

SIBYL

The US Database of Classical Iconography: Issues of Simplicity and Complexity in Design

Jocelyn Penny SMALL

"Why can't a computer be more like a car? Computers are irrational that's all there is to that. They're nothing but exasperating, irritating, vacillating, calculating, agitating, maddening, and infuriating things."¹

Résumé. Pascal s'est un jour excusé d'avoir écrit une longue lettre parce qu'il n'avait pas eu le temps d'en écrire une courte. Il en va de même de la conception d'outils informatiques. Cela prend beaucoup plus de temps de réaliser un système convivial que d'en faire un qui soit simplement utilisable. La création d'une base de données est un double travail. Premièrement, il vous faut comprendre la manière dont une discipline particulière fonctionne, c'est-à-dire, en termes plus philosophiques, son épistémologie, de manière à enregistrer des données qui soient sensées et organisées de la manière qui convient pour cette discipline. Second et peut-être plus grave problème, ce n'est pas seulement le concepteur, mais aussi d'autres personnes qui doivent être capables d'utiliser cette base de données. Cet article traite de la manière de résoudre ou non ces deux exigences contradictoires : préservation de la complexité des informations dans un logiciel simple d'utilisation.

Keywords: database, classical iconography, mythology, archaeology. **Mots-clés :** banque de données, iconographie classique, mythologie, archéologie.

¹ With all apologies to Alan Jay Lerner and his lyrics for *My Fair Lady*. This paper was one of the presentations at the seminar *Information without Tools?*, organized by the Metropolitan New York Chapter of the American Society of Information Specialists, New York, NY, on September 11, 1992.

✉ U.S. Center; Lexicon Iconographicum Mythologiae Classicae; Rutgers University; Alexander Library; New Brunswick, NJ 08903 (U.S.A.).
E-mail: jpsmall@zodiac.rutgers.edu

Certainly anyone first exposed to computers feels this way. When you buy a car, you don't have to decide what kind of carburetor to install. You buy a whole machine. At the same time it takes less than a day to learn how to use it, not necessarily well, but well enough. Moreover, that learning transfers to other machines of the same class. Computers do far, far more things than cars, which only do two things: stop or go, either ahead, back, or around. But the excruciating, tortuous effort to get computers to work sometimes overwhelms even experienced users. Why did I have to spend two days to get my memory manager to work with my memory cache to work with my split tasker to work with my word processor to work with my database, when I upgraded the operating system? Absolutely everything has to be specified in the most minute detail, and even though the steps are often the same from program to program, the triggers or commands are not. Why can't a computer be more like a car and bury its complexity beneath a simple surface that varies only in glitz and not in ease of use?

The *Lexicon Iconographicum Mythologiae Classicae* (LIMC) is an international project of nearly forty countries to publish a dictionary of classical mythology focusing on the representations of classical myth in classical art. Six of the eight volumes have already appeared. Here, however, I am not concerned about the publication or the international aspects, but solely about the computerization of the American material of the U.S. Center of the LIMC (US LIMC), which is based at Rutgers University.² As Director of the US LIMC, I have been working on the database for seven years and have reached the stage where the database has enough data to make it worthwhile to prepare a formal release of it as a whole. The worst part of the process is that the major group of potential users comprises those who know next to nothing about computers and are proud of it. I am positive that this is true, because there are many more scholars who do not like computers than those who do. So, how am I going to make a database that has over twenty files, not counting indexes and system files, with multiple and varying links to each other, easy enough for any reasonable person to use? Yes, I am only trying to please the so-called reasonable. Those who wave pencils in front of my face as exemplars of the ultimate word processor are welcome to keep their erasers.

Even scholars who use computers present problems. A few years ago the American Philological Association conducted a survey of computer use by its membership. From the responses Prof. Robert Rowland concluded that

² At this time I would like to acknowledge the extremely generous and long-term support of the project by Rutgers University and the National Endowment for the Humanities, the Division of Research.

“we now know that a large portion of the membership has become quite sophisticated in using personal computers for word processing. We also now know that the membership at large is abysmally ignorant of the wide range of additional possibilities made available for their teaching and research by personal computers.” How am I going to produce a database that can be used by someone who responded to the question on the computer survey of what languages do you know with Egyptian and Hittite?

