be found in a publication or publications from which they may be taken for digitizing. Once digitized, the data may be queried, subjected to one or another statistical analysis, or examined for group characteristics.

Such a project may seem to require little or no forethought since the scholar doing the work is the scholar using the results. That scholar can simply set out to enter the data as he or she wishes and go to it.

Since scholarship is built slowly, however, with new work based firmly upon predecessors, even such a seemingly trivial task as moving published data from something like a catalog into a table should not be done without some care. The process should be documented so that, should someone else want to use the table/spreadsheet, that secondary user will know how the data may be used. Similarly, the steps outlined here concerning fidelity to the original data must be honored.

Personal projects may seem to have little impact on anyone beyond the scholar involved, but simply publishing that scholar's results requires a trail of evidence that must be well documented.

Most projects, of course, will fall in the middle ground. They will involve some but not all of the extant data, some but not all of the technologies already discussed here, some but not all of the interconnections possible. Therefore, determining the boundaries of the project is the critical first step. Nothing can profitably be done until those boundaries have been established.

On occasion a project that has been in progress will be reconfigured to use digital recording systems in the midst of the project. Such a decision requires both that all the kinds of exercises discussed in previous chapters be undertaken for the sake of a completely digital approach with the new material AND that all the paper data be translated into the new digital form so that everything can be used together. (While it is possible that the directors of a project may choose to digitize new data but not extant data, I believe that to be such a mistake that it will not be considered an option here. Data should be all digital or all on paper in my view. A bifurcated approach might be defensible if a new director of a project charts a new course – especially after a hiatus of some duration – and expects little or no overlap between the old and new data, but how realistic is that?) This process of moving from a paper-based to a computer-based recording system presents such a different set of problems that it will be discussed separately, later in this chapter.

Who determines the scope of the project and when?

Whether a project is large or small, simple or complex, the scope will be determined by a small group of people, perhaps only one person, who have some specific interest and, presumably, funding for the work. The needs of that group or person will not only determine the scope of the project, but will almost certainly limit changes to that scope – because the original budget will have been determined by the planned scope. As a result, questions about project scope should ideally be aired before funding is sought so that early planning is as thorough as possible. That, in turn, means in most cases that some computer-savvy personnel must be involved and included in discussions of scope before there is funding. Otherwise technical issues may be understood too late to be effective.

This is one of those chicken-and-egg problems for which there is no ideal solution. If the computer experts are consulted late in the day, they will have no input in shaping the project scope. If they are brought in to the project early enough to have input as to scope, how will they be paid? Whether planning grants are sought or computer experts willing to consult without a guarantee of pay can be located, some solution to this dilemma must be found. If that solution is to omit consultations with computer experts until after scope and budget have been determined, the quality of the planning – and of the results – will be negatively impacted.

Determining Scope

The process of determining the scope of a digitizing project begins not with information but with people. Some scholar(s), museum professional(s), or funding agent will have decided that, for some particular reason, information from a project should be digitized. The reason may be general: this information should be in the public domain so that any scholar can find and use it; or it may be very specific: our new project needs better access to the data from that old project. In any case, the needs will likely be expressed in relatively general terms at the beginning. It is up to all, but especially the lead scholar and the computer personnel, to refine the needs and goals in light of their knowledge of the data, of the technical difficulties to be encountered, and of a predicted cost:benefit ratio.

A variety of issues must be considered. The first and most basic is whether or not the entire data set of the project should be digitized. It is unlikely that this will be necessary or desirable unless publication and analysis have not been started, but the question is the first to be asked.

A choice to digitize all may be reasonable if analysis is still in the beginning stages. Otherwise, the need for complete digitizing is hard to justify. The costs will be substantial, and they will escalate with each addition of a data type to the set of data to be digitized.

Deciding on narrower boundaries is very hard, however, and there will often be disagreements as those boundaries are sought. The beginning point will almost always be the question of use. Who will use the information from this project, and what information must be in digital form to enable that use? If anything is to be digitized, the object catalogs will be the easiest to justify. The objects, after all, are part of the general corpus of the discipline. For surveys the most obvious choices will be the survey data that define the collection areas and the object data sets.

What about transects, lots, loci? With each question there are multiple considerations. Is this particular set of data needed by a defined user? a potential user? a broad group of users? May the site or survey area be used for critical dating evidence by virtue of links to other sites/survey areas? Are there doubts about analyses that have already published? Have finds been used to cross-date material from other projects? These are the kinds of questions that will guide the decision-making. As each new table or data type is debated, the need for the data must be clear, and the extent of that need must be determined objectively.

Seemingly subjective questions about the quality of the project work when it was originally performed will also have an impact. Older excavations, for instance, may have been excellent when viewed through a contemporary lens but lack modern levels of precision. A modern excavation may have been sloppy, or the coordinate system for a survey may have been poorly established. In each of these instances how much information should be digitized if the precision or quality is uncertain? This is a difficult question to approach because it involves judgments about basic accuracy and precision of the original project data. Nevertheless, all these kinds of questions must be faced squarely. Digitizing poorly recorded information is of little value; more important, it may put into the archaeological record incorrect information which, simply because it comes from a computer, will carry an unwarranted stamp of authority.

All of these discussion and debates should be documented so that there will be a record of the decisions made and how they were made. Such documentation may prevent significant disputes in the future.

In the foregoing there have been many references to cost and to limiting the work. That may seem strange in a book such as this. After all, this is a book about the use of computers; I am an advocate. It is critical for all who work with digital data, however, to remind themselves again and again that the computer is only a means to an end. If the means becomes an end in itself, the aims have been perverted.

Digitizing for Preservation

In some cases materials that need not be converted into a more complex digital form such as a data table, do need to be available in digital form for the sake of internet access or portability or because they are subject to decay. In such instances scanning may be appropriate and sufficient. Paper may be readily scanned for such purposes, whether in the form of index cards or notebook pages. A word of caution here. Scanning to make the material available as page images assumes that there are systems in place for archiving the computer files, but the original paper should not be discarded; it should be stored in some safe, dry, temperature-controlled environment. This, of course, is true for all paper records, but it can seem reasonable to discard material that has been scanned because scanning involves no interpretation or analysis. In truth, however, scanning does not make discarding the originals a good idea unless someone is assigned to examine every scanned document to be sure the scan is complete and accurate. (This may seem overly cautious, but examinations of the results of putting old newspapers on microfilm have shown that accidents not only may happen but surely will.)

The foregoing implies that some but not all paper records will be scanned. It should not escape notice that such choices – scanning some paper records but not others, not to mention more actively converting some documents to a digital form but not others – are the end product of a value judgment. Some material has been judged to be more important that other material. Such judgments are an inevitable part of the digitizing process, but they must be made self-consciously, and they should be documented so that any user can learn what materials were omitted. This may seem far-fetched, but imagine letters to his/her home institution from a director of a long-running excavation. Those letters may contain both personal and project-related information. If they are made available at all, how does one treat them?

A Holistic Approach

Once the scope of the digitizing project has been determined, the project as a whole must be considered, and the temptation to deal with pieces of the whole, one at a time, must be resisted. Decisions about digitizing data must be made with the project as a whole in mind. Bringing digital technologies to a sea of pre-existing paper records requires that the digitizing work be approached as a complex, unified project, not as individual technical areas. Otherwise, the odds on a disconnected and disjointed series of poorly-related pieces are too high. At the end of day, the pieces must be well connected to one another or much of the effort will have been wasted.

Hidden Issues

A host of problems unique to dealing with data from old projects have little or nothing to do with the conversion of data to digital form, but they must be acknowledged at the outset. Many of these problems have to do with questions of responsibility and authority for project information when the project has been completed; these are people questions, not data questions. It is very important that those involved make every effort to understand and to honor the lines of authority for project data, lest other, unnecessary difficulties be introduced. In some cases, of course, there will be multiple lines of authority and even conflicting lines. There may also be differences of opinion as to the value or desirability of the digitizing work and consequent internal disputes. Tact and care will be critical in such circumstances, and nobody should start on a path such as this without a ready store of patience. The older the original project, the less important this may be, but the matter of who sees himself or herself as a stakeholder is critical, even when the individual's view is divorced from reality.

Tact and care may also be required to deal with the possibility that some project personnel are still working on the project but unable or unwilling to deal with digital records. If this circumstance arises, it cannot be ignored, swept under the rug, or assumed to be about to disappear. Every effort must be made to keep all who are working on the project both comfortable with the plans and comfortable with the data and their access to them. The foregoing contained no recommendation, but the implication of undertaking such a digitizing project is that digital data are preferred; that view cannot be abandoned if the work is to succeed. Nevertheless, a project participant who is computer-phobic must not find the material no longer accessible and must not believe that his/her needs have been ignored.

There may also be concerns with publication rights and responsibilities. If a project has not been finally published but publication rights have been parceled out, there may be significant issues needing to be settled.

If the previous few paragraphs seemed dry, that is because it is difficult to speak in the abstract about the kinds of personal issues that are so often involved when dealing with older projects. It is easy to think that we are all just so many automatons doing work as if there were only rational processes when, in fact, much of our work ultimately becomes emotionally charged. Every project will have its own complications, from physically frail but committed original participants to jealous and protective ones. Nobody should start down a digitizing path for an old project without understanding that such obstacles will exist and knowing that dealing with them would try the patience of a saint.

Finally, it will be imperative that there be some kind of a committee to conduct the planning. The responsibility for the project must not lie exclusively on one set of shoulders. There are too many portions of the planning and execution that need open and thoughtful discussion of alternatives, debates of multiple points of view, and considerations of differing approaches. There are also too many potential users, some of whom will seem obvious consumers of the data to one scholar but not another. Thus, one person should not undertake the planning and overall direction without assistance, in the form of colleagues who are prepared to offer the time required for careful review, debate, and discussion. The foregoing should not be taken to recommend that the individual committee members divide the work load by taking on individual pieces of the project. Such an approach is doomed to fail because the requirement of such a project is not the creation of pieces of the whole but the creation of a complex unified whole, which requires many heads for the sake of unity and completeness, not simply to spread the work.

The needs and demands of this work are significantly different from those of the new project that has the luxury of beginning with something approaching a *tabula rasa*, and the problems are quite different depending on whether the project in question has been completed and fully published, terminated as to excavation or survey work but not fully published, or is on-going. Indeed, the differences are such that I have chosen to discuss separately the processes for working on a finished project and those for working on an on-going one, leaving to the reader the differences that may exist between a fully published project and one that has completed the data-gathering phase but not the final analyses and publications. I will begin the discussion with projects that have been terminated. There are many problems with such old, completed projects. While the issues are also relevant to the processes of digitizing on-going projects, such continuing projects present still other problems and complications.

Digitizing Data from a Terminated Project

Digitizing a project that has been terminated must begin with a thorough study of the project, beginning with a good, thorough history of it. All the permits, research proposals, grant proposals, and similar descriptions of the project should be located and studied as the first part of building that history, and all publications should be catalogued. The initial aims of the project and adjustments to those aims are the most obvious of critical elements; the nature of the records will reflect those aims both by what they include and by what they ignore. (It may seem obvious, but the earliest step should be to determine whether permissions for the digitizing work are needed from institutions, governments, or funding agents. In some cases legal permission may not be needed, but courtesy will require a polite request nonetheless.)

Personnel involved, including an up-to-date listing of those still active in the field and those no longer active but still living, are critical both because the participants have information about how well the aims were carried out and because their advice and counsel will be invaluable when trying to understand the records. If any of the personnel are elderly and in poor health, they should be interviewed early on, and the information obtained should be properly catalogued and preserved, as should all the information obtained during this work. Of course, the personnel from the project who are still involved in some way or no longer involved but willing to be of service, will be important advisors. The records just mentioned – proposals, permits, and so on – will provide the bases for many of the most important discussions with the participants. Those documents will be particularly useful in guiding discussions of the distinctions between plans and reality for the project.

The participants should be able to provide a great deal of useful information about the way the project operated, the attention to detail followed, the kinds of errors to be expected, and the like. A long-running project that passed from one generation of participants to another is an especially difficult subject since the differences between approaches and emphases may be all but impossible to sort out. Nevertheless, every effort should be made to understand such matters. They will prove to be very important. Imagine, for instance, the excavation that uses multiple names over various seasons to indicate the same structure. These kinds of things happen more often than we care to admit.

The project participants will also be critical when the actual project records are examined. The records, after all, are more than data repositories. They are also indicators of what was deemed important and what was not; project personnel will provide corroboration at the least and may well be truly critical to a proper understanding of card files and other records. (It might even be worth the time and effort to record these conversations.)

Seemingly simple matters such as survey systems may not be so straight-forward and obvious as expected, and in some cases there may have been multiple competing systems over the years, either intentionally or not. How many projects have established a datum point for survey purposes only to find it gone the next season – or, worse yet, to find that it had been moved slightly by winter freeze-and-thaw processes but to learn that only after using the datum for a part of the next season? A new study of the evidence, which is what the digitizing process involves, requires understanding such matters, and the people involved are likely to have critical information in this regard.

Dealing With Reality

Much of this discussion seems to be based upon an unstated assumption that the paper records that supply the starting point for digitizing are themselves clear, accurate, complete, and uniform. Sadly, we all know that such an assumption is incorrect. In fact, there will be incomplete, obviously inaccurate, and illegible records. There will be drawings that have corrections, not to mention smudges and incomprehensible squiggles. There will be illegible handwriting in more places than you thought conceivable. There will be misspellings that defy correction, dimensions that defy reality, and photographs that seem to have been taken on another planet.

The possibilities for these frustrations are nearly endless, but that does not help us figure out what to do with/about them. Indeed, there is no simple answer or set of answers.

You will not be surprised that the first piece of advice is simply to document the problem. Scan the material if possible to show clearly the issue, make the choices clear and explicit, make your final choice equally clear and explicit, make the consequences to the larger data set equally clear and explicit. If the problem is one that can have wider repercussions, make that clear as well. Finally, make sure that the documentation about these kinds of problems – problems, after all, with the most basic level of the information – are seen by any user of the digital data as early in that user's examination of the records as possible. That is, put the documentation in the most prominent location you can.

At the end of the day, neither you nor anyone else can – or should – pretend to certainty where none exists. Therefore, the best thing to do is to find the best, most clear and explicit, most difficult to ignore way to present the problems honestly. If your choice for the correct reading of the squiggles or the letters intended under the smudge is proved wrong at some future date, the discipline will have been well served by your careful documentation.

Even such prosaic matters as the photographs can be enlightened by project participants. For example, I worked at Gordion, Turkey, in 1975 to take photographs for the first volume of the site publication because too many of the existing photographs were not only unacceptable for publication but simply substandard. In the process of preparing for that and of examining the photographs taken by the project director, Professor Rodney Young (by then deceased), it was determined that Mr. Young spooled film on site in order to save either money or, more likely, shipping space. However, he often spooled the film improperly, inside-out, with the result that many photographs were taken with the light-sensitive emulsion on the wrong side of the supporting medium (the film). The result: seriously under-exposed, low-contrast, not very sharp images that are always printed backwards unless the darkroom technician has been forewarned. Who could reconstruct such a sequence without the memories of those who had worked on the site with Mr. Young? Knowing this, however, it is now possible at least to order prints from his negatives with the specification that they be printed in reverse if the negatives are low-contrast and under-exposed.

