

Classics and the Computer: An End of the History

Many non-classicists from academia and beyond still express surprise that classicists have been aggressively integrating computerized tools into their field for a generation. The study of Greco-Roman antiquity is, however, a data intensive enterprise. Classicists have for thousands of years been developing lexica, encyclopedias, commentaries, critical editions and other elements of scholarly infrastructure that are best suited to an electronic environment. Classicists have placed great emphasis on systematic knowledge management and engineering. The adoption of electronic methods thus reflects a very old impulse within the field of classics. The paper knowledge base on Greco-Roman antiquity is immense and well-organized; Classicists, for example, established standard, persistent citations schemes for most major authors, thus allowing us to convert 19th century reference tools into useful electronic databases. Classicists are thus well prepared to exploit emerging digital systems. For many classicists, electronic media are exciting not (only) because they are new and exciting but because they allow us to pursue more effectively intellectual avenues than had been feasible with paper. While many of us compare the impact of print and of new electronic media, classicists can see the impact of both revolutions upon the twenty five hundred year history of their field. Electronic media thus allow us to pursue both our deepest and most firmly established scholarly values and challenge us to rethink every aspect of our field.

The title of this piece alludes to a 1993 article, "Classics and the Computer: the History," in which Theodore Brunner, the founder of the *Thesaurus Linguae Graecae* (TLG), described the development of computing in the classics up to the end of the 1980s. Those wishing an account of the first generation of computer based work in classics will find the story well documented in Brunner's work. This piece takes a broader, more schematic and more tendentious approach. To some extent, this approach is reactive: the rise of the personal computer and, more recently, of the World Wide Web has diffused computation throughout the daily life of scholars and especially of students. A Foucauldian scholar might compare the shift from a few very specialized projects and extremely expensive hardware to the ubiquitous e-mail viewers, web browsers, word processing systems, bibliographic database systems etc., to the

shift from the spectacle of the sovereign to the diffused microphysics of modern power. New computer based projects continue to emerge – the December 2002 “Ancient Studies – New Technology” conference at Rutgers will host presentations on roughly thirty computer based projects about the ancient world. Perhaps more significant, even superficially traditional publications distributed even now in traditional format (where paper-only publication or as electronic files that mimic paper presentation) depend upon an infrastructure that is almost entirely electronic. Digital systems have quietly become the norm. A follow-on to Brunner’s history would lead far beyond the growing but much more tractable body of work that Brunner confronted more than a decade ago.

The title of this piece reflects an argument as well as a defensive strategy. There should not be a history of classics and the computer, for the needs of classicists are simply not so distinctive as to warrant a separate “classical informatics.” Disciplinary specialists learning the strengths and weaknesses have, in the author’s experience, a strong tendency to exaggerate the extent to which their problems are unique and to call for a specialized, domain specific infrastructure and approach. Our colleagues in the biological sciences have been able to establish bioinformatics as a vigorous new field – but the biologists can bring to bear thousands of times more resources than can classicists. A tiny field such as classics, operating at the margins of the humanities cannot afford a distinctive and autonomous history of their own. For classicists to make successful use of information technology, they must insinuate themselves within larger groups, making allies of other disciplines and sharing infrastructure. For classicists, the question is not whether they can create a classical informatics but whether such broad rubrics as computational humanities, computing in the humanities, cultural informatics and so on, are sufficient or whether they should more aggressively strive to situate themselves within an informatics that covers the academy as a whole. Even academic technology, strictly defined, may be too limiting, since the most revolutionary information technology for twenty-first century philologists may well emerge from military funded programs such as TIDES (Translingual Information Detection Extraction Summarization), supported by the US military.

But much as some of us may struggle to minimize the future history of classical computing as a separate movement, classicists have had to forge their own history. Even in the

best of futures, where classicists customize general tools and share a rich infrastructure with larger disciplines, classicists will have to struggle mightily for their voices to be heard so that emerging systems and standards meet their needs. This article seeks to explain why classicists needed to create this separate history in the past and to ground current trends in a larger historical context. It is relatively easy to pass judgments on the past, but in using the past to argue for trends in the future this paper justifies any unfairness in its own perfect hindsight by exposing its own predictions for the future.