In the beginning I felt that if I could master the database program, so could anyone else. While part of me still feels that way, I also realize that not everyone wants to, much less needs to. As a result, SIBYL will be entirely menu-driven.³ When I was faced with computerizing the catalogue cards describing classical objects in American collections, I realized that what I really had to do was understand how archaeologists, art historians, and those of other academic fields would want to use our information. I had also to consider not just the way our data has been used in the past, but the ways I could anticipate it being used in the future. At an early stage of our computerization I heard a museum curator boast that no one had yet asked a question that the museum’s card catalog could not answer. It did not occur to the curator that no scholar of any ability would waste their time asking questions that they knew could not be answered. Something similar happens with computers. No one wants to waste time using them.

I had decided from the start—and this was an extremely important decision—that the main users would be scholars. This way I was able to reduce my problems regarding terminology to suit those in the field. Jargon was okay within certain limits. For example, it is the practice among classical archaeologists to say that Attic vases are made of clay, but that Attic statuettes are made of terracotta, which is fired clay, even though both the vases and the statuettes are made of the same fired clay. Since some scholars might want all objects made of clay, fired or not, I decided to go for the more basic word, clay, which was also easier to type. But—and this should be a crucial part of any system—not only do I have files that control our vocabulary and define it, but they also include synonyms so that either term, terracotta or clay, can produce the same result. The user should not have to divine our vocabulary, but should have a dictionary with definitions available.

Every single word in every single field in every single file will be indexed in one massive dictionary that records all of the word’s locations. For instance,

³ We have had extensive discussions as to whether users should have access to the command line. If they do, they will, on the one hand, gain far more control over how and what the system produces; on the other hand, they can inadvertently destroy anything from a record to a file to the whole system. Only testing will tell whether the menu-driven form will satisfy most of the users.

consider "Paris", a name which can refer to either the city or the Homeric hero in SIBYL's usage. Again, a database limited to a particular discipline has its advantages, since we don't need terms like "plaster of Paris" or "Paris daisy". Parenthetically, it should be noted that phrases like "plaster of Paris" function as a single unit despite their multi-word composition. Such compound terms are quite common and must be maintained as units for searching. To complicate matters, the term "Paris" in its two major senses, city and hero, occurs in a number of SIBYL's files. As the city, it is found in the Bibliographical File as a place of publication, and, naturally, can be part of the title of an article or a book or even a person's name. As an ancient findspot or provenience, it has a record in the Proveniences file. The one place it might be expected to occur, but does not, is as a current location for an object in the Objects File, because SIBYL at the moment contains material only in American collections. Of course, if a Paris, New York existed like Rome, Texas and had a classical object with a mythological subject, then it would appear there also. As the hero, Paris is the Latin name for Alexandros, the name used in the LIMC, and so the entry "Paris" appears as a synonym in both the Elements file and in the Titles file. Hence, if a scholar wants representations of Paris, the dictionary must be able to automatically link the index records for Alexandros to Paris. I might add that once you have the capability of searching on synonyms, foreign language versions can be added in a like manner. This is an issue that will become of increasing importance, as databases are distributed worldwide.

The front the program presents to the world, if it works, generally goes unnoticed. When it doesn't, everyone suffers. For a parallel example consider some of the really awful productions from would-be desk top publishers. There are "tricks" or rather known ways of making tasks apparent. For this purpose, I have hired a consultant, who specializes in the design of computer interfaces. When I and my computer programmer met with him, he stressed one simple, but incredibly important, principle: users want information not functions. You buy a hammer to pound nails not to display for its intrinsic beauty. As a result, SIBYL lets you search by topic. An "Iconographical Search", the heart of the system, totally disguises the fact that there are two files (Objects and Scenes) involved. Instead you get a choice of looking for: figures, dress, objects themselves, objects represented, titles, etc. At any time, you can press the F1 key for help in addition to the general help given on each menu. SIBYL allows the user to build a complex query word by word in a manner much like that of bibliographical databases.

If, as in the case of "Paris", there are multiple choices for the search word, SIBYL displays those choices from their listing in the Dictionary and lets the user decide which "Paris" is appropriate. If you did not choose the right one, you can, of course, always try again. Remember you are always alone with your computer.