At the end of this long preparation it should be clear what the aims of the original project were and were not, what information was most earnestly sought, what kinds of things might have been overlooked, what processes were rigorously defined, what processes were casually defined, and so on. All these things will have an impact on the ways different data sources are evaluated and the ways they are converted to digital form. For example, if ceramic colors were casually recorded by members of the staff without standards or color charts, this information should be part of the digital record to make sure that future scholars do not put too much faith in the color information. In some cases, it may be concluded that the data are not sufficiently reliable to be used seriously. Frustrating though that may be, it is better to understand the problem than to spend time, energy, and funds to digitize records that should not be relied upon or, worse yet, would falsely represent the found realities.

There will likely be critical issues to be discussed with the personnel concerning scholarly access and publication rights. All the publication and study plans made by the project personnel must be understood and respected. That is not to say that publication and study projects assigned but not completed must remain the responsibility of the person originally chosen; it is to say that such matters must be dealt with openly, honestly, and with the appropriate levels of consultation. In addition, those publications already produced should be studied for information about the ways data have been used and to understand the way those who worked on the project approached their data. For example, the presence or absence of regular and systematic chemical analysis of pottery says something about expectations that should not be ignored. (You may also find that there are significant corrections to the published record to be found in the marginalia of project-owned copies of publications. That information may be remarkably valuable and should be carefully preserved as part of the digital record.)

Having laid the groundwork for the digitizing, the real work of putting data into digital form for an existing project remains. It is an extraordinarily difficult job. It is so difficult because, generally speaking, the documentation about which we have talked so much in previous chapters is not available to guide project personnel as they attempt to graft digital techniques onto a paper-based recording system. Paper records and recording systems seem to speak for themselves. Something is recorded if it is there – on the piece of paper or the card in the file. Maps and plans are drawn in the way that is obvious from looking at those maps and plans that have been drawn. In short, a generalization that will surely offend some, there is usually little to guide the digitizing process in terms of written standards or procedures. That is why those conversations with project personnel are so important; examining paper records and talking with participants will likely be the only ways to reconstruct the kind of documentation we found to be so important in producing digital records.

Ideally there should be lists of data types recorded, with measurement units, limited vocabulary choices, and so on. Similarly, there should be drafting conventions to guide draftspersons as to line weights and types, scales used, the ways survey datum points are indicated, the ways elevations are noted, etc. To the extent that such documentation exists, the process of planning for digitizing should be far easier. It will be easier because many of the requirements will be more clear and obvious; otherwise those working on the digitizing must induce those requirements. In either case, there must be documentation to guide those who will translate the data into digital form before digitizing begins, and that documentation must become a part of the project archives. Users will consult it just as they would consult documentation of digitizing practices for a new project.

All of the foregoing work must be carefully documented; it will provide important foundations for the digitizing and for anyone using the digitized data. At the same time, however, much of the work may be implicitly critical of any number of project participants, and some may be quite explicitly critical. As a result, preparing the documentation requires considerable tact and a diplomatic touch. Perhaps the best advice to give here is the simplest. This documentation must be honest to be useful, but it should also be prepared with the kind of charitable approach you would like to see used when future scholars discuss your own work procedures in the light of standards of the next century.

Now we will turn to the various kinds of data to discuss the documentation that must be generated and how to move forward from documentation to digitization.

Field Notebooks

Day books or field notebooks are among the most important of the records of any project. If the digitizing process occurs after analysis and publication are complete so that the aim is simply adding to the digital record, the field notebooks may be less important. Even then, however, it may be decided that images of the notebook pages should be included in the digital record; this can be accomplished by scanning.

Before beginning the scanning, an inventory of the notebooks should be taken. If any is missing and cannot be found, that must be noted. In addition, the process of inventorying should include an examination of the ways the notebooks have been used so that, when they are digitized, the proper approach to indexing will be clear. One would assume that authors and dates are the key index items, but for every such assumption there will be an exception.

The notebooks should be scanned at 300 d.p.i., in full color, to create uncompressed TIFF files as the archival records. This resolution is almost certainly overkill, but it makes significant loss almost inconceivable; color is desirable, even if no apparent color has been used, because subtle differences in color, including faded paper, may be important. (Although it would probably be unnecessary in most instances, one might even include a color chart in each scan.)

The scanning process should be tested with and without the use of a black paper background for each page to prevent bleed-through from the back side of relatively thin paper. If the notebooks have different paper types, each type should be similarly tested. If there is any bleed-through, all scans of similar paper type should be made with the black background material in place.

The original scans should be archived, and reduced-resolution versions may be produced for regular use via computers, though this would seem a poor choice of expenditures for a completed project with archival access being the aim. In addition, advances in computer technology will render such reductions unnecessary, making the display size adjust automatically to suit the hardware. Even then, however, data storage requirements may suggest that low-resolution images be stored for normal access and higher-resolution versions archived for less frequent use and long-term preservation.

A simple database should be created for the day books. The database should make it possible to find any page from any day book according to the name of the author of the day book and the date of the work. Even if access to the individual pages may be gained without a data table, the data about their content should be in a table so that questions about the day books in general may be asked and answered, not just questions about what was written on a particular page. Other indexed links – for instance, links to notebook pages containing information about object findspots, based on object inventory numbers – may be desirable, but they may also require so much time and effort as to be effectively impossible.

As was noted in the previous chapter (Chapter VI, pp. 212-213), notebooks may be transcribed by scholars who use them. (The expense of transcribing notebooks makes it unlikely that any digitizing project would pay to transcribe all.) If so, the transcriptions should become part of the digital record so that they can be searched with simple text searches. If transcriptions do become part of the record, there must be a plan in place to deal with errors and updates of the transcriptions – a plan that does not remove portions deemed to have been incorrect but keeps all versions available for searching. Multiple transcriptions are unlikely in the extreme for a completed project, and executing the plan may reasonably be postponed until a need arises; this is a good example of the need for a well-defined scope, however. The plan for multiple transcriptions will be an important determinant of the underlying data structure; so multiple transcriptions must be part of the core system even if transcriptions do not exist at the time the data structures are determined and are not really expected to become part of the data set.

Data Tables

The first job with information seemingly destined for data tables is to make a full inventory of the information to be placed in tables. In what forms is the current information? Are there card files, notebooks, lists in other forms? It is all but certain that there will be lists of the catalogued objects. Other likely lists include lot lists, context sheets, personnel, photo lists, and lists of samples intended for testing. For each of those there must be lists of potential table columns. That is, examining the lists should yield a secure set of data columns and types recorded for each group: for instance, lists of ceramic objects might include data entries for body color (text), height (number), width (number), ... It is very likely that, in addition to the standard cards and notebook pages, there will be added notes and comments stapled to a form or paper-clipped to a page or scribbled in a margin. These kinds of non-standard data sources must be accommodated in the data table plan, as must incomplete forms.

Going through the lists might also yield the vocabulary for data entry used, but to what purpose in advance? The data entry process will create, as a by-product, a complete list of all terms for all entries. Since the actual data entry has already been completed, the obvious reason to create an advance list of terms would be to eliminate some terms and replace them with others, a process that should **never** be undertaken with extant data. The digital data must be faithful to the original paper version. On the other hand, advance examination of the data may, indeed should, be undertaken to determine what additions to the data will be required for a contemporary user. (See “True to the Original and Today’s User.”)

To return to the issue of data categories or table columns, the categories used in the paper lists will be used in the data tables. Those categories are the starting point; however, there are complexities. It is likely that, during the course of the project, information of a type not originally recorded has been added for objects found later in the work. For example, residue in pottery may not have been noted in the early days of a project but carefully determined later. In such a case, the date of the introduction of the new category is critical; the absence of information recorded prior to the addition of that category is different from the absence of residue on an object examined and documented later. Therefore, data about pottery recorded prior to the search for residue should make it clear that the observation was not attempted; otherwise, an empty column might be misinterpreted as the absence of residue, not the absence of an examination for residue. Particularly in a case such as this, where the record may only show the presence or absence of residue (with analysis in another table), there should be no confusion between an observation not sought and an observation that yielded a negative finding (no residue). Thus, "n/a" may indicate a category that is not applicable because the pot was not examined for residue while "no residue" or "no residue found" indicates that the examination occurred but yielded no evidence. (As indicated in "True to the Original and Today's User," the actual data table entry should be whatever was placed on the paper record, with, if desired, a "translation" added in another column – in another table. Thus, the actual entry might be only "n" for no residue with the translation supplied automatically.)

It is also very likely that, if a storage location for a find was recorded and the project is old, that location has changed, possibly many times; care must be taken to prevent moving such old and inaccurate data into a data table that makes it easy to search for information – but consequently easy to get incorrect information that is taken to be reliable. Faithfulness to the original may therefore require the entry from the paper records, but accuracy may require that there be a companion column – or columns – to indicate whether or not the information is reliable. (In order to prevent getting inaccurate information when searching for location in this example, there should be two columns, "original location" in the basic file and "current location" in a related file so that a user could not simply search blindly for location without knowing of the presence of two columns. The "current location" column should have two other columns associated with it, one for date of deposit and one for date of removal, and it should be in a child table so that all locations since the original one can be

True to the Original and to Today's User

The original data must not be compromised. It must always be possible to know, without doubt, what the original paper records contained. That is not to say, however, that a data table cannot augment paper data in ways that combine faithfulness to the original and maximum utility for users. Consider, for instance, the use of English measurements and metric ones.

If a project had used English units of measure, they should be retained. However, there is no reason not to add a companion column showing calculated measurements in metric units. The metric number can be calculated and the converted version of the measurement can either be stored or not, but the user would, in such a case, be able to see both what was recorded and a more familiar measurement that can readily be compared to measurements from other sources. It should be very clear which of the measurements is original so that possible errors may be more easily ascertained. (For instance, errors with metric measurements are more likely to involve misplaced decimals; those with English units are more likely to have errors with fractions.)

Another example might be the names applied to pottery. The term *amphora* may have been applied to all amphora shapes, regardless of size or use. Those in charge of the digitizing process, however, may prefer to call an amphora shorter than 15 cm. an amphoriskos, and they may want to distinguish transport amphorae from others. To deal with this situation, a column holding the original data would contain the term *amphora*, but a second column could be added to contain *amphoriskos* via a calculation (if shape equals "amphora" and height is less than 15 cm.) or *transport amphora* via a similar process. This solution requires that the second column contain either a new term, when required, or the original term when no translation is required; as a result, the second column holds all shapes in the preferred vocabulary of the new project personnel while the first column holds all the original designations.

These two examples show how a new data set can be faithful to the original data **and** of maximum use to the current user. Such solutions are not hard to find, but they are critical to providing both integrity to the original sources and usefulness for contemporary scholars, and they will require time and care.

Another step is required, though, to complete the process of making the data both faithful and useful. The added columns should ideally be placed in their own table(s), related one-to-one to the table containing the original data. That is, an ideal solution lets the base tables contain only the original data; added or converted data may then reside in tables clearly identified as containing data not in the original records. There might even be two such related tables, one for converted data and another for added data. This may seem overkill, but it is a good way to make certain that the original data are recognized as such.

recorded, making a full history of the locations of the objects available. Here again, it was not difficult to find a way to remain faithful to the data as recorded while nevertheless providing maximum utility to the contemporary user.)

During the course of the examination of the various data lists, of course, the potential relationships between/among lists must be noted. The tables should be related as fully as possible, though each table may be built individually.

At this point I may begin to lose you with generalizations. It is simply not possibly to provide many specifics without a particular data set to use as an example. You should read the following with a specific old project in mind if possible.

A modern student of an older project may insist upon including observations that were not made when the project was under way. To include a column for such observations may truly falsify the data, implying that the original participants recorded the evidence – or failed to when the column is empty. Adding such a column, then, requires that the column be clearly identified as new, not from the original examination. This can easily be done by adding a separate table with a one-to-one relationship to the original.

Similarly, outmoded vocabulary should not be changed. It may be augmented with "translations," as suggested in "True to the Original and Today's User", but there should be no doubt about the actual terminology used to describe the data in the first instance. This brings us back to the question of vocabulary. After the data have been entered, lists of all terms can easily be generated. The lists will show both the terms used and the errors – typos and misspellings. Typos and misspellings introduced in the course of digitizing by the data entry personnel should be changed, of course, but what about the errors in the originals?

Errors made by project personnel making written records may be diagnostic. That is, hand-writing is not like typing. We do not make the same kinds of errors. Therefore, someone misspelling when writing a word is more likely to be spelling it as he/she thinks correct than someone typing it the same way. So it may be useful to retain spelling errors, perhaps to be able to identify certain members of the previous project team. If such errors are retained, however, searches may not work properly; so table design must take these questions into account.

Again and again it will seem burdensome to make data tables that both preserve the original entry information and permit effective use. However, the best and most useful results will come from carefully preserving all the information. This is not a job, after all, that requires speed if speed means that accuracy, completeness, or utility must be sacrificed.

There will be many corrections of the written record already made and written on the file cards, notebook pages, or publications by project personnel. There will also be many hand-written notes changing the data – usually with the original information neatly crossed out. These differences of opinion, like those just discussed, should be honored fully. That is, the original and the corrections must find homes in the digital files, and there must be simple searches that will find all versions of the data. Preserving these differences of opinion will require the kinds of tables discussed in the database chapter to hold divergent views (see Chapter III, "Honoring Scholarly Differences," p. 98).

Drawings and Maps

Strangely enough, we begin again with an inventory. Here there is a serious question of competing technologies that might be applied; so the inventory information has a very direct impact on the direction of the digitizing project. The drawings may be appropriately converted either to CAD or to GIS form, and the choice should – as previously noted – not be made on the basis of what technology is familiar to project personnel. In some cases, the drawings may not be converted at all; they may simply be scanned for easy access. The choice here will depend mostly on whether or not the project is still in the analysis phase. If analysis is still ongoing, conversion to CAD or GIS may be valuable. If not, scanning will generally be sufficient.

Regardless of the ultimate destination of the drawing information, the first job is to scan the drawings as a form of archival preservation. (The scans may also be used as the bases for translation into CAD or GIS form, though, as noted above – Chapter IV, "Digitizing Existing Maps to Make Vector Maps," p. 138, and Chapter V, pp. 178 ff. – that is not the preferred mechanism.)

Very large drawings and maps will require an outside vendor for scanning, but both in-house and vendor-supplied scans should abide by the same standards. Scans should be in color (for the sake of uniformity, because of the likelihood of color shifts in the papers used, and to retain subtle color differences in pen or pencil lines). Resolution is harder to specify in the abstract. Certainly 300 d.p.i. is a minimum, but how high is reasonable? This is not a question susceptible to a general answer. Tests are the best way to settle the issue, and it may be determined that the best resolution is not a consistent number of dots per inch but a consistent number of dots per unit of measure in the real world. That is, drawings might reasonably be scanned at, for example, 100 or 1,000 dots per real-world meter so that all can be used together easily. Indeed, drawings might be scanned twice, once at the common scale and once at a high standard resolution for preservation.

Returning to the drawing inventory, part of the process should be an examination of the survey system(s) used in the project. The chances are all too good that surveying will have been accomplished in ways that have created problems when relating materials from different seasons. That is, survey datum points may have been moved between seasons or have been misunderstood; different project surveying personnel may have located important points differently, possibly inaccurately. Furthermore, attempts to reconcile different survey systems may require data that are not available. All these matters must therefore be investigated at the outset so that there can be a clear starting point – or at least a clear understanding of the problems to be encountered. In some cases, just knowing the problems may not be sufficient, and it may be necessary to reconcile different survey system before going further – or at least to try to reconcile them.

