Report of the Invitational NSF Workshop on
Scientific Database Management
Charlottesville, VA
March 1990

Anita K. Jones, Chairperson

Scientific Database Management

(Panel Reports and Supporting Material)

edited by
James C. French, Anita K. Jones, and John L. Pfaltz

Supported by grant IRI-8917544 from
the National Science Foundation

Technical Report 90-22
August 1990
Department of Computer Science
University of Virginia
Charlottesville, VA 22903

Abstract

On March 12-13, 1990, the National Science Foundation sponsored a two day workshop, hosted by the University of Virginia, at which representatives from the earth, life, and space sciences gathered together with computer scientists to discuss the problems facing the scientific community in the area of database management.

A summary of the discussion which took place at that meeting can be found in Technical Report 90-21 of the Department of Computer Science at the University of Virginia. This document provides much of the background material upon which that report is based.

Program Committee:

Hector Garcia-Molina, Princeton University
Anita K. Jones, University of Virginia
Steve Murray, Harvard-Smithsonian Astrophysical Observatory
Arie Shoshani, Lawrence Berkeley Laboratory
Ferris Webster, University of Delaware - Lewes

Workshop Attendees:

Don Batory, University of Texas - Austin
Joseph Bredekamp, NASA Headquarters
Francis Bretherton, University of Wisconsin - Madison
Michael J. Carey, University of Wisconsin - Madison
Vernon E. Derr, National Oceanic and Atmospheric Administration
Glenn Flierl, Massachusetts Institute of Technology
Nancy Flournoy, American University
Edward A. Fox, Virginia Polytechnic Institute and State University
James C. French, University of Virginia
Hector Garcia-Molina, Princeton University
Greg Hamm, Rutgers University
Roy Jenne, National Center for Atmospheric Research
Anita K. Jones, University of Virginia
David Kingsbury, George Washington University Medical Center
Thomas Kitchens, Department of Energy
Barry Madore, California Institute of Technology
Thomas G. Marr, Cold Spring Harbor Laboratory
Robert McPherron, University of California - Los Angeles
Steve Murray, Harvard-Smithsonian Astrophysical Observatory
Frank Olken, Lawrence Berkeley Laboratory
Gary Olsen, University of Illinois - Urbana
John L. Pfaltz, University of Virginia
Peter Shames, Space Telescope Science Institute
Arie Shoshani, Lawrence Berkeley Laboratory
Ferris Webster, University of Delaware - Lewes
Donald C. Wells, National Radio Astronomy Observatory
Greg Withee, National Oceanic and Atmospheric Administration

National Science Foundation Observers:

Y.T. Chien
Robert Robbins
Larry Rosenberg
John Wooley
Maria Zemankova

Other Contributors:

Umeshwar Dayal, DEC Cambridge Research Laboratory
Nathan Goodman, Codd and Date International
James Ostell, National Library of Medicine

Scientific Database Management[1]

1. Introduction

An interdisciplinary workshop on scientific database management, sponsored by the National Science Foundation, was held at the University of Virginia in March 1990. The workshop final report, a digest of the workshop proceedings summarizing the panel discussions and highlighting the workshop recommendations, is available as a separate technical report (TR 90-21) from the Department of Computer Science, University of Virginia, Charlottesville, VA 22901.

This document contains the individual panel reports from the workshop along with other supporting material used in the preparation of the final report. We have included the separate panel reports so that the interested reader will have the opportunity to form his/her own opinions. Self-describing data formats received much attention in the workshop so we have included an example of one international standard format (FITS) as an appendix. Because of the thoughtful issues raised by the participants in their position papers, we have included those also as an appendix.

[1]Panel reports and supplementary material used in the preparation of the final report of the NSF Invitational Workshop on Scientific Database Management, March 1990. The workshop was attended by Don Batory, Joe Bredekamp, Francis Bretherton, Mike Carey, Y.T. Chien, Vernon Derr, Glenn Flierl, Nancy Flournoy, Ed Fox, Jim French, Hector Garcia-Molina, Greg Hamm, Roy Jenne, Anita Jones, David Kingsbury, Tom Kitchens, Barry Madore, Tom Marr, Bob McPherron, Steve Murray, Frank Olken, Gary Olsen, John Pfaltz, Bob Robbins, Larry Rosenberg, Peter Shames, Arie Shoshani, Ferris Webster, Don Wells, Greg Withee, John Wooley, and Maria Zemankova. The workshop was supported by NSF grant IRI-8917544. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the panels and do not necessarily reflect the views of the National Science Foundation.

2. Multidisciplinary Interfaces

Panel members:

Ed Fox, VPI & SU
Roy Jenne, NCAR
Tom Kitchens, DOE
Barry Madore, IPAC/Caltech
Gary Olsen, Univ. of Illinois
John Pfaltz, Univ. of Virginia (chair)
Bob Robbins, NSF

2.1. Overview

From the perspective of users of scientific databases, it is essential that relevant existing databases be easily identified, that flexible tools for searching to obtain appropriate subsets be provided, and that processing of the data be done at a level suitable for investigators in multidisciplinary projects.

Our panel has focused on a few key issues relating to this process, so that policies and initiatives can be developed that will provide more efficient and effective access to scientific databases that are often obtained at great expense. It is essential that the entire process, from planning to create databases, to collection of data, to identification of suitable data formats and organizations, to selection or construction of access and manipulation software, to cataloging, to online use, and later to archiving for future requirements, be governed by standards that at once anticipate future activity, and on the other hand have as little associated cost as possible (including personnel, space, and direct expense).