The following broad movements – which overlap with one another and all continue to the present – provide one way of looking at the history of classical computing. First, beginning with Father Roberto Busa's concordance of Aquinas' Latin writing in the late 1940s, standard machines were used to produce particular projects (especially concordances such as Packard's *Livy Concordance*). Second, beginning at roughly 1970, large, fairly centralized efforts took shape that sought to address major issues of classical computing infrastructure. These projects included David Packard's Ibycus system, the TLG, the Database of Classical Bibliography, the Bryn Mawr Classical Review, the Duke Data Bank of Documentary Papyri, and the Perseus Project. Third, beginning in the middle 1980s, the rise of the personal computer distributed computing throughout classics and indeed the humanities. Fourth, beginning in the mid-1990s, the rise of the Web spurred a vast outpouring of smaller projects in Classics as in other fields.

The following section takes one particular point in time twenty years ago (1982-1983). The selection reflects the situation that existed when the author of this piece began his own work, but focusing on this point in time provides a brief case study with the particular illustrating more general conditions. The subsequent section provides a schematic view of classical computing. The conclusion briefly suggests how trends rooted in the past may carry us forward into the future.

Cincinnati 1983: Retrospectives and Prospectives

In the American Philological Association convention in 1983, held at Cincinnati, a crowded session considered the merits of two very different approaches to classical computing. I was part of a group describing a Unix based approach that we had begun developing at the

Harvard Classics Department in the summer of 1982. David Packard, who was then, and remains today, arguably the most significant figure in classical computing, was prepared to explain the rationale for his own Ibycus system, which he had spent years developing from the ground up to serve the needs of classicists. Then a graduate student and very conscious of my junior position, I was very nervous – if there was a tradition of young entrepreneurial leadership in technology, I had certainly not encountered it as a classicist. The underlying issues remain interesting topics of debate: how far do we follow generic standards and at what point do the benefits of specialization justify departures from broader practice?

Looking back after almost twenty years, we could argue that the positions we espoused had prevailed. The Ibycus minicomputer system is long gone and its successor, the Ibycus Scholarly Computer PC system, is no longer under development (although a few venerable Ibycus SCs continue to serve dedicated users to this day). The benefits which my colleagues and I placed on standards have, to some extent, proven themselves: the 10,000 lines of source code, written in the C programming language under Unix, which provided an efficient searching environment for Greek and other languages, still compiles and can run on any Unix system (including Linux and OS X) – it would not be possible to buy today a computer that was less powerful than the fastest systems to which we had access twenty years ago. The *Thesaurus Linguae Graecae* itself now uses a Unix server to provide the core string searching operations on which both Packard and my group were working twenty years ago.

Such a triumphalist retrospective upon the past would, however, be unjust and inaccurate. First, systems are only a means to an end. David Packard opened his presentation in December 1983 with an observation that put our discussions into perspective and that has shaped my own decision making ever since. He observed that software and systems were ephemeral but that primary sources such as well structured, cleanly entered source texts, were objects of enduring value. In fact, the TLG had released a core set of Greek authors in machine-readable form and our work concentrated above all on providing text searching facilities for this collection. Word processing and typesetting were major advantages of the new technology and the decrease in per page typesetting costs helped justify the considerable expense of any systems – Unix, Ibycus or other – at the time.

The TLG had begun work more than a decade before in 1972. The TLG began creating its digital library of classical Greek source texts by using the standard tools available from the UC Irvine computer center, but the problems of entering, formatting and verifying Greek texts were very different from those of the number crunching experiments and administrative databases for which those tools were developed. Packard's Ibycus system provided an environment far more suited to their needs than anything else available. Packard had gone so far as to modify the microcode of the Hewlett Packard minicomputer to enhance the speed of text searching. He created an entire environment, from the operating system up through a high level programming language, aimed initially at serving the needs of classicists. Decades later, the boldness and achievement of creating this system seems only greater to this observer. It would have been immensely more difficult for the TLG – and many other smaller projects (such as the Duke Databank of Documentary Papyri and Robert Kraft's Septuagint Project) – to have been as successful. Without the Ibycus environment, few departments would have been able to justify the use of computers in the seventies or early eighties. Nor might the field have been able to provide the National Endowment to the Humanities funders with the same level of support for the TLG.