Similarly, differences from draftsperson to draftsperson – or from one point in time to another for the same draftsperson – should be identified at the outset so that the drawings from different times and different draftspersons can be used well together.

Regardless of the survey problems found, for older excavations CAD is more likely to the proper software choice if the drawings are to be converted, but for survey projects GIS is the more likely choice. Having said that, however, there remain important questions.

The most obvious and important question is simply the nature of the drawings. If they are mostly of excavation trenches and the finds within them, it is likely that CAD will prove the better choice. The area of coverage will not be large in such cases, and CAD's facility with layers will be useful. The drawings can be georeferenced (connected to real-world coordinates) in CAD as well as GIS; so that is not an issue. It will be advisable to trace the drawings as described in the CAD chapter, and most GIS programs will depend on tracing in a CAD program. In fact, that is one of the reasons to prefer CAD for excavation drawings since using CAD may reduce the need for multiple programs and multiple competencies.

One problem with using CAD is the implied precision of CAD systems. When making drawings it is simply not possible to be precise to the real-world millimeter, and the process of surveying, drawing (on paper), and tracing for CAD could not retain that precision if it were in the original. Therefore, CAD models from older excavations must be carefully created and checked to gain some sense of the preserved precision, and accompanying documentation must carefully define the problems and the limits on precision.

If CAD is to be used, it may actually help solve some of the survey problems found. When multiple drawings contain the same information – a survey datum, the corner of a structure, the trench corners – those drawings can be related correctly to one another even if each is based upon a different grid or different survey coordinates. As a consequence, CAD may make it possible to build slowly from one drawing to another, at each step relating the drawings to a common grid, even if the drawings are based on different grids. (Relating one drawing to another will automatically make it possible to relate each to the other's grid.) All the paper drawings would thus be pulled into a single CAD model or perhaps a few CAD models if the conditions warrant that. Once the process is complete, the entire model can be georeferenced via a survey process (either using GPS technology or, for higher precision, using geodetic markers and a survey process). The result should be an internally consistent model, whether it can be accurately georeferenced or not.

At the end point, it should be possible to use the results to understand better any errors made in the survey systems used over the years of the project. Of course, the system described risks error propagation. That is, if one drawing is connected to another with common points that were poorly surveyed, there will be some error and the error will be magnified when another connection is made that also includes some error. However, one can generally rely upon at least the internal consistency/accuracy of a given drawing; so the described process of combining data from different drawings can be very useful if the potential dangers are kept firmly in mind.

An issue not yet mentioned is the scale of the original drawings. If the drawings have been made at a small scale, everything is more difficult, and the potential value of the drawings is reduced. It may be necessary to perform some simple trials to be sure that the existing drawings will support the kinds of digital records that are deemed necessary for the project to succeed.

The process just described – adding one drawing to another to build a model – can be carried out in GIS, but, as noted, most of the tools for tracing are CAD tools that permit tracing in a CAD environment and then moving the data into a GIS environment. (Scanning yields drawings that can be directly imported into a GIS system; however, importing a drawing means that the drawing is the unit, and the constituent elements of the drawing cannot be separated from one another.) Unless the area under study is very large, say more than a square kilometer, GIS would add little to the utility of the finished CAD model. Even if the total area covered is large, with multiple excavation areas in each of several parts of the larger permit area, CAD would probably be a better tool. Each area could be treated separately but all could be related to a common grid. There could be some discrepancies in location due to the shape of the earth, but they would be of little significance, and georeferencing of each excavation area via a separate process would remove those discrepancies.

If an excavation has actual findspots for all objects, including them in a CAD model or a GIS data set is quite possible, but there are advantages and disadvantages to each approach. In a CAD model the objects would be represented by icons (different icons for different object classes) and placed on layers named to permit sensible access/display. In a GIS data set the objects would be represented by points, and the information about the objects would populate various data tables and could be accessed via the characteristics recorded in the data tables. (Note: With a GIS, the users might see only points on screen under normal circumstances, though the points could ultimately be replaced by icons in secondary processes.) In addition, a GIS approach would permit seeking objects according to their distance from some common point or area, their inclusion in a bounded area, or their exclusion from a bounded area. CAD would require using the screen to define areas of interest since topology is not a CAD concept.

To be fair, some would argue that GIS should be used even for excavation trenches and finds. Not being persuaded by their arguments, I will not try to summarize them here but merely warn readers that the views just set forth are not universally held. It should be noted in this context that properly conceived data tables might be used on both the GIS data set and the object databases. CAD models could also be linked to the data tables directly within CAD software, but that is a process I do not recommend. (See Harrison Eiteljorg, II, "Linking Text and Data to CAD Models, " *CSA Newsletter*, XIV, 3; Winter, 2002 - http://csanet.org/newsletter/winter02/nlw0201.html.) A data table could be used in a CAD environment to generate icons and place them automatically in the proper locations in the CAD model, otherwise leaving them out of the model altogether.

GIS is certainly the appropriate technology for data from a survey project. The mapping features and the understanding of topology are critical for survey data. In addition, there are few actual physical entities that need to be drawn, and the coverage area is likely to be large, requiring the understanding of the shape of the earth that is built into GIS coordinate systems.

There are different pieces of information from the inventorying process important for GIS, though. If maps to be digitized are at small scales, they will place significant limits on the potential of the GIS work; scale must be checked to be sure what kinds of analysis are made possible or impossible. (Scale is somewhat more likely to be a problem with maps than drawings simply because the larger coverage area of a map can force very small scales.) In addition, it may be necessary or desirable to obtain more modern maps of some of the areas investigated, and scales of those maps will also be critical.

Although I have regularly argued that software should be chosen on the basis of its appropriateness for the task at hand, I must here admit that, particularly if a project is not large, existing competencies cannot be ignored. If someone involved in the digitizing of a completed project knows CAD or GIS well or if either appropriate CAD or GIS software can be readily obtained, this is not irrelevant. No sacrifice should be made that might reduce the utility of the results, but there are sufficient overlaps with CAD and GIS here that many projects might effectively use either. One important caution: If there is some practical reason to prefer GIS when use of the GIS will require CAD tracing for input, the value of the experience or of software availability may be vitiated.

I must restate the possibility of stopping after scanning the drawings and using neither CAD nor GIS. Digitizing a project simply for access to its information, with no intention for further analysis of the site or survey data, may reasonably yield a decision to do no more than scan drawings. There is no reason to add the time and expense required to convert drawings into a CAD or GIS environment if there are no analytic gains to be had. In addition, it may be possible to combine scanned drawings with a graphics program such as PhotoShop®. If the drawings are all of the same scale (or can readily be converted to the same scale), a great deal can be done by combining them in a single graphics file.

Photographs

Digitizing photographs may be an important part of any larger digitizing project. Decisions about digitizing, though, must be based on a realistic understanding of the existing archive and the costs of conversion. There must also be a realistic appraisal of the relative importance of the imagery already in hand and of the variety of those images. So we begin, as usual, with an inventory, in this case first in an effort to determine the nature of the various films to be scanned. There will surely be slides and black-and-white negatives. There may also be color negatives, though there will likely be few of them. For each film type, though, there will be different specific films, and the films used should be known. (The specific name of the film can be found along the edge of a 35 mm. slide or negative, in the margin between the sprocket holes and the edge of the film. On larger-format films the name of the specific film will be found in similar locations.) Slides are likely to be mostly Kodachrome®; black-and-white films will vary more widely, though Panatomic® X is likely to be the most common for a U.S. project. Although Kodachrome is likely to be the most common slide film and Panatomic X the most common black-and-white negative film, other films will have been used and each should be noted. Similarly, every color negative film used should be noted. (A range of dates may also be recorded so that the dates during which specific films were used may be known. This is likely to be possible only with the aid of notes for negative films, but slides still in the original cardboard mounts will show the dates of processing on the mounts.)

Each color film, whether slide (positive) or negative film, exhibits fairly standard long-term behavior in terms of color shifts and fading. Knowing the film type and age will help to determine the adjustments needed. This is not the place to discuss the ways in which such film degradation can be corrected in the scanning process or with software after scanning, but there are publications and web-supplied information about corrections. Professional scanning laboratories will almost surely be best able to determine the appropriate correction factors and to make them accurately and efficiently; they will have standard algorithms for adjusting common films. (A note here about physical handling. Even cleaning film effectively requires some study when dealing with anything more problematic than dust, e.g., mold or dirt particles. Kodak® publications provide useful guidance for cleaning film.)

Black-and-white films will generally present fewer problems. While films may have faded and, in extreme cases, suffered some degradation, the likelihood that complex changes will have occurred is slim.

Numbers of slides or negatives for each specific film should be noted. Here one need not have precise numbers but a sense of whether there are tens, hundreds, or thousands of images on a given film.

As the photographs are examined for film information, some effort must be made to determine whether most, all, or few of the images will be needed in the new, digital archives. That is an extremely vague statement because the determination of numbers will take several steps. The first step will be making a general appraisal of whether or not culling large numbers of the extant photographs should be undertaken at the outset because of the number of duplicates or otherwise unwanted photographs.

If the slides are to be culled, some criteria must be established so that the choices are not random or idiosyncratic. This brings us to the question of which photographs are to be scanned and which are to be to be ignored – and how to make those distinctions.

Some of the images may be aerial photographs; many will be object photographs. For excavations, some of the site images will be rather general; others will be of particular portions of the site, possibly cleaned and prepared for photography and possibly with some special circumstance calling for the photo. Survey projects may have images of people walking the fields, of areas under study, or of concentrations of finds.

Which are the most important of the photographs in the archives? One might argue that the object photographs were the ones most carefully taken, but it may, arguably should, be possible to re-photograph any object, and one would take those photographs in color today on the assumption that eventual publication will be electronic. So perhaps the object photographs are the least important to digitize (except in the case of objects that may have been restored since first photographed or otherwise altered, lost, stolen, or . . .); that can only be argued if the objects are readily available for new photographs and if it is realistic to think that new photographs can be taken on the available budget.

The site or field photographs are the ones that truly cannot be duplicated. Similarly aerial photos should be considered of high value, though it might technically be possible to duplicate some of them.

As is so often the case, priorities must also be matched to costs. That is, the costs of digitizing specific groups of photos may help determine priorities. It may be, for instance, that many photographs are not of great archival interest but are nonetheless needed for classroom use where high-resolution scans are not required (for PowerPoint presentations, for instance) or for use on the web, another place where high-resolution images are not required. Lower-quality scans are not so expensive, whether produced in-house or by an outside vendor, and may be perfectly acceptable for such uses. In addition, if a great many slides are scanned for classroom or web use, the resulting digital images can be used very efficiently to search for those images deserving of higher-resolution, archival scans.

Indeed, scanning a great many images in a kind of production-line setting without attempting the highest quality may be an excellent way to get many photographs scanned quickly while providing images to be examined (on screen) to determine which ones warrant eventual higher-quality scans. (Note the assumption that the original slides are not discarded after scanning. The originals should be kept.) This works only if the number of images deserving high-quality scanning does not represent more than about a quarter of the total. As the number deserving high-quality scans grows, the waste from duplication grows substantially.

If scanning is being done in-house, a good scanner with a bulk loader can be purchased and used on a large number of slides at a rather modest cost. It may well be possible to set such a scanner on "auto pilot" and let it scan all slides; they can be checked for quality so that some can be re-scanned if necessary. While the results will not be comparable to those achieved from a good outside vendor, they should be good enough for most purposes – and should supply the necessary images to use for determining what photographs need better scans. That is a more appropriate approach if there is a reasonable-quality computer available so that the scanning can be done without disrupting other work.

If the archives include Ektachrome, other color slides

Scanning In-House or Using an Outside Vendor

Photo scanners have become more and more common and less and less expensive. Modern consumer-grade equipment for digitizing film can certainly provide more than adequate results for all but the most demanding uses, e.g., to make prints larger than an uncropped 8″ x 10″ photo. (This is assuming there are no significant problems with color shifting or fading.) As a result, there is a natural assumption that film, either slides or negatives, should be scanned by project personnel. On the other hand, color shifts and/or fading, if pronounced, are better dealt with by professional laboratories where the combination of expertise and superior equipment will often prove valuable and worth the extra cost.

Less pronounced problems with color shifts may be overcome with experiments on a consumer-level scanner and photo-editing software. There are sources of good information on this, and one would be well-advised to set aside enough time to read manuals carefully, follow up with other sources, and seek local expertise. Before making a final decision, it would be prudent to experiment on some slides and negatives to see if some, none, or all should be scanned by an outside vendor.

Project personnel should also examine costs with care. It may be more economical in the long run to have film scanned by an outside vendor to avoid the need to purchase a scanner, dedicate a computer to the job as necessary, train personnel to do the work, and pay for the time required. In addition, an outside vendor – a good one, meaning not the least expensive choice – should combine superior equipment and well-trained employees to produce better scans.

Each project must decide this by making an effort to compare real costs; a good rule of thumb would be to add fifteen to twenty percent to the estimated costs of time and personnel for scanning in-house because of the training and learning curves required and the likelihood that more than one person will need to be trained. In addition, the estimate must be based on a realistic appraisal of the costs of the hardware required.

that have faded over time, or color-negative films that have faded, added care will be required to "repair" the damage, as discussed above. However, corrections may create a problem for the purist. Correcting for faded images or for shifting colors means that there may be no preserved version of the photograph as artifact, the photograph without modification. This seems to me to be a time when we can relax a bit and accept the fact that the original does not really exist but has been lost to age. Therefore, trying to preserve the slide in its current condition is probably not a useful thing to attempt. It is certainly possible, though, to scan faded slides twice, once without corrections and once with (or to scan without corrections and apply those corrections after the fact, saving both versions of the image).

The biggest surprise in dealing with images is the realization – often in an oh-my-God! moment – that scanning is really the easy part. Even the slow and arduous task of deciding which photographs to preserve is not the hardest part of the job. The hardest work is recording enough information – and the right information – about the images to lead people to the images they want and need. This, of course, takes us back to data tables.

Building the data table(s) for images will, once again, leave project personnel at the mercy of what has been recorded. A thorough study of the extant records will be required to gain the level of understanding needed to permit the creation of a good set of new and useful tables. Equally important, some guidance will be necessary for users; in order to locate the images they need, users will need to know what categories of information have been recorded and what terms have been used.

The creation of the data table(s) will, as a by-product, require some consistent and reasonable directory structure for storing the digitized images. In addition, of course, archival versions will need separate care and separate storage.

I noted earlier, when discussing "born-digital" images, that the original photograph should always be preserved without modification, even if the modifications result in a much better, more clear image that becomes part of the archives as well (see Chapter VI, p. 208 f.). That is true here as well. Photo editing programs can work miracles on scanned images as well as "born-digital" ones. But the original scans should be kept as well as the improved versions (even if the "original scan" has itself been modified for color shifts or fading automatically in the scanning process). There are multiple reasons for that, the simplest being that the software may make it possible to do even more with the original a few years from now. (Is the foregoing in conflict with my earlier comment about faded images? Yes. I can live with the contradiction, but you may prefer a more consistent position.) More important, there should be no concern that images have, in the course of improvement, come to represent their subject matter improperly; saving the original should provide a ready check on that.