We note that there are many standards in related disciplines that need to be reconciled if interoperable systems are to be truly functional - as a result, databases are often published in archival forms that are hard to analyze by other researchers. Also, the publication/cataloging/access issues of scientific databases are closely allied to the work of librarians, information scientists, and information retrieval researchers - and those disciplines should be involved in future database development projects. So called "meta-data" plays a crucial role in this entire process, and is especially useful for aiding cataloging, access, and data interpretation. Furthermore, education in the use of networks and access software as well as in data manipulation methods is essential not only for researchers and graduate students, but also for undergraduates who should be exposed to the existence and use of data in modern day science.

In dealing with these issues, our panel has focused on:

- **Meta-data** as an issue/term.
- **Publication** of databases as citable literature.
- **Locating** databases, and **navigating** through them.
- **Standards** to facilitate data usage.
- **Educational** needs for the effective use of databases.

It should be noted that these are not completely disjoint topics. In particular, the first three bullets are clearly related to each other, and each has implications on the issue of standards.

2.1.1. Important Considerations

While we chose to focus on the preceding topics, we also identified five important considerations which should accompany any discussion.

(1) It is essential that any database approach be simple and relatively inexpensive to use. Otherwise it fails to provide the service one wants of it. Expense may lie in the eye of the beholder; but, at least, simple common operations should cost less than complex infrequent operations.

By database use, we mean both its development by participating scientists as a repository of their data, as well as its secondary reuse in subsequent research.

(2) The purpose of a scientific database is to "facilitate" scientific inquiry — not to hinder it! It is to become a tool, or a resource, to assist scientific inquiry. Development of the database itself is not a scientific process.

(3) The development of standards must facilitate both the creation and subsequent reuse of scientific databases — but they must not become a straight-jacket. The database tool must allow for flexibility, creativity, and playfulness on the part of a scientist.

(4) We should guard against *single port failures*. By this we mean that the database system must not be so centralized that the failure of a single node, or site, renders the entire system inoperative. This warning also applies to the dangers of adopting a single database philosophy which might, in and of itself, preclude a certain style of doing science (e.g. object oriented versus hierarchical, fractal versus continuum), as well as guarding against concentrating the resources of an archive in a single physical location. Diversity of approach, multiple collections, competition for resources will allow the field to both survive and to flourish.

(5) A flexible approach to the *user interface* must be conducive to research at a variety of levels of sophistication. This commonly implies a layered implementation. Menu-driven, as well as command-driven options at the very least must always be available. Further, different functionally oriented interfaces may be needed to support (2) above.

2.1.2. Generic Types of Databases

In the course of discussing the major foci of our panel, we repeatedly encountered the fact that the relative importance of one approach in comparison to another is extremely dependent on the *type* of database collection under consideration. What is appropriate descriptive meta-data for one type of collection may be either completely unnecessary or totally inadequate in another. Different types of data collections require different access methods, and have different publication requirements. All too often, major disagreements (as in the evening plenary discussion) occur because the participants are implicitly assuming different database types.

Our panel observed that there is a spectrum of database types, which are characterized in terms of a number of dimensions (which need not be independent). The three we clearly identified (we suspect there may be more) are:

level of interpretation,
complexity, and
source.

(Note: in section 2 of this report, "complexity" is replaced by "intended analysis" as a result of discussions in Panel 2.)

Level of interpretation: At one extreme of this "value-added" dimension is a simple collection of "raw" data, or real world observations, and at the other extreme would be a collection of interpreted, or highly processed results, sometimes called "information". Examples of the former might be a physical collection of badger pelts collected in central Kansas or a file of sensor readings. It may

be the case that physical artifacts or instruments must be retained and that this can only be done in a single archive or at a single location; however, in general, replication of evidence is desirable when possible for future interpretation.

Examples of the latter extreme might include well-structured tables of summary, statistical, or aggregate data. The inference *rules* of a knowledge database would also be examples of the latter.

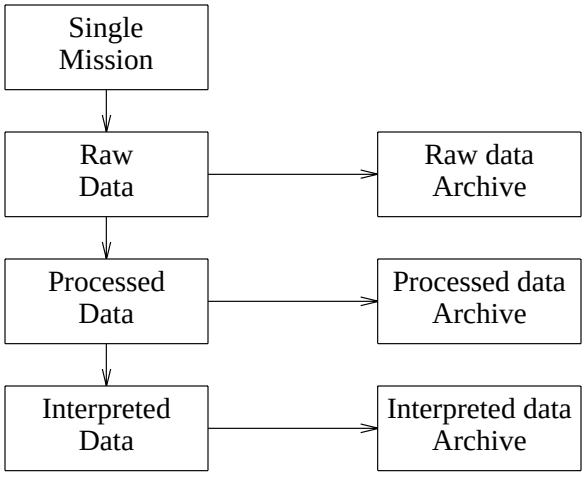
We note that there will typically be various interpretations of raw data, and that the interpretations, and/or models they relate to, may incorrectly represent important aspects of reality. On the positive side, however, the latter allow scientific theories to be developed and tested; since data will increasingly be stored in all-digital form, they should be replicated in a number of locations for increased reliability and security.

Complexity: This dimension may be measured in terms of the internal structure of a database, a kind of syntactic complexity; or in terms of its cross-relationships with other data sets, a kind of semantic complexity.

