# CONTRIBUTIONS

## *Commentary*

## How to Manage Data Badly (Part 1)

In a landmark article in *The American Statistician*, Howard Wainer (1984) presented ideas for "How to Display Data Badly," wherein good data are ruined by bad graphics. Wainer presumed too much. In this essay, I extend his concept by presenting ideas and examples of how scientific data can be managed badly so that they never even make it to the graphics stage. Modern database management software, continually improving hardware and networks, and many sound recommendations for managing ecological data (e.g., ESA 1995, NRC 1995) are making it increasingly difficult to manage data badly. It can still be done, however, and judging by various so-called "horror stories" one hears about adventures with ecological and other scientific databases, is done frequently. Those people still having trouble mismanaging data, whether they are database managers, administrators, or scientists, will find the following techniques helpful.

### Techniques for database managers

*Rule 1. "One world, one database"*

Two time-honored techniques used by database managers are: (1) make it hard to get data into the system, and (2) make it hard to get data out of the system. Although experts often use both techniques, novices may wish to start with just one. A data system that is far more complex than is necessary will usually do the trick. In *Principles of Data-Base Management*, James Martin (1976) suggested that one reason for failure of long-term databases is "Plans for the installation of a grandiose all-embracing system." This strategy is effective because it makes development time long, data loading slow, and data queries difficult to formulate. Discouraged data collectors and users will seek solutions elsewhere. In one case, a national marine water quality database

used such stringent quality assurance procedures that data collectors were loath to enter their data. Consequently, the users learned a lot about data quality but very little about water quality. Another organization thought that merging their scientific data systems with their administrative data systems would be easy "because they both use Oracle software."

*Rule 2. Users are losers*

Practitioners of the mystical cult of database management do not need meddlesome ideas from potential users of a data system, such as those gathered during those boring system requirements exercises. Do not compromise your design or processing efficiency by consideration of the system's usefulness to scientists. If scientists insist on contributing to the design, invite them to an Information Management Needs and Requirements Workshop (the name alone is scary), where you can use tips from Zave and Jackson (1997) on four dark corners of data system requirements engineering to stymie input.

*Rule 3. What's good for General Motors is good for science*

Design data systems for research projects the same way you would for a bank or an insurance company. The paths of scientific inquiry are as fixed as the steps followed in manufacturing and selling cars. It does not matter if the system must process thousands of transactions per day or one batch load per month. After all, a byte is a byte is a byte, no matter where it is found. Commercial software for database management systems (DBMS) is driven by the commercial market. More than one research group has found that their database designer was unaware of the differences between business and scientific databases (Pfaltz 1990), and that the design recommended was not suitable for their less structured data, less formal organization, and less predictable user needs.

*Rule 4. Reinvent the wheel*

Another powerful technique is to resist the efforts of unimaginative people who promote standards for formatting and exchanging data. Surely you can think of more interesting codes for species than the Integrated Taxonomic Information System and more clever codes for chemicals than the Chemical Abstracts Service. You can always develop better software than is available off the shelf. Moreover, it is unlikely that any previous data system built by others will be of any value to you. After all, if we did not reinvent wheels, they would still be made of stone. One organization, in a burst of creativity, let each of its branches come up with their own coding system for fish names, thereby making more work when they later wanted to search and merge species catch data.

*Rule 5. Data governance: totalitarian or anarchist*

People who collect data cannot be trusted to manage them. Seize control of all data and get them into a centralized system. This allows data sources to disavow all ownership and responsibility, and therefore not bother with subsequent corrections and updates. To avoid bias, metadata (information about data) should be written by people not familiar with the scientific discipline. This can provide much needed comic relief, as when software engineers interpreted "pH" as a code for telephone and then wondered why the value had only two digits separated by a decimal point. This same group was so keen to integrate data that they insisted that one group studying lakes and another studying estuaries use identical formats for pH. That made it easy to calculate the mean pH for the nation's waters, should anyone ever want to know the answer (e.g., Country—USA; Area—$9.36 \times 10^6$ km$^2$; pH—6.9).