You make your choice from the available "Parises", and then get opportunities to narrow or broaden that choice so that you get only Greek "Parises" or only Greek "Parises" on vases and not sarcophagi. In any case, successive lists are automatically saved. Too many of us have second thoughts or, worse yet, act too promptly on our first thoughts. At any time during the process, you can look at either one-line lists of the records or full screen reports. When all the criteria have been filled, either full catalogue cards or canned lists with pre-selected fields can be produced either to disk or to the printer. It may be of interest that our own experience in answering queries has been that scholars prefer the full catalogue entry to lists. You can also use the viewing screens that come with each file for a hypertext-like mode that enables you to switch to related files by pressing different function-key combinations.

So far I have focussed mainly on terminology and very little on the structure of the system. Why in the world do I have so many files in the first place? Is it really easier than using one, humongous single file to contain everything? If everything is in one file, then you know where to look. There. I use a relational database program called Advanced Revelation (AREV). It allows for up to 64 K per record with all fields being variable in length and fully and Boolean searchable. When setting it up, I realized that we were dealing with objects that could have more than one scene on them, a classic one-to-many relationship. The scenes in turn could have multiple elements—figures and things, like architecture, armor, and vehicles. Either I would have to have a file that depended on the smallest common denominator, the element, and then repeated all the information about the scene and then the object or I would have to go relational with more than one file. (Yes, I know I am using relational in the loose sense.) And when you are having more than one file, you might as well have many files. That way you have a truly modular, flexible system that enables you to keep good control over the data being entered.

One other capability of AREV determined the file design. It allows for repeating fields that can be associated with each other. I can catalogue a scene of the *Judgment of Paris* with all its elements and their associated attributes. Athena, for example, would hold a shield, Hera a sceptre, and Aphrodite a flower. Hence I am able to use only two files: the Objects File, with all information pertaining to the object as a whole, and the Scenes File, with multiple scenes and even multiple interpretations of the same scene. Within each Scene record, repeating fields relate the components of the scene and their accouterments: for figures, what they are wearing and holding; for things, it varies somewhat more. The result is that I can find all scenes with Paris and Aphrodite holding a flower and not a vase.

Why stop there? Scholars always want more, now. I have a file, Elements, which I have already mentioned, that defines the figures and elements in a scene so that I can find all representations of women or of certain professions like warriors or heralds. Conversely, for the person interested in a particular object, all vases depicted on Greek objects can be retrieved. This arrangement keeps the representation of a vase distinct from an object that is a vase. Furthermore, the Scenes File is also linked to the Titles File, which defines the combination of figures rather than just individual figures, although certain scenes may consist of only single figures, such as statues of Herakles or Aphrodite. One aspect of the design may strike some as a flaw: information in one file may be repeated in another—a violation of Codd's rules for database design. In this case, the data in the Titles file about the *Judgment of Paris* overlaps with the information about Paris in the Elements file. Paris is defined in both places as belonging to the Trojan Cycle. While it might seem unnecessary for this information to be repeated about Paris, who appears only in the Trojan Cycle, it becomes crucial for someone like Herakles, who appears in a number of different cycles in addition to performing his own labors. Consequently, each scene or story that Herakles appears in has to get its own classification. Moreover, and quite significantly, I see no reason not to repeat information, when it is genuinely helpful, makes the user's life easier, and even makes the design of the database simpler.

One last remark about the kinds of information recorded. There seems to be an assumption on the part of many librarians and information specialists that all information is hierarchical. Some is, some isn't. That Paris is a hero means that he belongs to a heroic cycle, specifically the Trojan Cycle, and further to particular subsets of it, such as the *Cypria* and the *Iliad*. This data is hierarchical; that his "profession" was that of hero, warrior, lover, and archer (note the multiples) is not. It is merely informational. It tells you more about the figure, but it does not fit any particular scheme that encompasses all that exists on heaven and earth. In other words, a modular system that incorporates both hierarchical and non-hierarchical information works best. Likewise, the overlapping of categories and bits of information more closely reflects the way we use language. Moreover, these files can also be used as part of a search engine for retrieval of full text stored in other databases. Why think of all the possible names or terms or anything, if the machine can not only produce them, but selections of them specifically tailored to your request. Why wrack your brains for the names of all the figures in the *Iliad*, if the machine can name them for you?

After testing at several sites, the database will be released in a runtime version, which offers three advantages: cheapness (one-tenth the list price of the full version); stability of design, since users cannot change it; and ease of installation, because the US LIMC would only be concerned with how its version