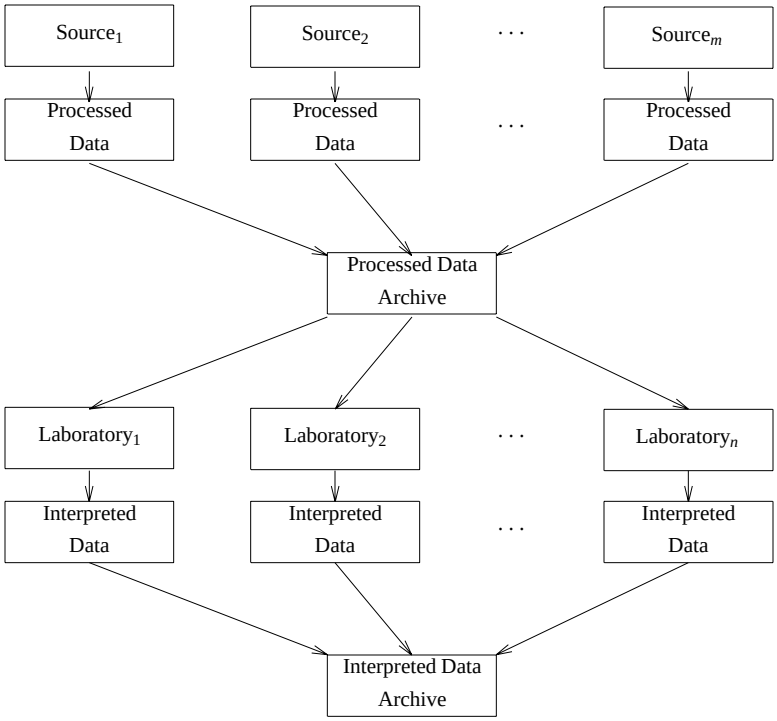
Source: We concluded that this dimension, which is not generally mentioned in the database literature, may be the most fundamental. In Figure 2-1, we illustrate a familiar *single-source* database environment. Here we envision a single mission, such as the *Magellan* planetary probe, generating the data that is collected. Such *raw* data may be retained in its original state in a "raw data archive". Commonly, the raw data must be processed, by instrument calibration or by noise filtering, to generate a collection of *usable*, or processed data. Finally, this processed data will be *interpreted* in light of the original goals of the generating mission.

Both the syntactic complexity and the semantic complexity of the interpreted data will be much greater than either of its antecedent data collections. It will require different search and retrieval requirements. Possibly, only it alone will be "published".

Figure 2-2 illustrates a typical *multi-source* data collection. This structure would characterize the



Single-source Data Collections
Figure 2-1



Multi-source Data Collections
Figure 2-2

Human Genome project in which several different agencies, with independent funding, missions, and methodologies, generate processed data employing different computing systems and database management techniques. All eventually contribute their data to a common data archive, such as GENBANK, which subsequently becomes the data source for subsequent interpretation by multiple research laboratories that also manage their local data collections independently.

In each of the local, multiple, and probably very dynamic, database collections one would expect different retrieval and processing needs, as well as different documentation requirements.

The data collections associated with most scientific inquiries will occupy middle positions along these various dimensions. For example, the primary satellite data collections discussed in the Ozone Hole case study by Panel 4 represent an almost perfect example of the linear structure illustrated in Figure 2-1. However, the introduction of ground observation data into the overall study moves it towards the multiple-source structure of Figure 2-2.

We believe that this classification of data collection types, however imperfect, is an important contribution by our panel.

2.2. Meta-Data

The panel discovered that virtually all of the so-called meta-data of interest to its members was those data items which described other data — in particular, that raw or interpreted data which constituted the scientific focus of the collection. But the meaning of term *meta-data* is extremely overloaded and highly charged. Consequently, we eschewed the term altogether and simply talked about *descriptive data*. See recommendation (1) below.

Ancillary descriptive data can be used to describe an entire collection or it may describe individual instances within the collection.

We identified two broad classes of descriptive data:

(1) **Objective:** This kind of descriptive data is in some sense "raw". These data are "facts", not amenable to later change. Examples of objective descriptive data would be:

identification
origin, or "authority"
format
how obtained, e.g. instrument, situation, method.

It was noted that the first two might be called "curatorial". It is the kind of data that is associated with library or museum collections.

(2) **Interpretative:** This kind of descriptive data is interpretive in nature. Some may be included when the collection is first created. Other forms of interpretive data may be appended over the lifetime of the collection, as understanding of the associated data develops. Examples might be:

quality (accuracy, completeness, etc.)
content (in the sense of significance)
intent (why the collection was assembled)

It was observed that objective descriptions ought to be simple and machine interpretable, possibly conforming to fairly rigorous standards. Subjective descriptions might be relatively unstructured, e.g. natural language strings.

Whenever possible, the types and formats chosen for descriptive data should be carefully reviewed by a panel of experts, including representatives from the information sciences, since this data is often essential for subsequent cataloging and database selection. Simple policies, such as having all references to the published literature be in standard formats, are essential, so proper linking and integration of data with publications is feasible in a cost effective fashion as hypertext and hypermedia methods gain acceptance.

2.2.1. Current Status

Treatment of different kinds of descriptive data vary.

(1) **Physical layout:**
This kind of low-level description which is necessary for data transport protocols is most developed. There are some standards in some disciplines (often in small user groups within a discipline). Most are evolving standards, such as FITS [WELL81].