Miscellany

Any project may have all kinds of paper records that do not fit in the categories already discussed. Ideally, they should be scanned for preservation purposes, but, in the real world, this is a process that may or may not be justifiable on a cost basis. It will, of course, require another data table as a finding aid; so it may not be a simple, inexpensive process. On the other hand, digitizing almost everything seems a poor choice. Better all or nothing in the sense that all the data should be available in one place and one form at one time.

It should have gone without saying that all the processes, decisions, organizational selections, and so on should have been documented as the digitizing work went forward. That information will be as important for this work as the comparable information is for a project that begins with a blank slate. The documentation is truly indispensable, and it, too, must become part of the digital record so that it can be examined whenever necessary.

Digitizing Data from and On-Going Project

The following assumes that the project with its data being transformed into digital form is not only on-going but will continue with the aid of the new digital techniques and that, at the end of the day, all the data are to be integrated. If the digitized incoming data are not to be integrated with digitized versions of the paper data, the problems are fewer but the reason for such an approach escapes me. Either all data should be digital or none.

The following has also been written with the implicit assumption that all new data will be in digital form. That, of course, may not be the case, particularly if a project is large and involves multiple teams. Not only do I believe such a process to be wrong-headed, the possible combinations and permutations of a digitizing process that is only partial are so numerous that I see no virtue in trying to deal with them. I leave to the reader the task of sorting out such a project.

Assuming that the records from some point in time forward will be in digital form and that those already extant are not yet in digital form but will be at some future date, what are the issues that confront the project personnel as they begin to design the digital approach for records entering the system for the first time?

There are two critical issues that deserve attention at the outset. One is obvious: the plan for the new data files. The second is less obvious but should come as no surprise: inventories of existing records.

Let us begin with the inventory, both because it is less obvious and because it is, in many respects controlling. That is, the existing records will weigh very heavily on future plans, suggesting avenues that should and should not be followed.

The inventory should list all the materials that comprise the project's records: file cards and card systems, notebooks (field notebooks and notebooks used for other records), maps and plans, photographs (slides, negatives, and prints), and publications. If there are any materials describing/defining/documenting the records or the recording processes, they will be of truly critical value and should therefore be included in the inventory.

For each of those material types the inventory should then describe and define the storage system fully and accurately. The aim here is to provide the starting points for all the efforts to build digital data files. For card files the names of the data categories will be needed, for instance, so that data tables mimicking the card files will have the same kinds of entries. Similar examinations are required of all the data types so that, in each area, the data from new work can be meshed with the old. In each case it will be necessary to look at many different versions of the specific material; at the least very early and very late examples should be checked to be sure that changes occurring over time are noted. (Consider again an artifact card file with "residue" as a category only late in the day, as discussed above, p. 226.) There is little to be gained here by recounting the work done with a completed project. Much is the same with an on-going project. What follows, then, is a discussion of matters that are different when the project is on-going.

Data Tables

The records of contexts and artifacts are likely to be the most important data collections for a site, those of artifacts and collection areas the most important for surveys. In either case, it is necessary to start with the data categories – those corresponding to database columns – that were used in the paper recording systems to get a good foundation for building the digital data tables.

There are two issues of importance regarding the data categories for the data tables.

The first, of course, is the deciding upon the necessary data categories: artifact type, material, dimensions, etc. A database designed to be used with the paper data – whether the paper data are converted to digital form or not – should present the data using the same categories as those paper records unless there are compelling reasons to change. For instance, let us assume that project personnel have decided to use Munsell colors to describe pottery fabric, but the previous system used color names without a scientific standard. It would seem that the two sets of data would then be incompatible as to fabric color; early data would have a color name; later entries a Munsell number. The plans to switch to Munsell colors, however, need not be compromised: Munsell colors should be used in the digital data tables. However, it should be possible to include an algorithm that will generate, for each Munsell color, a color name corresponding to the names already in use; the color name can then be added to the data table(s) for each object catalogued after Munsell colors have come into use. The result is that a scholar wishing to examine the pottery from both phases of the project can use a color name to find all fabric recorded with that color name or with a Munsell number that yields that name via the algorithm. Thus, a search of all the fabrics by color name will be possible, but not a search of all fabrics by Munsell color. This approach cannot solve the problem of missing data (Munsell color) for the first-examined artifacts, but it can let all be searched for color – albeit not Munsell color.

A different issue arises if original records were made using the English measuring system for dimensions. There is no reason to use inches and fractions in a new data table. Any number can be translated for any user without needing to continue with a discarded measurement style. Any computer can manage that. So, no matter which measuring system is used, the other version of the measurement can be calculated and presented – without removing the originally-entered one from the table. Of course, the names of the columns in the table(s) should make clear which measurement is the original and which the calculation.

Practical issues may be important as well. If the project is in its last few years with a huge backlog of paper-based data, would it make sense to worry about Munsell colors at all? If only a small fraction of the material could have Munsell information, is there any point? As much as I detest compromises with the data, I also loathe wasted effort that can yield no useful results.

This should sound rather familiar. Approaching project data always involves careful planning and thoughtful examination of various possibilities. There is no "right" answer to most of these questions, though there may be many poor ones. Here the issue is the need to make a single set of data tables that will work for the existing paper records and for the new data that will arrive from ongoing work.

The second issue regarding data categories is the actual data used in each category. Just as we have dis-

Internal Consistency

The process of digitizing a set of paper records will necessarily be an effort to impose internal consistency on a set of data that is not, in fact, internally consistent because the presence of errors large and small is inevitable. When working on an on-going project the same is true, but one has the opportunity to get back to the original evidence and to determine with some objectivity where the problems lie. Such possibilities for finding the origin of errors that have crept into a set of old records do not exist. The original evidence has been destroyed. Nevertheless, internal consistency is the necessary goal.

The problem may be most apparent with CAD or GIS, since multiple drawings may not align correctly, and the drawings provide quick, visible confirmation of a mismatch. However, it is fair to say that inconsistencies will abound. Much time and effort will be spent trying to correct them.

When the "errors" that lead to inconsistencies are corrected, choices are being made. One data item is being privileged over another, or one "fact" is being accepted only because there is no available alternate, even when that "fact" seems unlikely. It is likely that the people making those choices are right; they will have been studying the data at closer range and in more detail than anyone else. However, there can be no guarantee. Therefore, the choices need to be documented with care so that another scholar at another time can dispute them. In cases such as these, when evidence has been destroyed and only the records remain, ambiguities and inconsistencies should not be "corrected" without a trace.

cussed the need to have a limited and well-defined list of potential data entries in a database created from scratch, so there must be good, reliable lists of acceptable terms for data entries in a database that both digitizes and follows from a set of paper records with incoming new data. Moreover, the terms used in the paper records should be used in all the data files. Not only is there no need to reinvent the wheel, the terms that have been used should continue so that there is no confusion. Furthermore, the terms in use will have stood the test of time. Imagine the chaos created in the long run if the term *trefoil-mouthed jug* is used in the paper records and *oenochoe* is used in the data tables. It will be tempting to use new terms in many cases, but the temptation should be avoided whenever possible. (Of course, various work-arounds permit new terms to be used while older ones are recorded as secondary entries, see "True to the Original and Today's User." Such work-arounds are better than ignoring the problem, but it is generally better to keep matters simpler by keeping the terms simple.)

There is one huge advantage to using a vocabulary that has been used over prior years of project work. The terms will have been tested over time and winnowed down – or added to – so that the critical ones are all there. However, there are likely to be confusions and contradiction within the written records; there always are, whether the records are written or digital. Indeed, when the records are on paper, it is more likely that the terms will be inconsistent; the problems created in a card file by inconsistent terminology are not so vexing. The previously mentioned *amphoriskos* label, for instance, may be used in a paper recording system very inconsistently without concern; the inconsistency creates no real problems because automated searches are not possible. As a result, the process of examining the paper records for the sake of digitizing both those paper records and new project information must include copious notes about those paper records. The notes will guide the development of all the digital records, those from old paper forms and those from new work. The project personnel will need to make many choices about vocabulary to bring coherence to an inconsistent, possibly even chaotic, list of terms that a paper-based system permits. Of course, one choice is to permit less carefully controlled terminology to continue, but that will compromise use of the data significantly. The better choice is to use one or another work-around or one-to-one table relationship to create multiple versions of data entries that will permit better searching while preserving the original data.

In the case of a long-running excavation, there may also be difficult issues involving context names. A new excavation director may use a different system for naming contexts or constructing context names from constituent elements – a new system that may fit the older system poorly. This is a critical issue since it will be very important to search in a database for contexts with some logical terms. Therefore, considerable thought must be given to such a problem at the outset; one of the possibilities might be a system of context names that provides, as with Munsell colors, two choices, one that applies to all contexts and another that applies only to later ones.

Other data tables will reflect the particular ways in which the project has been operating. In each case, though, the aim is to create digital forms of data that can mesh with the paper forms when that becomes possible. At the end of the day, all the data must be accessible in ways that assist the scholar; so early compromises that seem inconsequential may not be in the fullness of time.

CAD Files

Drawings and maps present quite different problems. If there has been a consistent set of drafting standards, it should be continued, even though many of those standards are not relevant to the use of CAD or GIS. New drawings will be viewed in conjunction with older ones; so they should not seem foreign. The same would be true of maps. This is only important if all old drawings are not to be digitized, and this is more likely – and more defensible – with maps and drawings than with data tables. That is, existing drawings and maps may not be digitized, or, more likely, they may only be digitized after some delay, making it necessary to use digital and paper-based maps and drawings together. The need to search a single corpus via a computer does not seem to rise to the same level of importance with maps and drawings.

In the case of CAD, this may also mean that layer-naming systems need to take into account the drafting standards that have been used. For instance, all material that might be drawn with broken lines of a specific type should be on layers that can easily be selected so that all lines on those layers can be given the proper line type when necessary. The simplest way to accomplish that would be to add a single character to the layer names; that character would simply indicate the line type to be used (along with weight and color, if applicable). The point here is simple: to be able to produce plans from any period of the project that look as if they belong together and that aid understanding by virtue of their similarities.

GIS Data

The most important aspect of maps to be incorporated into a GIS data set will be determining what scales have been used and what the impacts of those scales will be on other data. If all the maps used have been made at very small scale, they may not be very useful with new data. That, in turn, may either change the way new data are mapped or entered into the GIS – or encourage the project to replace the maps with new ones drawn to a better scale.

The same issue of scale will apply to any aerial images available for the site. The smaller the scale, the more limits will be placed on their usage.

Field Notebooks

Field notebooks or daybooks should be scanned. Especially for those working on project analysis, these are the critical documents for working out stratigraphy, specific contexts, problems with certain finds, and so on. Scanning is not so valuable as a way to digitize the daybooks in the sense of making them more useful or easier to search. Scanning is valuable because it is the way to make the daybooks available to all the members of the team when, if, and as they are needed. Once the scans have been made, any member of the team should be able to read any notebook at any time and in any place, no matter how many other team members are reading the same daybook at that moment. All that is required is placing the scans on a server and providing the necessary access – password protection and indexing. Of course, this implies the presence of a web site where such resources can be shared by current staff, whether or not a wider public has access.

The foregoing means that the project personnel must, at the conclusion of each season, add the new notebooks to the corpus so that it stays up-to-date. The technical aspects of the scanning and access systems remain the same as with a completed project, while the need to keep current simply adds another end-of-season to the list of chores at the conclusion of the season.

If daybooks are digital from the beginning – still unlikely in the extreme – the problems will be fewer (no need for scanning) and the access system should be far better since text searches will be possible.

Photographs

Digitizing photographs will eventually be required by virtue of the need to work on past and present records simultaneously. Working with photographs requires the full panoply of strategies used both for a terminated project and for a new one. That is, it will be necessary to approach all the old photographs as carefully and thoroughly as if the project had ended. At the same time, new photographs will surely be digital, and all the problems and issues associated with digital photographs must be dealt with. In addition, the database must take ac-

How Many Photographs and Whose Photographs?

Projects that have been going on for some time are likely to have a huge number of digital images taken by a variety of scholars associated with the work. Some of those photographs will be considered the property of the photographer/scholar, not the project despite the fact that they should be part of the project record, either because they are properly of wider interest or because they are of specific value to other project scholars – or, more likely, both.

It is very difficult to determine how best to deal with this embarrassment of riches. The project archives need not contain every photograph ever taken (arguably someone should cull the photographs on a regular basis to remove duplicates and other unnecessary photographs), but the archives should certainly contain all extant photographs that a user might want/expect, and it seems logical to say that the decision as to which photographs fit that admittedly imprecise specification should be a decision made by someone representing the project, not the photographer/scholar. That clearly implies that all photographs taken in the course of working on a project – whether an excavation or a survey – should, at the option of a project representative, become part of the project archives. I think that is the correct view despite the problems it involves. Who should choose which photographs to place in the archives? The director? A committee? Or should all photographs to be placed in the project archives?

These are not easy questions, but the issue of long-term archival preservation of the photographs argues for automatic inclusion of all so that any photograph taken of/by/for a project will become part of the project's permanent archives. Since such an automatic process assumes an enormous number of photographs, many of which would be of very limited interest, there should probably be a regular process for culling so that the number of photographs does not become overwhelming. On the other hand, it may be easier to include all without argument than to spend the time required to cull.

The idea that every scholar's personal photographs belong with the project archives yields a further requirement. All such photographs must have attached data to make cataloging/searching possible. That, of course, means that each person taking photographs must know and use all the correct terms for identifying the photographs.

Supplying not only all photographs but appropriate data for them is certainly not easy. Fortunately, newer digital cameras record a good deal of information automatically; so the burden of data recording may be much lower with the right camera.

Similarly, computer personnel will face the nasty task of dealing with incoming photographs that, despite requirements to the contrary, will arrive without the necessary data.

Any less burdensome answer to this issue involves more casual control over what is, in the end, a product of the project. Therefore, it seems to me that the best and actually the simplest approach is the all-inclusive one. Every photo should become part of the project archives, and every photo should arrive with specified data. In the end, the requirement for data will ease the burden on photographers once they have become accustomed to it, making it clear to them what must be recorded. It may also encourage them to be more careful about taking photographs casually. More important, it will result in a better, fuller, more comprehensive project archives. (Note that having all photographs in the archives is not sufficient; copyright must also belong to the project or be shared between project and photographer. See above, p. 215.)

count both old and new photographs, film and digital – and that database must identify the original medium.

Project personnel will want to know what data items have been recorded about the photos already in the records. The information recorded should not be seen as a limit on the records to come, but that information may be very instructive. For instance, if the time of day has been recorded, it will prompt project personnel to continue with that (important) data item. Similarly, if the camera lens, film, photographer, and so on have been recorded, project personnel will be more likely to insist on continuing the inclusion of those items. On the contrary, if the lens used has not been recorded in the past, it may encourage the staff to omit that item in future data tables.

There is, of course, a danger that the recording in the past will have been so minimal that the results will be a temptation to record too little in the future – or even to record nothing at all. Common sense, here is in so many areas, does not go out of style.