The early 1980s represented a tipping point, for at that time new systems were emerging that would provide inexpensive and, even more important, relatively stable platforms for long-term development. The Unix operating system, C programming language, and similar resources provided tools that were independent of any one commercial vendor and that have continued to evolve ever since. MS DOS and the IBM PC appeared in 1981 – before most of our current undergraduates were born. The Macintosh (which has now built its current operating system on a Berkeley Unix base) appeared in 1984 – at the same time as many students who entered American universities as freshmen in 2002. New languages such as Java and Perl have emerged, and web browsers have provoked a substantially new model of cross-platform interaction, but tools developed under Unix twenty years ago can still run. They may be abandoned but only if better tools have emerged – not because the systems in which they were created are gone and they need to be, at the least, reconstituted.

But even if the early 1980s represented the beginning of a new phase in the evolution of

digital technology, the gap between our needs as classicists and the infrastructure at hand remained immense. The labor needed to adapt existing infrastructure to our needs was not so daunting as that which faced David Packard in building the Ibycus, but it was substantial.

Two classes of problems faced us, and they continue to face us even now. The first, to which I will return, consisted of tangible problems that we needed to solve. The second, however, arose from the gap in understanding between those of us who were new to the technology and our technological colleagues who were innocent of classics. Because we were unable to communicate our real needs, we made serious initial errors, investing heavily in tools that seemed suitable but proved, on closer examination, to be fundamentally unable to do what we needed. While many of the problems that we faced then have resolved themselves, the general problem remains: the biggest government funders of academic technology are the National Institutes of Health and the National Science Foundation whose aggregate funding (\$20 billion and \$5 billion respectively) exceeds that of the National Endowment for the Humanities (\$135 million requested for 2003) by a factor of 185. The Academic Technology specialists in higher education surely devote at least 1% of their time to the humanities, but the staggering disparity in support – governmental and private – for science, technology and medicine means that the humanities are trapped at the margins of decision making. If our needs require substantial added investment beyond those of our colleagues outside the humanities (not to mention classics in particular), we will have great difficulties. We must be proactive and influence the shape of information technology as early as possible, tirelessly exploring common ground with larger disciplines and taking responsibility for pointing out where our challenges do, in fact, overlap with those of our colleagues from beyond the humanities.

Our lack of sophistication, which became clear when Harvard began its own computing project in the summer of 1982, had one advantage. If we had been more knowledgeable, the department probably would not have moved forward but would have waited for technology to advance further. Instead, the department invested so many resources that we could not easily pull back. In the end, our work on the specialized problems of classical typesetting helped lower the publication costs of Harvard Studies in Classical Philology and the Loeb Classical Library, thus providing a justification for the initial investment. But if the results were ultimately

satisfactory, the process was deeply flawed and we were profoundly lucky to enjoy as much success as we did. Many other projects in classis and throughout the academy have faced similar problems and not been so fortunate.

Several months of intensive work gave us a clearer idea of the challenges that we faced in 1982. I offer the following as a representative survey to document the state of the art at the time.

Computer power and storage:

In 1965, Gordon E. Moore observed that the number of transistors that could be stored per unit area had been doubling since the transistor was invented and he argued that the trend would continue for the foreseeable future. The pace has slowed a bit – density has doubled every year and a half – but the exponential change has continued. Thus, twenty years ago we felt ourselves to control staggering computational resources and looked upon the previous years with satisfaction. Now, of course, the subsequent twenty years have made our initial systems appear primitive. The Psychology Department's Digital Equipment Corporation PDB 11/44 on which we did our work had far less computational power than the smallest desktop machine now available but served dozens of users typing in manuscripts or conducting experiments. For us as classicists, disk storage was a crucial issue. Modern disks allowed us to imagine keeping vast libraries – dozens of megabytes – of text online for casual searching and analysis. The Psychology lab at the time had two 80-megabyte hard drives, each the size of a small washing machine. The Harvard Classics Department needed more storage to mount the TLG and purchased the largest machine then generally available. The Control Data Corporation disk held 660 megabytes – four times the storage of both disks already installed. It cost \$34,000 (including our educational discount) and had a service contract of \$4,000/year. The disk arrived in a crate that we had to pry open. We needed a special purpose disk controller (\$2,000). Worst of all, we had to write a software driver to mount this disk, hacking the source code to the Berkeley Unix system. It took months before we could exploit more than a tiny fraction of the disk storage on this heavy, loud, expensive device. Our colleagues in the Ibycus world chose different hardware (at the time, the largest Ibycus systems had, if I recall correctly, 400