Few layout standards are cross-disciplinary. This is important for archival storage, as well as for transport. Investigation of ASN.1 (Abstract Standard Notation-1) and a variety of intra-disciplinary approaches is a priority.

(2) **Internal structure:**
A dictionary of relational schema is a description of internal structure. It is DBMS specific.

A class hierarchy is used to describe the internal structure of object-oriented databases. This kind of description is system or language specific (e.g. C++, Objective C, or derivatives).

(3) Cross-reference structure: There is a growing interest in hypertext and hypermedia efforts, and various international standards in development to coordinate linking between items. In the library science community, there are various standards for cataloging, including MARC, and markup standards, often based upon SGML [ISO86], that should be followed so that citations, co-citations, and other relationships can be automatically identified.

(4) Content/quality: Whenever no clear manipulation of descriptive information other than exhaustive search can be identified, the use of methods for handling unstructured textual information should be adopted. When a hierarchical structure is clearly identified, a markup scheme compatible with SGML should be used, so that context is clear and different types of elements are easily identified (e.g., section heading vs. paragraph). Various international standards for character sets and foreign languages should be followed.

2.2.2. Recommendations

(1) Serious discussions of scientific databases would benefit by avoiding the term *meta-data*. If ancillary data is descriptive of a collection as a whole, its structure, or individual instances within the collection, call it that. If the purpose of the ancillary data is to interpret the data, call it *interpretive data*, or if it is an operator to compare data items call it that, etc. Whenever possible, such descriptions should be done in a declarative form and if it is also possible, should be done based on some formal or axiomatic scheme so the semantics are clear and so that in some cases, machine manipulation is feasible. Furthermore, descriptions should always be packaged with the related data.

(2) Appropriate descriptive data specifications for major collections should be created by multidisciplinary panels. These panels should include specialists in library/information science and information retrieval research.

(3) Controlled terms used in descriptive data should reference standard lexicons established for the particular collection, or for a class of collections. Lists of synonyms and other types of related terms in the lexicon should be encouraged. (See "standards".) Superceded terms should be retained in synonym lists, possibly with attached warning of obsolescence.

(4) The scientific community should establish a pool of "documentation expertise". This would consist of specialists who are familiar with the scientific discipline and description methodologies. This expertise is analogous to the kind of editorial expertise available to the *Scientific American*.

(5) Standards of an appropriate description, and an appropriate lexicon should be evolutionary and discipline specific. But we should standardize the form of descriptions. As an example, [GROS88] describes protocols for producing extensions to the FITS standards, not actual extensions themselves. SGML [ISO86] was also offered as a possible standard for descriptions.

(6) Integrated methods to use lexicons and descriptive data at varying levels of detail, with flexible user interfaces and user modeling, with a variety of natural language like as well as Boolean-based query schemes, should be investigated, and tied in with educational efforts, to improve the ability of researchers to find relevant databases, and then data within these collections.

2.3. Database Publication

It was argued that database collections should be treated as a form of scientific literature, and as such should conform to generally accepted conventions of publication. One important convention is that assertions should cite the sources of information used. An interpretation or assertion derived from a database collection should cite the database.

(1) There should be a reasonably standard way of uniquely citing (referencing) a database collection or, when appropriate, some subset of items within a collection.

(2) A literature/database citation must be permanent and recoverable. An agent publishing a database should assure access to the database in the form originally published.

(3) By published literature, we normally mean refereed literature. As with literature, refereeing a database can only provide a measure of quality control, but not guarantee the accuracy of the data. In particular, databases should be refereed in terms of our understanding of the discipline at the time of publication.

2.3.1. Current Status

CD-ROM's can be assigned an ISBN.

MARC (used in library catalogs) provides standards for cataloging databases.

Publication today is usually in two forms: by media that is distributed (e.g., tapes, CD-ROM, diskettes), and by network access (e.g., for FTP, for client-server style access or download). While standards for distributed data have clear precedent and can be extended to new media types, their "status" in the scientific community must be increased. This relates to the issues of who manages databases, what is the role of PI's vs. national and international data centers, who finances the maintenance and preservation of databases, and other issues.

On the side of network access, the current situation is even less clear. There are a growing number of online databases on the Internet, but no coordinated policy over NSFNET for cataloging, giving credit, financing service and user support, and other related archive issues.

2.3.2. Recommendations

(1) The NSF should include the matter of cataloging and the prior problem of publication of databases in its planning of NSFNET and other national networks such as the planned NREN. Research proposals for projects leading to databases should be required to include plans for publishing, cataloging, and either maintaining or transferring results to national archives.

(2) The scientific communities should be encouraged to develop methods to encourage the publication of important databases, and to properly reward those involved.

(3) There should be an international standard format for citing database collections. A standard to identify subsets within a collection should be developed.

2.4. Database Location

"Browsing", a term that occurred in several position papers, can refer to three different levels of data location described below. It is concerned with the discovery, location, or finding of potentially interesting scientific information. The inability to find that relevant associated datasets even existed was cited by the case study panel as a major issue in the "Ozone Hole" initiative. "Browsing" roughly corresponds to the kind activity that all of us employ when searching the card catalog or the stacks of a library of printed literature. Our panel believes that it helps to isolate three rather different modes of data location, all of which play a significant role.

(1) **Discovering the existence of database collections:** At a very gross level, one may discover that a particular organization has a certain number of named (or referencable) collections. This is very much like finding the book titles a library holds in a particular discipline. Roy Jenne distributed a listing of datasets at NCAR as an example. On-line inventories exist as described below.