When adopting data and metadata standards, avoid the middle road. You can get along with scarcely any standards (an absence of any data policies common to all groups can let data sources express themselves freely and preserve our rich data diversity) or, conversely, lay them on thick. You can require metadata with formats so onerous they will be ignored (admirable, but common). On the other hand, you can do what one group did and simply include a FAQ (Frequently Asked Questions) file with the data. (Usually FAQs have not been, but the authors probably liked the answers anyway.)

An infallible way to frustrate data users is to let them choose from multiple, mutually inconsistent versions of the same data set. For example, Schmidt (1998) had to invest considerable detective work to get a consistent data set from published and underground versions of data from the Geochemical Ocean Sections program.

Although engineers keep developing new algorithms for detecting and rejecting bad data (Zhang et al. 1992, Baldick et al. 1997), blindly relying on computer programs to validate data can lead to trouble. In one well-known misapplication of computerized range checks, NASA computers programmed to delete concentrations of ozone below a certain value to eliminate "noise" failed to detect the ozone hole over Antarctica (Edwards 1998).

*Rule 6. Silicon is thicker than DNA*

Communicating with computers is easier than with humans. Despite complaints about the writing of computer people, a few phrases of computer jargon and acronyms can express a thought that would take plain English many sentences to explain. And compared with computer languages, English is a frail, illogical language that is full of conflicting and inconsistent rules. Much of the poor communication between computer people and normal people results from use of this imprecise language. English even has the peculiarity that if enough people violate a rule of grammar for enough time, the rule changes to meet the practice! (Try that method with syntax errors in a computer language.) Machine language, the ultimate in clarity and conciseness, has no tolerance for solecisms.

Data modeling techniques, which are used to design databases, can provide fertile ground for confusing scientists and administrators. Entity-

relationship diagrams can be translated into computer files far more easily than into scientists' and administrators' heads. Hay (1998) points out that database managers can proudly take most of the credit for the bad reputation of data modeling; too often, the data modeling software receives all the glory.

*Rule 7. Sell! Sell! Sell!*

Database managers must promote their systems vigorously. Whenever anyone asks for a certain feature or has a data set they would like to add—no matter how irrelevant or unsuitable for the system—you should promise to add it. Successful database managers, anxious to please anyone with a glimmer of interest, will never say no to a single bit of data and will try to make their data systems be all things to all people. An associated rule, so commonly used that it hardly needs to be stated, is "Always underestimate the time needed to bring a data system online."

*Rule 8. Mapping administrators*

If getting scientists to manage data has been likened to herding cats, then getting administrators to pay attention to data management is like herding lemmings—they are going to plunge off some cliff no matter what you suggest. At least a cat can sometimes be lured with a bowl of milk. When working with administrators, remember that their one weakness is an unnatural fondness for Geographic Information System (GIS) maps. Axiom: Never show an administrator a table of data; always use a GIS map instead. Even data collected in a single laboratory experiment can be plotted on a state map with an arrow pointing to the location of the lab building. In his seminal book *How to Lie with Maps*, Monmonier (1991) shows some of the clever things that can be done with maps. Dreadful data can often be swept under the rug of a colorful map.

## Techniques for administrators

*Rule 9. Talk the talk*

With everyone from the President on down talking about a national information infrastructure, be sure to join the national fervor for getting data flying all over the place. But be careful not to get involved in the painful task of creating an effective data management system in your organization. One of the biggest disadvantages of using database management software packages is that they burden the project with the need to get organized, to make decisions, to coordinate, and to be consistent. Take special care to avoid CASE (Computer-Aided Software Engineering) tools used for designing databases; these are particularly insidious in demanding feedback from administrators about organizational procedures. Data managers may be slow to recognize your perspicuity, but you will know what you want for a data system when you see it. Another good use of "messy" organizational problems is that they can be as effective as technical ones in inhibiting data sharing (Evans and Ferreira 1995).

The considerable work needed to create a sound scientific database, much of it done during the early stages, is worthwhile only when there is long-term intent to maintain the database. Improvident administrators can take advantage of this weakness in the system life cycle by limiting themselves to short-term commitments. Quint (1998) described the mounting death toll of databases. Even if the data system succeeds, the costs of long-term maintenance are often not included in project budgets (ESA 1995, Farrey et al. 1999).