Conclusion

Assuming equally rigorous processes and careful planning, digitizing a completed project is more difficult then preparing the way for a new project. The work must honor the approach of past record-keeping practices while putting the data into new forms. This is not easy and may seem to force a good many compromises. For that reason, a team-based approach, with more than a single person planning and organizing the work is critical. The short-cuts that seem critical to one person may seem trivial to another and vice versa; so this is one of those classic cases where many heads are better than one. Indeed, many heads are necessary; one is inadequate. It is also work that demands a controlled, self-conscious process so that it evolves in ways that do not result in either wasted effort or inadequate results.

VIII

Protecting and Preserving the Archaeological Record

Introduction

In each of the previous chapters there has been some discussion of the need to preserve archaeological records. This is such an important aspect of work with digital technologies that it demands a fuller discussion here.

Whose Responsibility Is This?

Protecting the data from an archaeological project is a critical responsibility. As has often been said, the data are the real fruit of the work. Therefore, the responsibility lies squarely with the project director or directors. While much of the actual work may be found in the job description of the computer expert, the ultimate responsibility for the data lies with the director(s). The director must be sure that the work is being done and that the way it is being done is both technically and practically appropriate.

All the data from the project must be the responsibility of the director – or they will be nobody's responsibility. No specialist's data should be separately treated or preserved; all information belongs with the corpus of project data. Similarly, all kinds of data – databases, GIS data sets, CAD models, digital photographs, sound recordings, and any paper files – belong together as parts of a unified whole.

There are few rules to go by in this arena; so there must be careful consultation and a well-designed and thoroughly specified plan for preserving and protecting the data. Furthermore, the plan must be updated at least annually. A long-running project will see substantial changes in computer technology and in data storage techniques over the life of the project; those changes must be taken into account in the evolving plans for data preservation.

Protecting Data During the Life of the Project

During the life of a project – whether that is a long or short period – digital records need to be backed up regularly, and there needs to be a clear and explicit form of version control. Nobody should ever have to wonder which file is current and which is an out-of-date back-up.

Back-up can be done by writing files to an external disk, to CDs or DVDs, or to a network server at a remote location (in a corporate, government, or university computer center). The latter may be the safest process, but it is also the one least likely to be available from relatively remote project locations. If access to a network server is not possible or reliable, as will remain the case for most projects for a good while, the next-best choice is to backup files on CDs or DVDs. The reason for that is two-fold. One, writing to CDs or DVDs requires no extra equipment, generally speaking, since most current computers have CD/DVD burners. Two, the disks are not alterable. (This means that one should not use CD-RW or comparable DVD technology to make a re-writable disk.) At the time of this writing, CDs may be marginally preferable because they are so widely available, so inexpensive, and so universally supported. However, most projects will have DVD burners, and many will need the extra capacity of the DVD. (In either case, scholars should choose manufacturers of media that will last. Some CDs and DVDs are far less likely to suffer from known types of degradation than others.)

If work in the field involves a server to hold all files, another approach to back-up is both simpler and more robust. That is the use of a RAID (Redundant Array of Independent Disks) system. There are several versions of RAID systems, but the simplest involves two disks that are identical. Every time data are written to one, the same data are written to the other. Since large hard disks are now relatively inexpensive, a simple RAID system is a good, practical way to secure data.

The problem with using CDs or DVDs is the proliferation of copies of the data files. For each back-up there may be one or more copies on CDs or DVDs. (I would recommend two to be safe.) At what point ought those copies be discarded? Ought some copies be retained permanently? There are no certain rules for this, but I would suggest the following.

1. Each back-up copy should be labeled with the date, an indication of the meaning of the date (e.g., all data entered up to the beginning or end of that day), project name, file name(s), software used for recording the data, and the name of the person responsible for making the copy. (In addition, all paper forms from which data have been transferred to the computer after a particular backup should be stored together until the files with the data from those paper forms have been backed up.) Needless to say, each copy should be examined to be sure the data have been transferred correctly. This can be done easily with file-comparison routines.

2. Back-up copies of data files that have been superseded and are no longer needed should be physically destroyed so that they cannot be mistaken for current files. The destruction should take place immediately after newer copies have been made and examined. Those being kept as the last of a particular organizational structure – e.g., before adding a new column to a data table or before a change to the layer-naming system used in a CAD model – should be marked as carefully as possible so that they will not be used again by mistake. They should be stored in some location separate from the data in continuing use.

3. The initial copy of a database should be kept until the project has been completed and archived. That set of files will be empty or nearly so but will show the state of the database system at the very outset of the project. It will be the most simple and complete expression of the original data organization.

4. The last copy of a database, CAD model, or GIS data set before any organizational change should be kept until the project has been completed and archived. It should be labeled with the date and the nature of the change(s) that made the files obsolete.

5. At the end of a season copies of all files should be made in triplicate. The three copies should be carried back to the home institution(s) separately, and the copies should be separately stored so that they cannot be adversely affected by the same phenomenon (e.g., fire, theft, water damage).

6. If files are altered during the off-season, back-up copies may be made on a network server or on CDs or DVDs. (See below.)

7. When returning to the field, two copies of the current files should be taken to the field on CDs (separately from the computers with the same files, ready for use). One copy should remain at the home institution until the end of the season, and that copy of the files should be destroyed when the new season has been completed and the new data brought back (unless they need to be kept in accordance with the requirements in 3, above).

The frequency of back-ups will depend on the speed with which paper records are processed. At the end of a data-entry session when the number of processed paper records reaches some pre-defined number, a back-up process should be initiated. The timing will be determined on the practical basis of the amount of work the project can risk having to do over again. That is, if every day's work is backed up on CDs or DVDs, only the number of paper records processed in a day might need to be entered over again, but that schedule will use up a large number of CDs or DVDs in short order. If, on the other hand, data files are backed up only every week, fewer disks will be needed, and more time will be spent re-entering data in the event of an accidental loss. This is not a theoretical question but a practical one that will be answered differently by each excavator. (There are ways to enter data that can make backing up files much easier. A system that places all new data in temporary files for insertion into the full system at a later date can be backed up by keeping copies of the temporary files rather than the base files. Backup files are then much smaller. The problem is that the processes used to make and insert temporary files are more difficult and time-consuming. They also require someone in the field to be present and responsible for inserting the temporary files into the base files at the appropriate times. Nevertheless, there is much to recommend a system based on the use of temporary files.)

The most difficult problems with data integrity and control often occur in the off-season. If scholars are working on any of the data files during the off-season, it will be difficult to keep track of new versions or to control the proliferation of versions. One solution here is to force all off-season work to be done through the computer center and its servers at the home institution of the project director(s). That, of course, requires considerable consultation with the personnel at the computer center, and that, in turn, may require some changes in plans based upon the use of personal computer software in the field. The extra effort will be well worth the trouble, however, if data are to be altered in the off-season.

The other solution is to force all additions and alterations to be done on personal files and transferred to the base system only by the project computer specialist before the beginning of the next season. This is a time-consuming process, but it is efficient if there are not many scholars making changes in the off season. If team members simply want access to the data during the off-season, they can be given CDs or DVDs with the data. Their use presents no problems for long-term data storage.

Both approaches to data added by individual scholars working away from the project and the director start from the same assumption: the director is ultimately responsible for the data and for all aspects of data organization, preservation, and presentation. No member of the team should be in a position to add data to the system, edit data already in the system, or, worst of all, alter the organization of the system in an individualized fashion. Otherwise, data integrity is at risk, and that integrity is arguably the single most important characteristic of the data that are produced by an archaeological project. The care and concern lavished on the planning stages would be useless if individual scholars were allowed to alter things according to their own perceived needs after the system had been created.

It is more difficult to give general advice about digital sound recordings, digital photographs, and video in digital form. The best way to save and back-up such files is to store them on CDs or DVDs in groups. As the files of a particular type become numerous enough to fill a single disk., each disk will contain only files of a given type. Those back-up files become the archival files as well when the CDs or DVDs have been filled, meaning that multiple copies must be made. The decisions about timing and procedures must be made early on and with care, and they are ones that must be guided by common sense. But these decisions must be guided also by the understanding that the original files, before editing, must be part of the archival repository.

What About the Paper Records?

Caring for the paper records that have been digitized in the normal course of project work, e.g., locus or lot forms, also requires a process that is well-planned and well-monitored. Each time files are backed up, the paper records associated with the new back-up copy should be stored and marked to indicate the version of the data files with which they are associated. Only then, after the digital files have been copied to a secure source, is it safe to remove the paper records to a new location where access may be more difficult.

It may be better to scan paper forms than to store the paper. That is a choice to be left to the individual project, but retaining only scanned copies of paper records is certainly an acceptable procedure. Discarding the paper records after they have been scanned should then present no risks. Of course, the scans of the documents must be indexed and cataloged so that anyone can refer to them. (Since they are stored only as a precaution, the scanned documents should be stored on CDs or DVDs, with a simple catalog of contents included on each disk and a full table of contents stored as a part of the database information. The scans of the documents need not be stored with the other data.) Redundant copies must be kept to prevent loss. In the case of scans of documents, of course, regular re-writing of CDs is unnecessary. Once a CD or DVD has been filled, that disk and its copies need only be copied when their useful life is nearing an end.

Implicit in the foregoing is the need to keep careful records of the back-up processes. There must be a clear trail of information about all changes and back-up processes, and it must be readily available, whether in the form of a computer file or a paper document.

Archiving Archaeological Information

Making regular back-up copies of data files is critical, but that is not the same as making permanent copies that will be available in perpetuity, archival files.

Paper data, photographs, recordings, videos, films, and digital data of all kinds are alike in one critical respect. To the extent that any of these forms of records contains original information from an archaeological project, those records are irreplaceable. The process of excavating or collecting objects from field survey leaves the scholar with a changed world; the beginning conditions and the changes wrought by archaeological processes are only known if the records of excavation or survey activity have been carefully made and are properly preserved. It does not matter whether the information was recorded with a computer or with a chisel on stone tablets; we must know as much as possible about the original conditions and the changes wrought by archaeological processes if we are to understand the most basic archaeological information. The records tell us about context and about the objects, features, structures, and relationships that make the artifact record meaningful.

Sadly, archaeologists have not been exemplary stewards of their records in the past. An archivist once said to me that the typical archival material left behind by a retiring archaeologist was crumpled paper in the waste basket that had been left to prop open the office door as they scholar left. That may be an exaggeration; however, many institutions are not equipped to deal with voluminous records from an excavation or survey, and few scholars feel compelled to spend their valuable time preparing records for archival preservation. There is not a strong tradition in archaeology for those interested in digital records to build upon.

There is, however, a compelling reason for those of us interested in digital records to pay special attention to the requirements of digital archiving. If a set of paper records is damaged or partially destroyed, the loss can be incalculable, but it may not be total. That is, the portion of the paper records preserved does provide some information, even if it is a small part of the whole. In the case of digital files, the loss can be very different. Even slight damage to a digital file can be fatal, rendering the loss total. At best, expensive outside expertise would be needed to retrieve data. That being the case, our obligation to preserve the data for future generations imposes greater burdens upon us in the digital world than it did in a paper-based world.

Preserving Digital Information

The key problem for preserving digital information lies in the nature of the digital data file. Files are nothing but sequences of binary numbers – one's and zero's. The leap from a series of one's and zero's to real information is not a leap of faith but a very precisely specified series of translations from those one's and zero's to instructions, letters, and numbers. Each file type is constructed – encoded – according to a set of instructions unique to that file type and decoded according to a complementary set of instructions.

The encoding and decoding processes demand not only that the processes be coordinated so that the coding and decoding together take a user from one batch of information expressed in a given program to the identical information expressed again in that program but also that the file in question remain pristine from the time it is encoded until the time it is decoded. Removing a single one or zero from the beginning of such a file may render it useless by making the decoding instructions operate on the file from the wrong beginning point. Such a seemingly trivial mistake is not trivial. To the contrary, it can be fatal.

There are thus two requirements for preserving digital data. First, the files must be preserved precisely, without change. A change comparable to the turned-down corner of a page or the holes left by a removed staple can become a fatal flaw. Second, the encoding/decoding schema must be preserved so that the file can be reconstituted; that, of course, is the responsibility either of the manufacturer of the software used to create the files or – in the case of non-proprietary formats – of the broader community of computers users. Neither of these requirements is so onerous as to present a serious challenge, but care is necessary.

File Compression or Double Encoding

There is one potential problem with digital preservation that is often overlooked. As mentioned in Chapter II (p. 43), computer users often compress files so that they require less disk space and/or so that they can be more easily sent over a network. There are specific programs designed to compress files according to one or more of several compression algorithms. Compressing files adds a second encoding/decoding process and subjects the files to a secondary risk of damage. Therefore, as a general rule, no files to be used in an archaeological project should be compressed. The foregoing was carefully phrased to leave open the possibility that files being sent to someone outside the project and therefore not to be used further in the project might be compressed for transmission convenience. Otherwise, compressing files adds a risk that simply does not have enough rewards to be justified.

There is one exception to the prohibition against the use of compressed files. One of the most common compression techniques is the JPEG compression algorithm used for images. Many digital cameras use that format as the standard way to encode images at the time the photograph is taken, despite the fact that it is a "lossy" format, sacrificing some fidelity to the original when compressing files. In addition, programs designed to display and manipulate digital photos can directly read and write JPEG files. (By comparison, database files, CAD files, GIS files, and even text files are normally compressed only by external programs after they have been saved in the original format. They must then be decompressed before they can be used by the appropriate program.) Finally, JPEG files are used on the Web. As a result, the JPEG format has become a common one. Indeed, if the original image has been stored as a JPEG image at the outset, there is no value in changing the format later; the losses cannot be restored.[1]

Unusually important photographs – a phrase that can only be defined by an individual scholar considering a particular photo – should not be taken and stored in JPEG format. Either a "raw" file (a camera manufacturer's unmodified output, sometimes requiring the manufacturer's software for manipulations) or a TIFF format should be used for such photos. (It could be argued that the "raw" formats used are unlikely to remain in use for long, changing along with camera hardware, making conversion to TIFF, a stable standard, desirable.) Those files should then be the archival versions of the photos. Modification should be done without changing to the JPEG format, though photos for specific purposes might be saved in the JPEG format. The key is to be certain that any photograph can be traced directly back to the uncompressed original, with no compression between the original and the final version except, when necessary for a specific purpose, for the very last saving of the image.

---

[1] If JPEG images are manipulated and re-saved, there can be additional loss of fidelity to the original. The saving process will, each time the image is saved, re-compress the file. In the long run, therefore, the image may deteriorate noticeably if it is altered and saved many times. As a result, JPEG images that may be altered should be opened in the photo editing program and then saved in that program's native format until all changes have been made. Only then should the images be saved in JPEG format again. The original JPEG file should be kept as the archival original, and the altered version should have a new name to distinguish the two versions of the image.

Long-Term Preservation Adds Problems

The proverbial fly in the ointment with digital preservation is the regularity with which commercial file formats used by commercial software – the encoding and decoding schema – change over time. Even the same computer program will produce files according to new coding specifications every few years. The changes are not made to be perverse, though it can sometimes seem so, but to add capabilities. The result is a need to keep files in contemporary formats to make it possible to use them even when the original file formats have become obsolete. (Non-proprietary formats change less frequently, and they are more likely to be replaced entirely by new formats that have been generated by a standards organization and bring substantial, not incremental, change.)