megabyte drives) but the basic parameters were the same for all of us. Disk storage was cumbersome and expensive. As I write this, the smallest disk that I can find contains 20 gigabytes (30 times as much as our CDC behemoth) and costs \$200 (150 times less). The price/performance ratio has thus increased by a factor of c. 45,000. This does not even consider the fact that this 20 gigabyte drive plugs directly into a computer without modification and that it fits in a notebook.

Machines have grown so fast and inexpensive that it is perhaps difficult for most of us to imagine the extent to which hardware constrained the way we designed systems and thus the questions that we could pursue. The extraordinarily high cost of disk storage meant that Packard chose not to create indices for the TLG. All searches read through the texts from start to finish. Packard modified the microcode of the HP minicomputer to increase the search speed, thus providing another reason to build an entirely new operating system. Limitations on storage meant that digital images of any kind were impractical. The first TLG texts that we received at Harvard were in a compressed format that reduced storage by c. 20%. We had to write a program to decompress the files – the program was short but we had to write one ourselves as none existed and this simply added to the overhead of working with the TLG documents.

Greek Display:

The graphical displays which we now take for granted were not in standard circulation. Displays were monochrome terminals that could display letters but could not display drawings, much less color. Even displaying textual data posed major technical barriers. The TLG had already developed an excellent ASCII encoding scheme for classical Greek and we had no scientific reason to use anything other than BETA Code, but our colleagues insisted that they needed to see fully accented classical Greek. We thus had to devote substantial energy to the font problem – some of us who worked on Greek fonts in the period still view “font” as the one most disturbing four letter word in English.

To display Greek, we needed to use special terminals that could display customized character sets. We designed Greek fonts on graph paper, converted the dot patterns into hexadecimal codes, programmed the data onto chips and then physically inserted these chips

into the displays. The monitors that we used cost \$1,700 each and provided users with shared access to the overtaxed minicomputers in the psychology department.

Networks:

The Internet was still tiny, connecting a few key research institutions over a relatively slow network. When we first began work in 1982, we had access to no machine-to-machine networking other than log-ins via dial-up modems. To provide our colleagues access to the machine in the William James building, we purchased massive spools with thousands of feet of twisted pair cable and then ran this cable through a network of steam tunnels and conduits to Widener Library and other buildings. We did have e-mail within our single machine but we only gained access to inter-machine e-mail when we joined a network called UUCP. At the time, the machine would effectively grind to a halt every time it processed mail. Nor was mail a very effective tool, since we knew very few people outside of our own group who had e-mail accounts. Simple file transfers were beyond us and the current Internet would have seemed outlandish: experience near the bleeding of technology tends to generate a schizophrenic attitude, alternating between the visionary and the cynical.

Multilingual Text Editing:

David Packard developed an editor for the Ibycus that could manage Greek and English. When we first considered working with Unix our best consultant suggested that it would take twenty minutes to modify the source code for the standard Unix text editor (Vi) to handle Greek. In fact, the Unix text editor assumed an ASCII character set and would have required a complete rewrite to manage any other character sets. We spent a good deal of time trying to develop a multilingual text editor. We had made good progress when the Macintosh arrived. The Macintosh knew nothing about languages at the time, but it understood fonts and multiple fonts were enough to serve the basic needs of classicists. We abandoned our editor and resolved never to address a general problem that the market place would solve for us.

Text retrieval:

Pioneers such as Gerald Salton had laid the foundation for the science of information retrieval in the 1950s and 60s. Techniques already existed to provide efficient searching of textual databases. The tools at our disposal were also powerful and flexible. Unix provided a superb scripting language and development environment within which to reuse existing programs. Unix provided several text searching programs (the infamously non-mnemonic `grep`, `egrep` and `fgrep`). We had the source code for everything and could thus modify any existing Unix program. Searching the TLG and other scholarly textual databases should have been easy.