*Rule 10. Do less with less*

Clearly, one of the best ways to create poor data sets is to underfund data management, and, in fact, this technique is commonly used. If every study of research projects ever done since the invention of computers has recommended that 10–20% of the project budget be spent on managing the data, why not impress your budget people by allotting only 5% in your study? This will put the data system on a death march (Yourdon 1997). Often, administrators can apparently achieve a data management system solely with hardware and software, thereby overcoming the superstition that qualified people are a crucial commodity. National data centers have been called "data cemeteries" because of inadequate funding to handle the flood of incoming data (French 1990).

Frequently, administrators can indulge in one of their favorite pastimes (technology transfer), where the objective is to leave sophisticated data systems and equipment with groups that do not have the trained personnel or budget to operate them. One of the nicest GIS applications ever developed for managing natural resources is gathering mold in a tropical jungle because the donors forgot to add people to operate the system.

## Literature cited


Baldick R., K. A. Clements, Z. Pinjo-Dzigal, and P. W. Davis. 1997. Implementing nonquadratic objective functions for state estimation and bad data rejection. IEEE Power Engineering Review **17**:67.

Edwards, D. 1998. Data quality control/quality assurance. *In* W. K. Michener, J. H. Porter, and S. G. Stafford, editors. Data and information management in the ecological sciences: a resource guide. LTER Network Office, University of New Mexico, Albuquerque, New Mexico, USA.

Ecological Society of America. 1995. Report of the Ecological Society of America Committee on the future of long-term ecological data. <http://www.sdsc.edu/FLED/FLED.html>

Evans, J., and J. Ferreira, Jr. 1995. Sharing spatial information in an imperfect world: interactions between technical and organizational issues. Pages 448–460 in H. J. Onsrud and G. Rushton, editors. Sharing geographic information. Center for Urban Policy Research, New Brunswick, New Jersey, USA.

Farrey, P. M., M. L. Mooney-Seus, and H. C. Tausig, editors. 1999. Out of the fog: furthering the es-

tablishment of an electronic environmental information exchange for the Gulf of Maine. Report 99-1. New England Aquarium, Boston, Massachusetts, USA.

French, J. C. 1990. The challenge of scientific database management. *In* NSF invitational workshop on scientific database management, March 1990. National Science Foundation, Washington, D.C., USA.

Hay, D. C. 1998. Making data models readable. Information Systems Management **15**:21–33.

Martin, J. 1976. Principles of database management. Prentice-Hall, Englewood Cliffs, New Jersey, USA.

Monmonier, M. 1991. How to lie with maps. University of Chicago Press, Chicago, Illinois, USA.

NRC (National Research Council). 1995. Finding the forest in the trees: the challenge of combining diverse environmental data. National Academy Press, Washington, D.C., USA.

Pfaltz, J. L. 1990. Differences between commercial and scientific databases. in NSF invitational workshop on scientific database management, March 1990. National Science Foundation, Washington, D.C., USA.

Quint, B. 1998. The mounting death toll. "Dead databases" revisited. Database **21**:14–22.

Schmidt, G. 1998. All that is labeled data is not gold. EOS, Transactions, American Geophysical Union **79**(28):336.

Wainer, H. 1984. How to display data badly. American Statistician **38**:137–147.

Yourdon, E. 1997. Death march: the complete software developer's guide to surviving "Mission Impossible" projects. Prentice Hall, Upper Saddle River, New Jersey, USA.

Zave, P., and M. Jackson 1997. Four dark corners of requirements engineering. ACM Transactions on Software Engineering and Methodology **6**:1–30.

Zhang, B. M., S. Y. Wang, and N. D. Xiang. 1992. A linear recursive bad data identification method with real-time application to power system state estimation. IEEE Transactions on Power Systems **7**:1378–1385.

*Note: Part 2 of this essay, to appear in the next issue, covers techniques for scientists who do not wish to fall behind database managers and administrators in managing data badly. Then it shows the synergy that can result from all three groups working together to mangle data. An epilogue confesses that this article was written in an ironic tone, and provides a few simple suggestions on how to manage data well.*

*Stephen S. Hale*
*Atlantic Ecology Division*
*U.S. Environmental Protection Agency*
*27 Tarzwell Drive*
*Narragansett, RI 02882*