(2) **Locating "relevant" database collections:** It is one thing to discover that a database collection, such as GENBANK, exists. It is even more important to be able to locate databases that are likely to be relevant to a particular inquiry. A title by which a collection is known may be descriptive; but frequently the level of description is insufficient to indicate probable relevance. In traditional literature, abstracts are provided to give a more accurate indication of content, and thus probable relevance.

(3) **Finding useful data in a collection:** One seldom wants the entire collection to answer an inquiry; frequently it is a very small subset that is desired. Standard database query languages are designed to address this latter kind of data location.

Cross-disciplinary searching is important. So, vocabulary mapping aids and systems for searching database descriptions in multiple databases will certainly be required.

2.4.1. Current Status

Beginning with the CONIT system at MIT and related systems such as the University of Illinois, dating back to the 1970's, a variety of systems have been developed to serve as "front-ends" to database systems. These often allow users to connect to systems, sometimes allow a common command language that hides the query syntax of systems like Dialog and BRS, and in some cases map vocabularies across databases. The larger early systems have led to networks and PC versions.

There has been an increasing federal awareness of the need for information about the availability and status of datasets from about 1973 onward. Under the National Climate Program Office, information about 800 datasets held by the U.S. and Canada (many with global extent) were collected in 1979. Disciplines covered were meteorology, physical oceanography, hydrology, and related satellite data. In 1980, the World Meteorological Organization (WMO) started an INFOCLIMA system under which dataset information was gathered from many countries. It includes selected manuscript data as well as digital data. A thick document is available; a searchable floppy disk version is expected in mid 1990.

Other approaches to the on-line discovery and access to databases can be found in [BURT89, COTT86, WILL86].

NASA took on the task of developing a National Master Directory for the U.S. about 1987. (Sometimes it is called the NASA directory.) Under this program (for the broad spectrum of Global Change research), descriptions of datasets and data centers can be searched on-line. It can be readily reached from Internet (NSFnet, Span, etc.). The concept has been to keep the data descriptions relatively simple at the national catalog level. About 1000 datasets are included. More detailed inventories can be viewed at the data centers. In many cases, the National Directory can pass search control to a local data center. In this way, a user of the National Directory can ask to be connected to the on-line system at JPL or to a NOAA center, etc. Some of this ability to pass control has been implemented; parts are under development.

Many collections have free text descriptions of content that are analogous to the traditional abstract. It is not evident that these descriptions are sufficient for the browser to use them as a card catalog. This is especially a problem for a scientist looking for information which was collected by another discipline, and likely not for a purpose he is familiar with.

2.4.2. Recommendations

(1) The discipline of "information science" has established many precepts associated with the location of relevant data. Neither it, nor the information retrieval research community should be ignored when approaching the issue of scientific database management.

(2) Individuals are encouraged to ensure that significant databases are described in the National Directory. (Some feel that such voluntary participation is too weak, and would urge an unspecified form of coercion.) Agencies should adopt policies that ensure that descriptions of data collections are entered into this on-line catalog.

(3) There are several forms of query that recur within the scientific community:

a) identity to key value
b) identity by synonym list to key value
c) similarity to key value
    1) text
    2) number
    3) coordinates (space and/or time)
    4) subsequence in series, or sequence
    5) proximity in a (mathematical) graph.
d) recursive application of a rule (e.g. moving down a hierarchy to its tips, or leaves)
e) recursive subcomponent matching

a) and b) are easily implemented. Effective implementation of c), d), and e) will require further research.

(4) We need further experimentation with access to databases, with advanced systems and methods, to ensure that scientific users can indeed find relevant databases.

2.5. Standards

Standards are crucial to all scientific activity. They facilitate the exchange of information and ideas. In a sense, the standards imposed by a discipline literally determine the nature of the discipline and its ability to communicate with other areas of science. While standards may be important, it was noted that they should not be imposed "from above". (This would seem to contradict the conclusion in [NATI88] p.85.) Many of the best standards evolve through usage. In particular, it was observed that standards should not be arbitrarily imposed from above by federal agencies. Professional societies should play a strong role in establishing suitable standards for their associated disciplines, which should then be supported, or possibly enforced, by relevant funding agencies. These societies should work together to have consistent standards in areas where there is overlap. It was observed that:

(1) Standards of database identification across all disciplines will go a long way towards solving the database citation issue.

(2) Standards of terminology can be established by means of lexicons. But such lexical standards should probably be restricted to individual disciplines, or even specialities within a discipline.

It should be possible to access the meaning (some standard interpretation) within any lexicon.

(3) Lexicons should be "user friendly". There should be mechanisms that translate the various terms an end-user may employ into the terms of the database. These equivalences themselves should be accessible.

Possibly the ultimate user friendly lexicon would be a comprehensive one based on regular English. But emphasis on this at this time seems a bit premature. Instead we should emphasize the standardization of special sublanguages for the communities that use them, with some attempt at making them accessible to a more general public as well.

2.5.1. Current Status

There are many different lexicons in use in varying disciplines — more research is needed into the application of standards.

2.5.2. Recommendations

Lexicons should have not only hierarchical (broader, narrower) and simple related (e.g., 'see also', 'see related') connections, but other suitable relationships for the scientific domain (e.g., "site" of a disease in a medical collection). Work on relational lexicons and standards for them are an important area of investigation. Whenever possible, relationships across disciplines (e.g., work on Unified Medical

Language, or Energy Vocabulary) should be done in an anticipatory fashion and should be used.