In fact, it proved quite difficult to build services that our colleagues would actually use on the generic tools of Unix. Three issues confronted us. First, we needed a reasonable interface. Few classicists even today have proven willing to learn how to use a Unix environment directly. Since we were working more than a decade before Web browsers radically reduced the labor needed to create simple interfaces, this task required both programming and design work.

Second, we needed to generate standard citations for the search results. The Unix search utilities returned lines that had a particular pattern. The TLG files had a complex scheme for encoding changes in book, chapter, section, or line numbers. The search program had to examine each line in a file and update the various registers, as well as deal with the exceptions (e.g. line 178a, line 38 appearing before line 34 etc.) that have found their way into our texts. Adding such routines to the fine tuned text scanning modules of the Unix search routines proved non-trivial and would have required virtual rewrites.

Third, speed was an issue – the systems to which we had access were too slow. We learned quickly why David Packard had modified the microcode of this HP computer to increase linear search speeds. Ultimately we were able to match the linear search speeds on the Ibycus on DEC VAX computers by rewriting the core search loop in VAX assembly language so that we could utilize a special pattern matching language. Of course, the VAX computers were more expensive (and more powerful) than the HP computers. Also, while classics departments owned Ibycus computers, we shared all of our machines with many other users – and others did not appreciate our clogging the system with searches that slowed down the disks and the CPU alike.

Fourth, searching classical Greek raises two problems. First, classical Greek has a complex system of accentuation, with the accent shifting around the word as inflections vary. Thus, searches need to be able to ignore accents and scan for underlying stems: e.g., searching for forms of the verb *pe/mpw* (“to send”), we need to match “e)/pempon” and “pem/peis.” We can write regular expressions to accommodate this or simply search for “pemp” or “pe/mp” but such permutations can become complex and require knowledge of vowel length and other features of the language. More significantly, Greek is a highly inflected language. The search tools developed under Unix implicitly assumed English – with its minimal system of inflections – as its model. The tools at our disposal simply were not designed for a language in which a single verb can have a thousand different forms.

Ultimately, we developed a multilingual full text retrieval system from scratch. The system used a set of inverted indices, added 50% to the storage needs of the TLG (a significant factor then), but provided almost instantaneous lookups. The system comprised more than 15,000 *lines of code* [\[clarify reference\]](#) when completed and provided a reasonable TLG solution for a decade, finally yielding to personal computer based programs to search the subsequent TLG CDs.

From 1983 to 2003: Past trends and prospects

The details listed in the section above provide an insight into one particular point in time. The problems described above warrant documentation in part precisely because it is hard to remember now what barriers they posed. Looking back over the past twenty years, the following broad themes stand out:

Increasingly powerful hardware:

Moore’s law continues to hold. Even if technology were to freeze at 2003 levels, we would still need a generation to digest its implications. The cost of storing textual databases such as the TLG is almost zero. We can store hundreds of thousands of images, vast geographic datasets and anything that was published in print.

Visualizations:

These include not only virtual reality displays and geographic information systems but also automatically generated timelines and other data visualization techniques. The result will be possibilities for more holistic analysis of the Greco-Roman world, with philologists making much more effective use of art and archaeological materials than before.

Language Technologies:

I single out the broad class of “language technologies,” a rubric that includes machine translation, cross-lingual information retrieval (e.g., type in “guest friend” and locate passages in Greek, Latin, Arabic, Sanskrit, etc.), summarization, clustering, syntactic analysis (and the possibility of generating large syntactic databases for Greco-Roman source texts), etc. The US defense establishment is investing heavily in rapidly deployable tools for “low-density languages” (e.g., languages for which few if any computational resources exist) – intelligence analysts have found themselves compelled to develop capabilities in languages such as Albanian and Pashtun. The underlying resources for these techniques are bilingual text corpora, morphological analyzers, on-line lexica, grammars and other knowledge sources. Classicists already have these resources online and other languages promise to follow suit. The next twenty years promise to introduce a golden age of philology, in which classicists not only explore new questions about Greek and Latin but also explore corpora in many languages which they will never have the opportunity to master.