Two avenues for keeping digital files current have been suggested by the computer cognoscenti. One path is based on the assumption that the information in computer files can be presented properly only with the original software. This may, in fact, be quite true for certain kinds of digital data, especially data used by graphic-intensive programs that present images with state-of-the-art effects. As the state of the art advances, images or video presentations can become glaringly out-of-date. To deal with this problem of data presentation, some have recommended that archival preservation should include the preservation of computer systems so that a secondary user will be able to duplicate the experience of the original user many years later. Knowing that such long-term preservation of hardware is impossible except for extremely specialized institutions, those who take this view have argued that the best approach is to develop programs to emulate old software and hardware in current computers so that old data can be presented on new computers as if they were being presented on the original machines. Thus, a Windows computer in 2050 should be able to emulate a 1988 MS-DOS 4.0 machine with an Intel 80386 processor, and a MAC running OS X, version 20 in 2050 should be able to emulate a 1988 MAC running System 6.x with a Motorola 68030 processor.

While the most dramatic effects of emulation would be seen in graphics applications where advances are more visually apparent, there is another, less evident area of impact. Many scholars who collect data in digital forms include special features in their implementation of the data collection/presentation processes; those special features are generally activated by scripts and macros. For instance, the older propylon CAD model referred to in the CAD chapter includes a series of scripts to display specific phases of the building's history, with or without reconstructions. Similarly, implementations of complex databases regularly include routines to display certain selections or sub-selections of data that the scholars in charge deem to be important. Such scripts and macros should continue to perform properly in an emulation environment.

This path for retaining digital data in useful forms requires emulators for any and all operating systems and computers used for collecting archaeological data, and each new change in current software and hardware will require new emulators. Hardware for reading older media may also be required. However, this approach does not require that anything be done with data files other than storing them safely, making certain that they remain in pristine condition. While that is not a trivial task, it is not difficult. It requires vigilance, not brilliance.

The serious down-side to emulation is the complexity it brings to the job of accessing old data files. A typical user would need the emulation software (perhaps several layers of it, with a machine running Windows 2050 emulating a machine running Windows 2040 emulating a machine running Windows 2030 emulating a MAC running . . .) and many versions of programs for database, CAD, and GIS data. Of course, that "typical" user would also need the ability to use all those old operating systems and the programs running on them. For instance, that mythical 2050 user of a Windows machine running the then-current version of Windows might need to know how to use a dozen versions of Windows (each of which would be emulated by programs running inside the then-current version of Windows) and many versions of FileMaker and Access and their successors, at least one version of each program per major revision of Windows – not to mention various CAD and GIS programs. (The foregoing assumes that such a user does not need to access data developed on a MAC or a Linux machine, for which further complexity would be required.) This complexity creates a major problem that its proponents have not adequately addressed: it will discourage people from using data for which they have inadequate software or too little experience with the software in question. Over time, the number of people with inadequate software or experience would become virtually the entire spectrum of practicing archaeologists.

The second approach to long-term preservation of data files assumes that the information content is the critical issue and that the problem of data presentation is not so severe as to require emulation. Therefore, those who favor this approach recommend that data files be regularly altered as their formats become obsolete. That is, files written with software that has evolved and consequently writes data in a new format, should be re-written in that new format. This sounds more difficult than it is, since all commercial programs that require changing file formats for a new version include automatic update procedures. In those rare instances where one program is replaced by a competitor, there are also likely to be file-upgrade routines because a widely-used program will not be replaced unless potential users can bring along their old data.

Using this process of file alteration – called data or file migration – has its own negative side-effects. The macros and scripts used in files will continue to work in new versions of those files only until there are major changes in the software. They may work for a time (as do all the scripts written for AutoCAD mentioned above – for now), but there will come a time when the changes in the program or files render them obsolete. In addition, the data files are unlikely to be available in every format users might want. CAD files, for instance, may only be available in AutoCAD's format, and database files might be available in only FileMaker, Access, or MySQL formats. Those difficulties with multiple formats for similar data require that data files be stored, when possible, in non-proprietary formats that can be imported into various proprietary formats – as if we were to store scholarly text only in Esperanto on the assumption that translation into any language is possible when required. Non-proprietary formats, of course, require discarding the scripts macros, and other special features of the software used to create the data files.

Actual archival processes differ from place to place, but, in general, archives that use file migration to maintain access to data will accept files in a variety of formats but will want most standard formats to be augmented by non-proprietary versions of the same files. Otherwise, they will migrate the files into non-proprietary formats on receipt. For instance, a DOC (MS Word) file containing the expository text concerning a project might be accompanied by an RTF file (another MS format but one publicly specified and therefore accessible via virtually all word processors) with the same data. Similarly, database files might be submitted in FileMaker, Access, or MySQL formats – but also in the non-proprietary delimited ASCII format; AutoCAD files might also be submitted in the DXF format. When possible, the non-proprietary files will be the ones sent to users since they allow a user to access the data with virtually any software. As I write, this area is changing rapidly with the introduction of generalized open-standard file formats.

The use of non-proprietary formats has some significant drawbacks. In some instances features may be lost. For instance, a DOC file using multiple authors' revisions and maintaining those differences in the final copy could not be duplicated in the RTF format. Database files will create more common problems. In the typical database application the links connecting files to one another are implicit in the data storage system, but the most neutral format for data tables used today – delimited ASCII files – will not maintain those links. As a result, full documentation is required to specify things that might otherwise be assumed. In fact, of course, all those specifications should be explicit anyway, but the use of non-proprietary formats requires more – and more careful – documentation to be sure that all bases have been covered. (The documentation of the older propylon files ran to 16 single-spaced pages, and that was a very small project.)

Creating Non-Proprietary Files

Because of the potential pitfalls that can accompany migration of data to non-proprietary files, it is preferable for those new files to be made by project personnel rather than archival personnel. Project personnel are much more likely to notice any problems in the first instance and to be able to evaluate their importance in the second. For instance, when I archived the model of the older propylon I did not create a DXF version of the model because I did not believe that the DXF version would be adequate. The archival repository may disagree and make a DXF version, and I cannot (and would not) try to prevent that, but I chose not to make such a file because I thought it would be inadequate.

On the other hand, I did make (and check) tab-delimited ASCII files from the data tables. There were no complications with the tables, which were not large to begin with. I also made an RTF version of the text documentation. In fact, there seemed to be no reason to archive a DOC version.

Avoiding Migration Pitfalls

Knowing that data or file migration is the preferred way to preserve data files for future access, scholars should avoid using file formats that present obvious migration problems. For instance, some modern database files can contain data in other formats, images have been the most common but other formats are often included now as well. The database files therefore contain other files (in their own native formats) within them. (See BLObs, Chapter III, p. 95.) Such files present a kind of migration Hell. For each file type within the database file there is a potential migration need, and for the file as a whole there is a separate migration process. Such complex file formats, therefore, present a host of problems and complicate the migration process unnecessarily; their use should be avoided.

CAD programs often have similar problems when database information is attached. Some CAD programs include their own database processes; others link to external databases of particular types. In both cases the potential problems with data migration are significant, and in the case of external data tables, my experience has been that stability of such links from version to version of the CAD program cannot be ensured. The risks here outweigh the rewards.

For GIS programs the problems arise more frequently with secondary files produced in the analytic process. Keeping track of them requires considerable care and forethought.

Data from Outside Sources

Data may be gathered from outside sources – commercial, government, or web-based sources. Those data files will rarely be – from a legal point of view – part of the project data for which the project may take responsibility. The project will not own them or have the right to archive them as part of the project data set. Nevertheless, potential users of the project data must have access to the files and information about them. Therefore, you will need to document: their data structures, any modifications required by the project, the dates of the files actually used, the file formats, and, of course, explicit information about the sources of the files, including office and/or web addresses. It will be important to be as specific as possible about the versions of such sources actually used since change over time in the data files could yield different analytic results. A major issue is permitting replication of data analyses so that future scholars can have confidence in your work; so this documentation is critical.

Metadata: Resource Discovery and Documentation

Data to be archived requires what is commonly called metadata, information about the data to make it possible for others to use the data. Metadata actually covers three kinds of information, distinctions that are too seldom honored. The first may be called resource discovery data, the information about the files that might be found in a library catalog and that would induce a scholar or student to want to access the data for use. These are the data needed to learn about and locate a resource. The second may be called documentation, the information needed by a user to put the data files to use. The third may be called management data, the information needed by a repository to manage the files. Since the management information is not the concern of the scholar, either as user or as depositor, the need for management data will be the province of archival repositories. We must deal with the other kinds of metadata: resource discovery and documentation data.

In actual practice, the archival repository will doubtless present project personnel with lists of terms/questions to help provide both resource discovery and documentation forms of metadata, but the needs of the repository should not be the only ones an excavator or survey director considers. Anyone working on an important project should have his/her own notions of the important categories of information for resource discovery, and, as has been discussed *ad nauseam*, documentation cannot wait until the end of a project.

There are various standards for resource discovery metadata, and data depositors will be required by the depository to provide – or help provide – the information. However, the project director should insist upon the inclusion of those types of information he/she deems important, whether they are included in the standard categories or not. The project director is more likely to know the important search categories than are the archival personnel. A scholar working in the Maya world may have quite different notions of important resource discovery information from a scholar working in south Asia, and the archivist may not full appreciate the needs of either specialist.

Documentation metadata for database, CAD, and GIS files have already been discussed, but there must also be general documentation that attempts to provide what any user of the records would need to understand and use the files – before turning to the actual files, no matter the type. Potential users of the material need to know about the project as a whole – its aims, size, scope, and so on. They also need to know about the planning for digital records, not simply that digital recording was to be used, but what kinds of digital records, related to one another in what ways, and used for what specific purposes. What technologies were planned from the beginning, added as the project progressed, added after the field work had concluded? Most of these questions should have asked and answered before the project began, but some will have to do with unpredictable working processes.

Potential users, for instance, should be alerted to problems with data input if there were any. They should know if there were difficulties of any kind that might affect the reliability or accuracy of the data.

This is also the place to critique the plans made at the beginning. No project can be perfect, no matter how carefully planned and executed. There will have been glitches. If honestly described, repeating them will be less likely, as will erroneous interpretations.

In the final analysis, the scholars' aims are to make the data accessible, to provide whatever information is required to assure access for other scholars. Archiving the data without such information serves no real purpose.

Archaeological Data Repositories

There are not many repositories for archaeological data in digital form – institutions that will attempt to preserve the data in perpetuity. European countries, by and large, have the advantage of more leverage – both carrots and sticks – to have data treated properly; so the Archaeology Data Service (ads.ahds.ac.uk) in Britain is a good example of the kind of archival assistance that should be available.

In the U.S. such repositories are neither as well utilized nor as broad in terms of their holdings. The Archaeological Research Institute at Arizona State University (archaeology.asu.edu/) is now the only general archaeological digital repository in the U.S. known to the author. Two other attempts at creating broad archives were not sustainable. There are other archival repositories that accept more limited ranges of documents, and many institutions will attempt to archive data from their own projects.

Similar to archival respositories, new web sites are being created to gather data from many projects in order to provide online access to large aggregations of data at a single web site. These resources provide access to information from many projects with a single query over the web. Some, but not all, of these projects will actively archive their data. Few, if any, will archive the data from any specific project separately from the data derived from other projects. Few, if any, will accept all the data files that you may need to place in an archival respository. As a result, if you choose to use such a web resource in place of a simple archival repository, you should be certain to check as to the archival procedures in place.

When to Archive Data

Project data should be archived as soon as possible. There is a simple, practical reason for that. The longer the period from data recording to data archiving, the more likely it is that important issues concerning data organization, reliability, and ultimate utility will have been forgotten. Furthermore, waiting makes it more likely that the full data system will languish in an almost-complete state for years; whereas archiving the data will force the director(s) and computer experts to finish the tasks of cleaning up files, performing the last data checks, removing duplicates, and so on.

Speedy archiving may mean that the project director(s) will need to negotiate with the archives regarding access to the data. To the extent that access to the data should be limited – a proposition some of us would dispute at its very core – archives will permit that for a period of time, but a limited period.

Preserving a Project Web Site

Development of web sites is explicitly not a subject of this book. However, so many projects will have web sites that some comments about their long-term preservation are required.[2]

A web site provides the public face of a project; so the content of a web site should be preserved as part of the record. General statements of purpose, annual reports of many kinds, often informal, and a great many other documents will have been used on a web site over time. Not only should they be preserved; the state of the web site at any given moment should be clear and explicit.

That means that every version of every document should be preserved along with the dates of availability on the web. (Corrections of simple spelling or grammatical errors need not be separately preserved unless the error might have had an impact on meaning.) In addition, the structure of the web site at any given point in time should also be carefully defined. While it is not necessary to preserve the

---

[2] As projects deem it more and more necessary to have their own web sites, some have decided that it is both appropriate and valuable to use the web site to provide access to project data as the data are recorded. As ideal as that may be, doing the work required to keep access to the project data as complete via the web as it is on site is expensive, time-consuming, and of questionable value. Few projects have sufficient resources to cover the cost of such work, and no project known to the author has managed to start down this path and live up to its own expectations. Putting new data up on the web as the project operates has great appeal but questionable real value, and doing so is very costly.

fully functioning site at each stage of its existence, defining the organization and preserving the documents would permit any scholar to know what information, analyses, and references were being put forth at any moment in the web site's history. That, in turn, would help any scholar understand better the intellectual history of a project, its impact, and its influence.