2.6. Education

The advent of new approaches to scientific enquiry, and of new technologies, does not ensure their use. People must learn how to use them. The role of education is to convey discoveries, techniques, etc. to others. Education should be a fundamental component in developing the use of scientific databases.

It is uncertain whether formal education is the right way to teach about scientific database use. The way most of us learned to use a library was offered as an analogy. Except for a brief introduction to the card catalog in high school, most mastered the retrieval mechanism on a trial and error basis.

2.6.1. Current Status

(1) Formal courses regarding the design and use of database technology are non-existent outside of computer science and information science.

(2) Database theory, as taught in most computer science curricula, is virtually irrelevant to practitioners who want to design and use many scientific databases.

2.6.2. Recommendations

(1) NSF sponsored workshops to spread an awareness of database potential.

(2) NSF should encourage development of multimedia and/or interactive training products that will make students, researchers, and other professionals aware of the availability and utility of scientific databases.

2.7. References

[BELK87] N. Belkin and W. B. Croft, ‘‘Retrieval Techniques’’, in *Annual Review of Information Science and Technology*, vol. 22 , Elsevier, 1987, 109-145.

[BURT89] H. D. Burton, The Livermore Intelligent Gateway: An Integrated Information Processing Environment, Vol. 25, 1989.

[COTT86] G. A. Cotter, ‘‘The DoD Gateway Information System: Prototype Experience’’, *Proc. of the Seventh National Online Meeting*, New York, May 1986, 85-90.

[EVEN88] M. Evens, *Relational Models of the Lexicon*, Cambridge University Press, 1988.

[FOX87] E. A. Fox, ‘‘Development of the CODER System’’, *Information Processing and Management 23*, 4 (1987), 341-366.

[GROS88] P. Grosbol, ‘‘Generalized Extensions for FITS’’, *Astronomy and Astrophysics Supplement 25*, 6 (June 1988).

[ISO86] ‘‘SGML: Standard Generalized Markup Language’’, ISO 8879, ISO, 1986.

[ISO87a] ‘‘Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)’’, ISO 8825, ISO, 1987.

[ISO87b] ‘‘Specification of Abstract Syntax Notation One (ASN.1)’’, ISO 8824, ISO, 1987.

[MARC81] R. S. Marcus and J. F. Reintjes, ‘‘A Translating Computer Interface for End-User Operation of Heterogeneous Retrieval Systems; Part I: Design; Part II Evaluations’’, *J. of the American Society for Information Science 32*, 4 (July 1981), 286-303.

[NIST90] *NIST 1990: Report on Hypertext Standardization Workshop*, NIST, Gaithersburg, MD, Jan. 1990.

[NATI88] *Report of the Committee on Mapping and Sequencing the Human Genome*, National Research Council, Washington, DC, 1988.

[WELL81] D. Wells, ‘‘FITS: A Flexible Image Transport System’’, *Astronomy and Astrophysics Supplement 17*, 6 (June 1981).

[WILL85] M. E. Williams, ‘‘Electronic Databases’’, *Science 228*, 4698 (Apr. 1985), 445-456.

[WILL86] M. E. Williams, ‘‘Transparent Information Systems Through Gateways, Front Ends, Intermediaries, and Interfaces’’, *J. of the American Society for Information Science 37*, 4 (July 1986), 204-214.

3. Emerging and New Technologies

Panel members:

Don Batory, Univ. of Texas
Joe Bredekamp, NASA
Mike Carey, Univ. of Wisconsin
Y.T. Chien, NSF
Glenn Flierl, MIT
David Kingsbury, George Washington Univ. Medical Center
Arie Shoshani, Lawrence Berkeley Lab (chair)
Ferris Webster, Univ. of Delaware
John Wooley, NSF

3.1. Scope of Scientific Database Problems

3.1.1. Data

Datasets may be modified with operators to transform them into forms which are more appropriate for a given purpose. Though ideally the datasets formed at each stage in such a process should be saved and archived, in reality, often only the latter stages are preserved, because of cost.

As successive operators are applied to the dataset, in general the degree of global dependency of the data point increases.

In the summary that follows, only those data types (in general, digital) that can be treated with computers are discussed. There is a large class of analog data types (for example, strip-chart recordings, deep-sea cores ...) that are not generally amenable to scientific database management without some kind of conversion to digital format.

Engineering data

These are generally raw values in engineering units (like voltages) that frequently are not kept.

Operator: Convert engineering values to physical values (like a temperature).

Physical values

These are the uncorrected values in the units of the physical value of interest. Datasets of uncorrected physical values are sometimes saved.

Operator: Correct data to a calibration curve, or adjust to values derived from another source.

Calibrated data

The raw physical values may be adjusted with other values, and corrected with calibration operators. Datasets of calibrated values are normally preserved. Datasets of this type are frequently used for scientific purposes.

Operator: Remove outliers or provide other quality assurance procedures.

Validated data

The calibrated dataset may be passed through quality assurance procedures which may remove outliers, correct data points according to some algorithm, and transform the data to match the constraints of other measurements. The validated datasets are the ones most commonly used for scientific purposes.

Operator: Aggregate, average, correct, or conformally map the dataset.

Derived data

The calibrated or validated datasets may be aggregated in some way, such as in gridded or averaged values. Products such as maps or graphs may be prepared from the data.

Operator: Construct new datasets using physical models to constrain the data fields.