Annotation managers:

Classicists have a long tradition of stand-alone commentaries and small notes on individual words and passages. All major literary classical Greek source texts are available from the TLG and many from Perseus. Authors can already publish annotations directly linked to the passages that readers see (rather than buried in separate publications). The hypertextual nature of web reading is stimulating new tools and new opportunities with classicists, who can bring on-line a rich tradition of annotations.

Rise of library repositories:

The World Wide Web spurred a generation of pseudo-publication: documents more broadly available than any print publication in history could at any given time reach millions of machines. The same documents often ran, however, under individual accounts, with many URLs being changed or pointing to documents that were no longer online or, arguably worse, that had been substantively changed since the original link had been added. A variety of library repositories are now coming into use. **[please exemplify in footnote]** Their features differ but all are dedicated to providing long-term storage of core documents and to separating authors from preservation. In the world of publication, alienation is a virtue, because in alienating publications, the author can entrust them to libraries that are designed to provide stable access beyond the lifespan of any one individual. *Unless we transfer stewardship – and control – of our work at some point, then our work will not outlive us.* **[please clarify]**

Convergence of needs:

The examples listed above reflect a common theme: as our computing infrastructure grows in power, the generality of the tools developed increases and the degree to which classicists (and other humanists) need to customize general tools becomes more defined. Where Packard had to create a whole operating system, we were, twenty years ago, able to build on Unix. Where we needed to work on our own multilingual text editor, Unicode provides multilingual support now at the system level. The set of problems particular to classicists is shrinking. We are better able now than ever before to share infrastructure with our colleagues not only in the humanities but in the rest of the academy as well. The rising NSF sponsored National Science Digital Library (NSDL) will, if it is successful, probably establish a foundation for the integration of academic resources across the curriculum. Nevertheless, classicists need to reach out to their colleagues and to begin influencing projects such as the

¹ Among the best known are D-Space (<http://www.dspace.org/>) and FEDORA (<http://www.fedora.info/>).

Unknown
Field Code Changed

NSDL if these science-based efforts are to serve our needs in the future. Otherwise, we may find that simple steps that could radically improve our ability to work in the future will have been overlooked at crucial points in the coming years. Our history now lies with the larger story of computing and academia in the twenty first century.

siemensr 10/2/03 4:00 AM

Deleted: Greg Crane

References for Further Reading

[Below is what has been identified as being needed for the Works Cited, you may add up to 25 references]

Ancient Studies – New Technology, conference held at Rutgers University, December 2002, http://tabula.rutgers.edu/conferences/ancient_studies2002/.

Brunner, T.F., Classics and the Computer: the History, in Accessing antiquity : the computerization of classical databases, J. Solomon, Editor. 1993, University of Arizona Press: Tucson. p. 10-33.

The Bryn Mawr Classical Review: <http://ccat.sas.upenn.edu/bmcr/>.

Busa, R., La terminologia tomistica dell'interiorità; saggi di metodo per un'interpretazione della metafisica della presenza. 1949, Milano,: Fratelli Bocca. 279.

The Database of Classical Bibliography: <http://web.gc.cuny.edu/dept/class/dcb.htm>.

The Duke Databank of Documentary Papyri:

<http://scriptorium.lib.duke.edu/papyrus/texts/DDBDP.html>.

Packard, D.W., A Concordance to Livy. 1968, Cambridge: Harvard University Press.

The Perseus Project: <http://www.perseus.tufts.edu>.

Thesaurus Linguae Graecae Project: <http://www.tlg.uci.edu/>.

TIDES (Translingual Information Detection Extraction Summarization):

<http://tides.nist.gov/>.

GREG CRANE

gregory crane 10/28/03 10:16 AM
Deleted: Brunner, Theodore. "Classics and the Computer: the History," .1993. ◊
December 2002 "Ancient Studies – New Technology" conference at Rutgers ◊
TIDES (Translingual Information Detection Extraction Summarization) ◊
Father Roberto Busa's concordance of Aquinas' Latin writing ◊
Packard's *Livy Concordance* ◊
David Packard's Ibycus system ◊
the TLG - *Thesaurus Linguae Graeca* [?] ◊
the Database of Classical Bibliography ◊
the Bryn Mawr Classical Review ◊
the Duke Data Bank of Documentary Papyri ◊
the Perseus Project