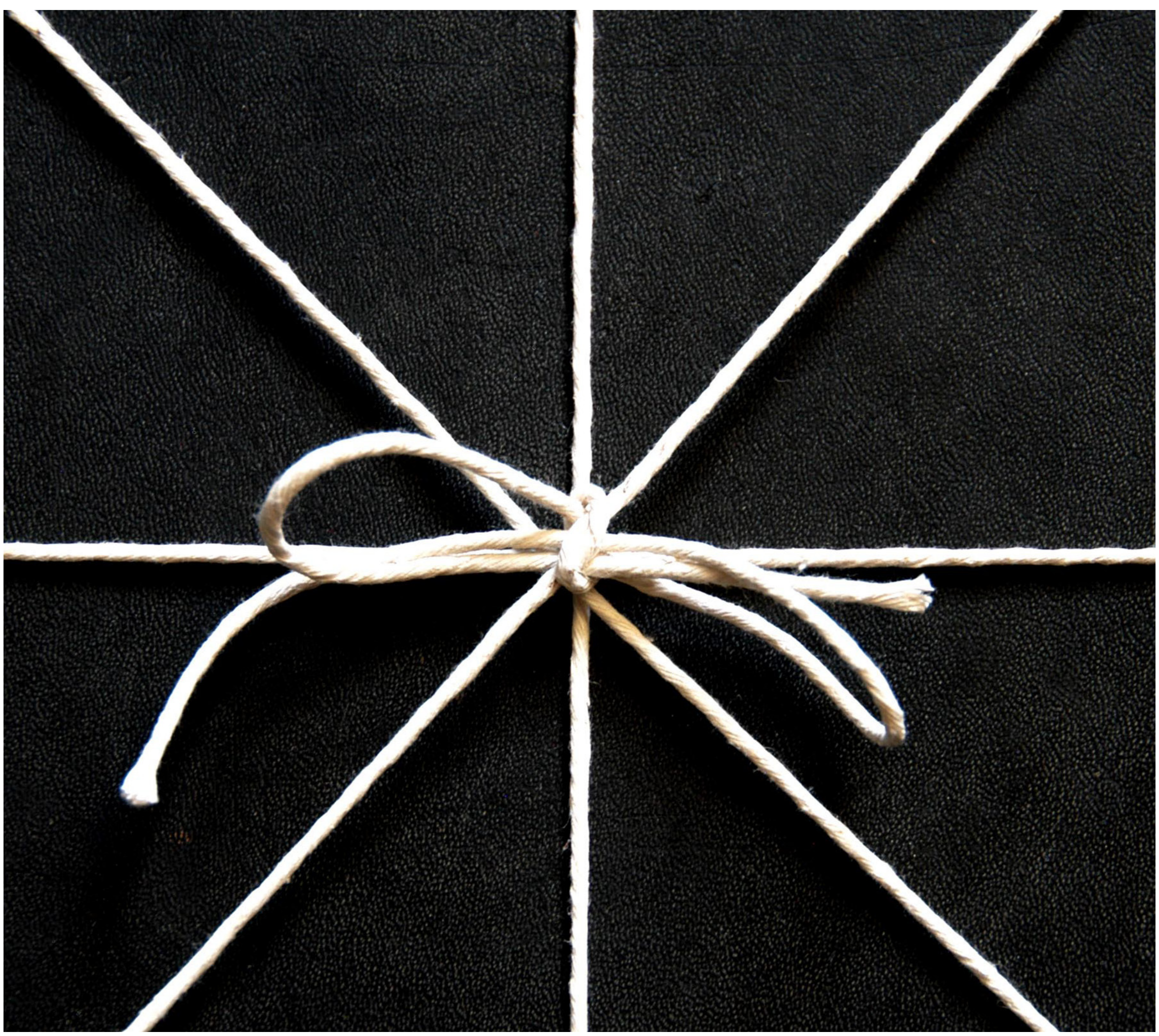
In many cases – e.g., photographs or maps – the materials on the web site will be materials preserved already as part of the project data set. Such materials do not need to be separately preserved again as parts of the web site, but unambiguous designations of those materials, their dates of availability, and their positions in the site organization are required. To the extent that basic database, CAD, or GIS files are part of the web site, they do not need to be saved a second time as part of the web site; however, to the extent possible, the nature of those files as they changed over time should be stated and explained. That is, a scholar looking at the web site at a specific point in time should be able to determine the contemporary condition of the database, the CAD model, or the GIS data set. Since so much of the data will have been tagged with dates on entry, that is not as hard as it may seem.

Conclusion

When discussing the problems of elicit excavations and artifacts that make it onto the art market, archaeologists bemoan the loss of information that results from removing the objects from their contexts. In the long run, losing control of the information about objects, context, and excavation procedures can be just as damaging. We owe it to the scholars of the future to protect and preserve the information as well as the objects. We owe it to the people who fund our work. And we owe it to the people whose lives we are trying to illuminate.

IX

Conclusion



Introduction

This book alone will not prepare anyone to go into the field and construct a database, a CAD model, or a GIS data set. Nor can a reader commence digitizing records from an older project. It should, however, prepare a reader to ask the right questions and to think about important issues of data structure and documentation in ways that will inform databases, GIS data sets, and CAD models. Those contemplating digitizing records from an old project should have some good ideas about where to begin and how to proceed. It should also help with choosing appropriate software.

Some added information should be useful. Most of these topics have been touched on previously in passing, but they deserve to be brought out again and/or more directly.

Timing

Waiting too long to involve the computing personnel in project planning is a common and potentially disastrous mistake in the preparation of an archaeological project, and the importance of early participation of computer experts was noted in the discussion of digitizing information from old projects. There is so much involved in planning the computer aspects of a project – even a project of moderate size – that those responsible for the computing should be involved from the very beginning. At the simplest level, they must plan for the hardware that will be needed. They will select the computers, operating systems, and peripherals that must work together as a seamless whole, and, absent planning in this area, scholars tend simply to use the equipment and software they have, whether it is the best for the job or not.

There is another problem with waiting to add those responsible for computing to the team. If they arrive late to the work, they will not only be rushed; they will be less able to discuss their options with the project director(s) and among themselves if there are multiple computer experts. The director(s) of computing must be able to examine the possibilities for data organization and storage with the project director(s) because there will be many options and no perfect choices; rather there will be many desirable ones competing for selection. A slow and deliberative process will produce benefits in the end and permit time for making better choices and for creating, at the end of the day, a coherent and unified data set.

That deliberative process will be especially important if the project is complex. A holistic approach to the data to be collected cannot be imposed after the fact but requires an iterative process of making choices, examining outcomes, adjusting choices, and so on.

Choosing the software and preparing the organization of the data is a crucial part of the project planning. The organization of the data must reflect the excavation or survey methodology, but it may also, as noted earlier, provide crucial checks for the project directors concerning the way they will approach record keeping. The scholar responsible for computing should be a part of the team from day one. Note the use of the word scholar. The person responsible for archaeological computing should be an archaeologist who knows computers, not a computer expert who has been treated to a short course on archaeology. In the real world we all inhabit, that may be impossible, but an archaeologist must be the "point person" for the use of computers on any project, whether new work or digitizing data from an older project. That scholar may direct a technical person or, in the case of more complex projects, a technical staff; however, the scholar must be the one making the important choices.

Only salvage projects have a real need to begin work without the luxury of some advance planning. Therefore, those undertaking salvage projects should have some processes and systems at the ready so that they are not caught unprepared.

General Approach

This book has assumed the use of personal computers – Windows machines, MACs, or Linux computers. There is, however, a world of assistance awaiting many scholars in their government, corporate, or university computing centers, most of it relating to more complex solutions that are also more robust. Those solutions rely upon network access to data and often upon the use of network-based servers as the real data repositories. In general, archaeologists do not have sufficient access to high-speed Internet connections in the field to make full use of the facilities at the computer center from afar. Nevertheless, the possibilities should be explored at the outset. If the project will be able to maintain high-speed communications with the home institution at all times, the use of the computer center should be considered. This, of course, is especially relevant to those working on the digitizing of information from old projects.

Using networked servers to access project data during the off-season (as discussed earlier) may provide a different reason for using computer center facilities. This possibility should also be explored.

Any use of computer center facilities should be explored carefully before the project software has been chosen since software choices might be affected by a decision to offer off-season access through the computer center.

Project-Wide Planning

This text has been written, of necessity, as if the three critical technologies discussed could operate independently. They can, but a good excavation or survey plan approaches the data to be gathered in a holistic way. The critical question is not how to use database management systems or CAD or GIS; it is how to gather, store, analyze, and preserve all relevant data. Therefore, from the very beginning the planning must assume that all digital data are interconnected. That, in turn, means than all planning must be comprehensive. Only after the larger questions have been settled can the smaller ones involving specific programs – entry systems, software choices, etc. – be approached.

An integrated approach demands consideration at the very beginning of the project of such mundane things as systems of labeling that permit the same item to be labeled in the database, CAD model, and GIS data set so that information about it can be retrieved. For example, a feature excavated in a trench must be given an unambiguous ID number of some sort, generally one determined by the excavator's numbering system. If that feature is to appear in a database, a CAD model, and a GIS data set, that label must be useful – and equally unambiguous – in each, or there must be some certain and explicit way to link all occurrences of information about the feature.

Similarly, data originally incorporated in GIS and CAD systems must be treated in ways that permit the data to be used as, where, when needed. The goal is not a good CAD or GIS system. The goal is data available when and where needed.

This integration must even extend to one of the technologies barely discussed here – digital photography. All digital photos must be labeled in such a way that they can be found according to various criteria. That should mean, as discussed already, that data tables will be required to index the photos; that linkage between photos and their subjects should be considered before the first photo is taken. The indexing needs will virtually always turn out to be more complex than expected.

Database to GIS Linkage

It is hardly necessary to discuss the linkage between database information and GIS files. Most of the GIS files are, after all, database files with attribute data. Nevertheless, it is crucial to consider carefully the content of the files and the relationships between/among them to be certain that the file structure permits asking (more important, answering) the kinds of questions that will be at issue. For instance, if it must be possible to see on a map all the cooking pots found in a certain context, the file structure must be designed for that.

Database to CAD Linkage

Linkage between database information and CAD data is a bit more difficult, and it has been discussed already in the CAD chapter. As with GIS-Database linkage, plans must be made early to make possible the kinds of links and connections desired.

CAD to GIS Linkage

Linking CAD and GIS elements is much more difficult, and some duplication is inevitable. This is especially true if the GIS is raster-based.

It is not as common to use GIS and CAD on the same project as to use either technology with database management systems, and ordinarily either CAD or GIS is the prior technology on the project. For instance, an excavation recorded with CAD system may spawn a survey as a secondary project, employing GIS technology there; whereas a survey project using GIS software might result in an excavation within its boundaries where CAD software would be required. The priority of one or another software package will impact the coordination between the two. If GIS software is used first, the likelihood is that objects will be recorded in ways that permit displaying their positions in a GIS setting; if CAD software is used first, on the other hand, it is likely that object positions will be included in the CAD model and that CAD layer names – and naming systems – will be involved. As a practical matter, then, one or the other software approach will tend to dominate. Regardless of which one is dominant, fitting GIS and CAD together requires careful consideration of CAD layer names and GIS layer/theme designations and data table design. In addition, there should be a clear division of labor. It may not be necessary to display object locations in both programs, for example. To the extent that there is duplication, however, it should be carefully coordinated so the entities are not entered twice but transferred from one system to another with as little human intervention as possible. In addition, there must be some checks on editing procedures to make sure that any entity changed in one form of data access shows as adjusted in the other.

In the long run, it is likely that CAD and GIS software will merge, and some movement toward that merger has already taken place. In the interim, though, it is up to users to make sure that the two program types can be used together well. (It must also be admitted that the prediction of this merger was first made years ago, and the merger has still not happened.)

CAD vs. GIS

Until CAD and GIS software have merged, if that does happen, there is a necessary debate about which kind of software to use when – and why.

In the recent past, the choice has too often been made for the wrong reason, that being the software with which someone on the project is already familiar. This decision is too important for such a factor to be the determining one. The differences between the two types of software are not trivial, and those differences can significantly impact their utility. The specific needs of any project must guide this choice, as has so often been said about many issues here.

GIS software excels at bringing data from disparate sources – including public data sets – into a complex matrix of information and permitting analysis with the benefit of all the data and data types included. GIS is also required when notions of topology must be inherent in the system, not added by the way data are organized.

Given the nature of the real world we work in, GIS is also required when a large portion of that world is to be encompassed in a single project. The discrepancies between the real world and a Cartesian grid are too great if the coverage is large. Of course, the cut-off between a size requiring GIS and one acceptable for CAD is not clear or absolute. It is more likely that a project will have extensible boundaries and need GIS because of the potential for expansion than that the project will begin with a coverage area obviously too large for CAD.

CAD software, when properly used with careful layer control, is superior when 3D models are needed. Good layer management can provide excellent access to non-graphic data, including attached databases. However, the data one may use in a CAD setting is limited to the data gathered and entered for the project and by project personnel. Using public data of any kind is possible but will generally require substantial extra effort.

CAD software can only be used where a Cartesian grid system is adequate, that is, where the coverage area is not too large.

It has been said here often but bears repeating in this context: advance planning is the key to making this decision wisely. If the data gathering and recording practices for a project are designed well in advance of the actual work by people who understand archaeology and computing, the odds on the choices being good ones are much better.

Personnel

The person responsible for project computing should be, first and foremost, an archaeologist. That person obviously must have considerable experience with computers and computing, but his/her first allegiance must be to the discipline and the information, not the computers. He/she must also be able to deal with people effectively, not just machines. In the end, using computers is as much about training, understanding, and comfort as it is about hitting the right keys. If the people using the computers feel as if they are just machines themselves, there will surely be problems.

The director of computing should not operate alone. This is not just a matter of having someone prepared to take over if necessary. It is a matter of having a forum for debate as problems arise. In the best of circumstances, the director of computing will be able to debate solutions with the project director. In any case, there must be someone with whom problems and potential solutions can be debated. As there are no right answers in computing, there are many bad ones. Open debate and discussion will reduce the number of bad ones chosen. As already suggested, this is especially important with digitizing old projects because it can seem that the simplest approach is to set different people loose on individual pieces of the whole without first making sure what the whole will be.

There is a natural tendency to assume that everyone today is computer-literate. Most academics are proficient with email, word processors, and the Web, and more of them are becoming familiar with presentation programs every day. Few, however, have taken the time to become adept – not simply familiar but adept – with the technologies we have been discussing. As a result, a project team is likely to include many individuals who are not particularly

Staff Training

Training staff members is a step that can too easily be by-passed. If people are uncomfortable with data entry or edit procedures, they are more likely to make mistakes. They are also far more likely to feel uncertain and to proceed slowly and tentatively as a result. Training sessions should be numerous; everyone should be as comfortable as possible.

The training sessions should also give everyone a chance to comment on procedures, and those comments should be carefully weighed. If criticisms are simply ignored, the result will be undesirable at best. More important, thoughtful criticisms will lead to better designs. Similarly, data entry experience should help guide planning and inform adjustments.

In the case of old projects digitized after the fact, there may be more training needed – and there may also be needs for training people to assist those who simply will not learn to use computers well enough to access data on their own. Such people cannot simply be frozen out because they are unable to gain the necessary comfort level with computers.

well-prepared for what awaits them. They will be able to fill in forms or otherwise follow instructions, but they may or may not be prepared to make themselves part of a team in terms of helping to make the system work well – and evolve to something better.

Therefore, the project director and the person responsible for the computer work must be ready to explain well and carefully the aims and procedures used. Everyone should be familiar with as many computer jobs as possible, and all should have a sense of ownership that will encourage them to report problems and suggest solutions. The computer expert will spend as much – probably more – time working with the project personnel who are using the computers as with the computers alone.

One of the most creative bits of database design I ever accomplished was done in response to comments from a project participant who did not think a data entry procedure I had designed properly reflected the way the excavation itself had been planned. Those who sorted the pottery lots were expected to go beyond the paper forms with which they began by editing stylistic designation as they deemed necessary. My original data entry system did not permit that. The project participant who had tested my system was right in her assessment, and it was up to me to make an adjustment. It took some time to develop a process that always started with the right terms and permitted additions or alterations, but the end result was better for all, and, in the final analysis, did not take so much of my time as to be a problem. Had that person simply viewed herself as an automaton with no avenue for input into the final result, the data entry process would have remained less well connected to the paper field forms and field personnel practices. By the same token, had I thought I could ignore her concerns, the system would also have remained less useful.

Personnel Records

There should be at least one data table for personnel records. Not only should the basic information about staff members, e.g., seasons of working and contact information, be there, individual personnel should be connected to information recorded by them via the database – whether in the field or in the lab. Knowing who entered what data can provide important information for the project team, from knowing to accept one person's pottery typologies to knowing to cast a skeptical eye on another's. Most useful will likely be the ability to examine terminological inconsistencies within the team (but finding such inconsistencies will require a project member to examine data on a regular basis, something definitely to be recommended). Project personnel may feel that having their names connected to data they have entered implicitly puts them under a cloud of suspicion; so it must be made clear that the point is not to find people doing things wrong but to be sure that the information is full and complete, that systematic errors are found, and that terminological questions are made evident so they can be addressed. Attaching names to interpretive data also makes it possible for someone examining the data to know with whom to speak about a difference of opinion. Under no circumstances should there be an attempt to fool project personnel by denying that their names are connected to their data entries or field records.