Interpreted data

Datasets may be constructed from validated datasets together with physical models to produce derived sets that meet the constraints of some kind of physical model. Such datasets often have global dependencies, according to the type of model that is used. Procedures to produce interpreted datasets using dynamic models are commonly used in the atmospheric sciences to produce global datasets.

3.1.2. Metadata

A functional definition is: Metadata is information required to identify datasets of interest, their content, validity, and sources. There is no agreed-upon standard formulation of an ideal metadata set. There is general agreement that most databases suffer from incomplete metadata. Examples of the kind of information that is valuable (essential) to retain with the data are:

General identifying information such as: who collected the data, when the data were collected, and where the data were collected.

Characteristics of the device(s) that collected the data.

Transformation operators (e.g., calibrations) applied to the data.

Programs used to manipulate or modify the data.

Models used in processing or interpreting the data.

Documentation relating to or derived from the data along with technical manuals related to the data source and relevant publications, reports, and bibliography.

3.2. Desirable Data Types and Manipulation Operators

3.2.1. Data Types/Representation

Database systems for scientific use must be capable of accommodating a wide variety of data types and data representations. In addition database users require the tools to manipulate these data in a wide variety of ways. The following list represents the major groups of the data types that will be required by a scientific database. The list is not considered exhaustive, but represents the major classes of data and manipulations common to most scientific disciplines.

(1) Individual values. This group is comprised of alphanumeric character strings, including integers, floating point, date, etc. These data would be subject to the usual set of manipulations (eg. sorting, arithmetic functions, searching, etc.).

(2) Two-dimensional images, including both graphical and grey scale images. Typical manipulations of data of this type include, smoothing, feature extraction, enclosure, comparison (differencing), data extraction, segmentation, and contouring. An example of data of this type is a digital representation of an autoradiogram. A sample manipulation would be the extraction of the position and intensity of a band from an autoradiograph which contains many bands. The extracted information will then be used in an application or placed in an individual value field(s) of the same or another database.

(3) Three dimensional images including computed simulations of data such as electron density maps and images from computational models, for example, global weather maps. Data of this type would be subject to manipulations such as rotation, stereo comparison, planar slicing, and spatial transformation. Images of this type would also be subject to data extraction where possible, and the resulting extracted data might also be a 3-dimensional image.

(4) Spatial objects such as representation in a geographic information system and vector graphics. These representations would be subject to manipulation within GIS applications as well as further vector graphic manipulation and subobject extraction, and containment.

(5) Text has been identified as a unique data type in addition to the alphanumeric fields mentioned above. Text might play a unique role in databases, especially in the representation of associated ‘‘meta-data.’’ Text is subject to string matching, keyword searching, approximate matching to find other related materials, frequency counting, and other text manipulation procedures.

(6) Sound is a unique data type collected in databases of such items as bird songs, whale communication and human voices. Sounds are subject to comparison (sound pattern matching for identification purposes) and to spectral analysis.

(7) A data type identified by the term ‘‘blob’’ was recognized as being a discrete entity in many database systems. Blobs are stored in a database field and may be directly retrieved (as data type blob) but are not subject to any further manipulation in that form. In some cases a blob may be subject to modification by an application at which time it would be transformed into another data type.

3.2.2. Type Extension

It was judged that modern databases would include some mechanism for type extension as part of the database system. In this case type extension includes a variety of both user defined and query defined functions. This results in a modular approach to database architecture where user defined functions may be imported from a collection of potential tools. This functionality would lead to the development of derived fields, type extensions and type conversion.

3.2.3. Type Constructors and Associated Operations

To define the schema for any scientific database, one will need to employ a number of type constructors to capture the complex structure of typical scientific data. Some are straightforward and exist in today’s commercial database systems, such as the well-known record and set constructors; in such cases, existing data manipulation languages (DMLs, or query languages) are capable of dealing with them. Others will require the addition of new DML operators in order to make full use of their capabilities.

Both directed and undirected sequence constructors will require additional operators such as precedence operators, string operations, approximate match searching, and multi-sequence operations. Time-series constructors will be subject to sampling, time series analysis and transformations (e.g., spectral analysis). Multidimensional array constructors will be examined for adjacency, multidimensional searching, collapse of dimension, and be subject to matrix mathematical operations.

Another type-related facility that was viewed as important is the notion of inheritance, an idea with roots in semantic data models and object-oriented languages that is now being embraced by the majority of the database community. Inheritance simplifies the task of defining new types that are related to other, existing types in the database schema; it also simplifies the design of application software in some cases, since routines to process objects of a given type T can also process objects whose types are a subtype of T.

3.2.4. Composition and Nesting

In order to define complex structures that represent the variety of data structures needed in scientific applications, the ability to compose and/or nest the above structures is needed. For example, a contour can be represented as a (circular) sequence of points in 2-dimensional space. Similarly, there are biological structures, such as maps that can be represented as sequences of intervals. The ability to compose and nest constructors to form such structures is useful for the generality of defining data structures and operators over them.

3.3. Scientific Environment

Scientific database users work in a wide variety of environments, from a single personal computer or workstation, to supercomputers, to networked heterogeneous systems. Correspondingly the needs for flexibility, efficiency, and capability vary dramatically. But, more and more frequently, scientists find their work requires data of many different kinds from many different sources. Scientific database systems must deal with the complex and changing working environment, must locate and retrieve information in a form that the requestor can interpret, and must be able to manipulate data of varied types.