There will likely be project participants at the extremes – either so computer-averse that they really don't want to have anything to do with any computer or so savvy that they want to do things on their own. Dealing with the former may require tact and care, but dealing with the latter can be more demanding and present potentially greater problems. If those most adept decide to add capabilities they deem necessary, they can wreak havoc either by making unauthorized changes or by adding their own separate bits of data in ways that deprive everyone else of the added information.

The site architect must also be brought into the project early if CAD is to be used. The CAD work will require advance planning and careful consultation among architect, director(s) and computer specialist(s). The most obvious need is for a good, well-designed system of layer names, but drafting conventions should also be well-defined in advance.

Maintaining Consistency

Terminological consistency, label consistency, and operational consistency are important to the success of a project. If some pots are called coarse ware and others cooking ware and if that is a difference without a distinction for the site, the database will be difficult to use effectively. The same, of course, can be true for terminological problems in any part of the data record, and a good deal of "ink" has already been used on this issue. There is no good way to prevent the problem completely since permitting only terms from a pre-determined list may deprive the person entering information from using the proper term if it has not been put on the list. Free entry, on the other hand, may lead to terminological hell. The only truly effective and systematic approach I am aware of was described in the Side Bar, "Limiting Data Entry Choices," Chapter III, p. 88.[1] That discussion, however, ignored the difficulties of multi-lingual projects, treated briefly elsewhere (Chapter III, p. 99.)

Consistency of weight and measurement labels is much easier to maintain and hardly need be mentioned. This is not an issue changed by the use of digital technologies. Simply stated, such labels should not be part of the data entry process. All units should be explicitly stated on forms and in data documentation. Thus, weight in pounds or kilograms, ounces or grams should not be the choice of the person entering the information but the choice of the person responsible for organizing the data. The same would be true for distances and artifact dimensions. Units of measure are parts of the record-keeping manual and should never need to be stated in the data tables. Even using a C.E. or A.D. label should not be the choice of the data entry person. A check box should be used to indicate whether or not a date falls in the common era or a negative sign used for all dates B.C./B.C.E. Such careful plans to avoid potential inconsistency are not new to the field, though some of the ways of dealing with them may change a bit with the move to computers. (If technical analyses yield B.P. dates, of course, there must be a different solution, also carefully crafted in advance.)

Operational consistency is a term I have used to cover more than one area. First, there is the issue of specificity of typologies. As previously mentioned, it is difficult to enforce consistency as to the level of detail for typological identifications, particularly when the people entering data may not be experts on the artifacts in question. A pot labeled Late Helladic II B 1 by one person might be called Late Helladic by another who is less willing to be so specific. The issue here is not error but the ability and willingness to make fine discriminations, something that cannot be expected of all in the field. There are at least two approaches to this problem, assuming that a full complement of artifact experts cannot be kept at the site at all times. The first is simply to ignore the problem for field data entry and assume that the final study of the pottery (or other artifact type) will include correct and properly specific typological identifications. That is probably a good approach, but it does assume a relatively short time between artifact recovery and specialist examination – or a long period during which the data can only be used with great care. The other approach is to make typological data entries at least consistent, if not specific, so that some kinds of searches can safely be carried out in advance of the final study of the artifact type. Data entry personnel should not use terms that are ambiguous (e.g. LH II A/B) and for which nobody would know to search but should provide as much specificity as possible – and document doubts or qualifications in another column just for such purposes.

Operational consistency also refers to the way project participants gather and enter data. These kinds of consistency issues are not new for the excavator gathering pottery or other artifacts in lots defined by excavation procedures, but they may be new for technical experts such as surveyors who are gathering data for a

1 Needless to say, terms that are needed in the course of a season should be added without delay. The process described is for routinizing the procedure for keeping the required terms both up-to-date and limited to those truly necessary.

CAD or GIS system. In the CAD chapter there was discussion of data generalization, the inevitable need to take fewer data points than might be needed to define a wall or a feature fully. Similarly, the issue of generalization in GIS data gathering procedures have been shown to be critical. Questions about data gathering and density should not be left to individual project members who may change from time to time. There should be standards that can be expressed clearly and understood easily so that the data taken in year one for a GIS map of a water course or for a CAD model of a hearth are comparable to data taken for another water course or another hearth five years later. In short, the manuals used for excavation procedures need to be expanded to include some of these data-gathering and data-entry procedures that are specific to CAD and GIS.

Changing Data

There will be mistakes in data entry. There will also be changes that come not from mistakes but from confusion, changes of opinion, expert opinions over-riding field personnel, and other sources. Many scholars treat such changes as simply alterations of the record that need leave no trace. Of course, a paper-based system of card catalogs retained the old with the new in most instances, since a correction would not obliterate a prior entry.

It is not particularly difficult in a computer database to maintain a trail showing corrected data, and doing so is good practice. It may not be necessary to provide elaborate procedures that permit people to gain access to the trail of previous entries at the push of a button, but keeping that trail in a shadow table is a good way to make sure that minority opinions and mistakes are not permanently removed from the record. Those errors may turn out to be of interest; indeed, they may turn out not to have been errors at all.

Some scholars may question the need for this care with corrected data, but it is an important part of archaeologists' respect of scientific method that such records be kept. Only with the aid of such records can we be sure that questions about facts and interpretations can be approached openly. (That's why we use ink instead of pencils so often in the field, after all.)

The procedures used to permit tracking changes are likely to be the same procedures used to permit contradictory opinions to live together in a database or a CAD model. Therefore, preparations for correcting and modifying any data set should be made from the beginning, and they should permit both corrections of the record and disputes that are not resolved.

Software Choices

Scholars tend to use the software that is supported by their home institution. That is a natural choice – and not a bad one. There are, however, other considerations, and they should be weighed along with institutional support. For instance, obscure programs, no matter how superior, will inevitably present problems. The problems may be in the form of finding personnel familiar with the programs, integrating data from other sources, or archiving the data files; it is necessary to make an effort to be sure those problems have been identified – and their importance examined – before such programs are put to use.

If support by the home institution is very broad, then there are many issues to be considered when choosing from a group of acceptable alternates.

1. Hardware platform. If the database system you choose runs only under Windows, that is very limiting. If it runs only under Linux, the limits are much more significant. Does that mean you should choose only a database (or CAD or GIS system) that runs on all the major platforms – Windows, Mac, Linux? In a word, no. But you should try to find programs that run on more than one of those platforms, if possible. In some cases that is not possible. AutoCAD and ArcView, for instance, run only under Windows. Much as you or I may believe that to be

unfortunate, it does not change the programs' advantages and may make them the choice regardless of the limitations imposed by single-platform software. On the other hand, Access has few advantages over FileMaker, which runs on Windows and MAC machines. Neither has a significant advantage in terms of capabilities over MySQL, which runs on Windows, MACs, and Linux machines. But MySQL is more difficult to use since, for all practical purposes, it requires using a programming language to make it work well in an archaeological environment. So what is the trade-off? MySQL software will run on all three platforms but is more difficult to set up – and different to set up on each platform. FileMaker runs on MACs and Windows machines; Access on Windows only. The differences are minimal. In this case, the choice – for me at least – is only between FileMaker and MySQL. Access is not in the running.

As I write this on a MAC – with a Windows PC humming on my left – I may curse the darkness that I consider Windows to be, but I must use the machines that do the work. So must you. It would be irresponsible to permit a preference for a particular operating system – whether Windows, MAC, or Linux – to force you to use inferior software. Happily, things are changing rapidly. The newer Intel-based MACs can now run Windows as fast as a PC, and there is new software in development to permit machines to emulate one another more widely. This is a moving target. MACs in particular have been obliged to adjust; it is now relatively simple to run either Windows or Linux on a MAC, and I am now routinely doing for the sake of using AutoCAD.

A different possibility, though one only available to a relatively small sub-set of archaeologists, is that of using university, government, or corporate resources to run software on network-based servers and to access data from the field over the Internet. As noted above, this can only be accomplished if the field operations are being carried out where good broadband access to the Internet is available and reliable, and that will be rare for the field archaeologist. However, that can mean that the most complex computer work is done by well-trained personnel in a computing center. Although no excavation should be too dependent upon outside personnel such as those from a "back office," it may be prudent to use the skills from such a source, provided that project personnel retain control.

The foregoing is not meant to imply that you should have multiple computers of different types running on a project. The point is avoid the limitations on access to data imposed by any software or hardware choices that reduce the possibilities for others to use the data. If your program runs only on Windows, then MAC or Linux users are automatically left out, regardless of the software they own, unless the data files can also be used by software that does run on their machines.

This discussion is also not meant to suggest that any particular platform be used. For a variety of reasons, Windows is the default choice, but it is a necessary choice only if the project will be using 3D CAD (for 2D usage, AutoCAD's advantages are less critical) or a GIS that requires Windows. For databases, 2D CAD, and GIS, Linux and the MAC may be equally good choices. Because of their relative lack of popularity, Linux and MAC computers tend to be less vulnerable to virus attacks and similar mischief; some would argue that UNIX-based operating systems such as Linux and the current MAC OS are actually less vulnerable, but it may be more accurate to say that they do not provide the same numbers of computers to damage and have therefore not been so frequently targeted.

2. Software capabilities. It should go without saying that the software chosen must be able to perform the tasks at hand. However, it is common for the needs to expand as a project goes forward; so software should be carefully examined to be sure that it can meet the needs of the project in five years' time, not just for the first season. Colleagues are especially valuable in this part of the selection process, and insightful reviews of virtually all software can be found on the web. If you cannot find colleagues who are using the software you are considering, it would be prudent to try to find out why.

Such considerations as network access during the off-season should be considered when looking at software features. If it would be convenient to permit colleagues to access and query data during the off-season, that should be a priority in software selection – and it may again drive you to cooperate more closely with your computer center.

Also among software capabilities must be the potential for the programs chosen to work together when necessary. This is particularly an issue for GIS. Since databases lie at the heart of GIS data, using the same database system for both the data tables and the GIS data tables makes good sense.

3. Price. It is all well and good to act as if price is not relevant, but it is. So make sure what the long-term costs will be. For software there are several questions. How often do upgrades come out? How expensive are they? How many copies of the software will you need? Is it likely that all will need to be upgraded with each new release? Can you use your institution's site license? Can you buy an academic version at a lower price?

The questions for hardware are different but equally important. What is the frequency-of-repair record of the brands under consideration? Are the machines consistent as to components so that the repair record is meaningful? What about turn-around time for repairs? The web can be a valuable resource for these issues as well.

4. Personnel skills. If you have people on the project already familiar with Access, for instance, it may be worth considering that program, despite its other shortcomings. On the other hand, if you have project participants who are familiar with a program that seems unsuitable, do not let that blind you to the program's shortcomings.

5. Wide use. While your choice cannot be made on the basis of others' choices, it is important to weigh the popularity of the software you are considering. If a particular program is widely used, there are real advantages that should not be ignored – more training opportunities; more third-party training books; more people to offer help, perhaps even macros or scripts that can be very helpful; possibly better support, though that is not a given; users' groups. It is also possible that there are advantages that only become apparent after considerable experience with the particular program; so checking with serious users can be a very good idea. Most important, a very widely used program provides a ready audience for your data. If many people are able to use a given program well, most of those will be able to use it with your data as well as their own. (Beware of equating program sales with program use. Access is sold so widely that one would think that it to be truly popular. However, so many people own Access by accident, as part of MS Office, that its real use is not even close to the number of copies in circulation. By that kind of measure NotePad, the mini-word-processor included with Windows is probably the world's most popular word-processing program, but few people actually use it – fewer for anything significant.)

6. Ease of use. This is not as important as one might think because the more complex features of database, CAD, and GIS programs will generally be used by specialists for whom ease of use is less important than specific program features. The ease of data entry is important, but that has more to do with the computer specialist's skills than the program's inherent capabilities.

7. Customer support. This should be important, but it is less important than it once was because the likelihood of serious problems has declined over the years. In addition, good support from a software manufacturer is virtually a thing of the past. Real human beings seem no longer to be involved at most computer companies –at least not in their telephone trees.

8. Stability. A program such as Word or AutoCAD that has been around long enough to be fully developed is not likely to change as often or as significantly as a newcomer. By comparison, FileMaker was regularly changing and adding

features as it evolved from relatively humble beginnings. More stable programs generally mean fewer problems and upgrades.

The stability of the company is also an issue. If you are considering a new program from a company that has no track record, there is an obvious risk.

9. Data portability. No matter the software, it must have some way to export data in forms that permit other programs to import that data – without fear of loss or damage. That may seem a given, and it should be; however, new programs arriving on store shelves often can import data from existing ones (they must do so to get any customers) but not export (not needed at the outset). The user who has important data in any given format is captive if the data cannot be moved to another format. Generally speaking the export format is – and should be – a non-proprietary one. (Oddly enough, the important non-proprietary format for word-processing document, RTF, is so standard that virtually all word processors can save files in that format directly, but few users bother.)

Documentation

One last time. Document the data files and collection procedures AS YOU GO. The documentation will be better, and it will be an easier process as well.

Ambition

Another caution. It is very easy to look around at the capabilities made available by modern software and then decide to do something very exciting because you can. Making video or audio recordings, for instance, has become remarkably easy. The fact that it is easy to make the recordings does not mean that either technology is necessarily useful in the field. Saying that, I do not mean to imply that either technology is without value. However, the benefits and the costs of any recording system should be weighed carefully before starting down the road. An example closer to home is the use of 3D CAD. For how many projects is the addition of the third dimension warranted by the gains in information, given the added costs, personnel needs, and time?

Equally to be weighed with caution is the web site. A web site can be a critical form of outreach and publication for a project. It can also be a sinkhole for time, money, and intellectual effort. For instance, some projects have attempted to keep online databases up to date for direct access from the web as work progresses, but I know of no project that has succeeded in that task for more than a season or two. As the project grows in complexity and the finds grow in volume, the complications brought to bear by the effort to maintain a current web site expand exponentially. The result is inevitably disappointment at least and discord within the project over resource allocation at worst.

Living in the Real World

The cautions and recommendations made here are all appropriate, but any archaeological project will have limits in time, money, personnel, or . . . . Therefore, there will be times when the recommendations made here must be set aside. When that is necessary, it is critical that the choices be made self-consciously. Trade-offs may be necessary, but they should never put data at risk, and they should be made with a full and explicit understanding of the consequences.

Conclusion

All the advice in the world will be of little use to the archaeologist headed for the field if the planning does not begin soon enough. Similarly, advice is of little value without a commitment to provide those resources – both financial and intellectual – required for the work. In the end, computers used wisely will assist with the recording, preservation, and access to archaeological project data. Used unwisely, the computer will consume time and money without providing much benefit in return.

Even if a project must begin without the required lead time for planning the care and feeding of its data recording systems, it is possible to set aside time during and after the first season to fit the recording system to the needs of the project more correctly, but that can happen only if the director makes it happen. For any project the field director is the only person who can make sure technology is used well.

As it is important to plan well, so it is critical to review decisions and to revise plans when required. Changes that require retraining of personnel should be undertaken with reluctance, but no scholar should start down the digital path without an understanding that rather frequent changes of plan will be needed.

Index

D

E

F

G

H

R

S

T

U

V

W

X