3.3.1. Client-server Model

One approach which has proved fruitful in commercial distributed data systems and in other problems involving heterogeneous hardware/ software systems has been the client-server model. In this model, unlike the case where a program reads data directly from a disk or tape file, a ‘‘client’’ requests information via a message sent to a ‘‘server.’’ The data is then returned via messages (inter-process communication links). The principal advantage of this approach is that the client programs become insulated from the hardware details and from knowing whether the data is stored locally or remotely. Servers can also translate internal word structures or even data structures as the information is passed along. Many of the vexing data interchange problems can be made transparent to the users.

3.3.2. Networking

In many cases, when the requestor is adequately informed about the data sets and is suitably prepared to deal with the format, high volume transfers of data may still involve shipping of tapes or other storage media. For exploratory work, however, a distributed environment is common and a suitably high-speed network is required. SDB’s should be able to communicate efficiently and effectively over networks.

3.3.3. Private Software Interlink

One cannot expect that any new system should or could be adopted by all. Private databases will still be necessary for efficiency in processing, for convenience, or for historical reasons. New SDB systems must support links to existing systems so that users can either export their database to others or import other data into their systems.

3.3.4. User Interface Simplicity

User interfaces are, or should be, highly personalized entities; one need only observe a ‘‘discussion’’ of the merits of different operating systems or editors to realize that individuals seek tools which best fit their working preferences. Simple and intuitive user interfaces are highly desirable; on the other hand, one should not sacrifice capability. A SDB should not be restricted to a single user interface.

3.3.5. Heterogeneity

Data bases will be different in many aspects, for example, data models, scheme description, disciplines, and machines. For the scientific user, a system would ideally hide these differences and permit access to the data and to information about the data with minimal impediments.

3.3.6. Database Mostly Add-on and Read-only on Servers

For a large number of applications, the basic data is not altered by the majority of users. In some cases, it may not be changed at all, although new versions may appear; in others, a single group is responsible for correcting the data. In other cases, the major change to a data base is the inflow of new data. Thus many of the issues concerning commercial database designers-- updating, concurrency, transaction rollback-- may not be important for SDB's.

3.3.7. Multi-server Coordination/ Notification of Clients

One caution, however, is that it is not uncommon for there to be multiple copies of a data set, and it is necessary to ensure that these remain consistent. Furthermore, a mechanism for alerting users to changes in the data set may need to be considered. This may take the form of a notification the next time the data set is accessed or a notice on a bulletin board.

3.4. Types of Users of Scientific Databases

Among other characteristics, scientific databases are defined by a more diverse user community than those for a typical business database. The actual types of users of a given scientific database are very likely to evolve with time, and in some cases, even the most-frequent category of users might change with time. For the sake of completeness and simplicity, users are defined broadly from the creators to the end users. The first individuals to desire access to the database will presumably be the individuals involved in collecting the data in the first place. For many applications, the data Collectors are likely to be the major end users and thus their needs will set the content and details of the expected applications. For an emerging field that is rich in experimental content, but poor in theoretical formulation, the expectations of the data Collectors will be paramount, but if a theoretical basis for understanding the phenomena under study exists, Analysts, notably including modelers, will also place constraints on the requirements set for developing a new scientific database. Utility to a wide community over the time period for which the data will be of value will depend in part on the Archivists, who will have expectations in terms of stability, efficiency of updating, access to earlier versions, and other requirements. The next level user will need to be intradisciplinary and interdisciplinary Integrators who ensure appropriate standards are met; namely, standards that provide access by scientists not expert in the original experimental system. Finally, Administrators who check the quality of the system, conduct time checks, and other management functions, will be users as well.

On the whole, the diverse list of users underlies the central issue. The entire user community must be taken into account in the development of the initial database. The principle difficulty is that the individuals utilizing the data at a latter date are likely to come from disciplines other than the data Collectors or any of the early users. A change in the approach to the development of new databases is necessary to accommodate this difficulty. In particular, it is clearly easier to ensure that essential data do indeed impact on disparate communities if the interdisciplinary Integrators are considered in the initial design phase. The challenge for these Integrators and for all others in the database design and implementation is to ensure sufficient flexibility and perhaps the archival of adequate metadata, to accommodate the needs of unanticipated user communities.

3.5. Current and Emerging DBMS Technology

Our panel identified a number of examples of existing technology which are relevant to scientific data management problems. These include flat files, hierarchical and network DBMSs, relational DBMSs, special-purpose data managers such as image or graphical data managers, information retrieval systems (for text searching), and hypertext and hypermedia systems. We also identified new trends in database management systems such as extended or extensible DBMSs, object-oriented DBMSs, and logic-based DBMSs. We first briefly review what each of the current technologies provides for handling scientific data problems, in terms of the kinds of objects and manipulation requirements discussed previously, and we attempt to point out where they fall short of meeting the full needs of scientific data

management. We then address the question of what emerging database technology has to offer in the area of scientific database support.

3.5.1. Current Technology

Historically, data was first stored in the form of flat files, and much data is still stored that way today. The obvious advantage of flat files is that they can be used to store any type of data. The clear disadvantages, which led to the development of DBMSs, include lack of any structure or semantics associated with the data. Thus, any of the data types discussed earlier could be stored in flat file form, but the burden of manipulating them would lie completely with the application program. Hierarchical and network based DBMSs improve on flat files by offering support for managing records rather than unstructured data, and also provide notions of relationships (e.g., CODASYL sets) and indexing. However they are still relatively low level data managers, requiring application programs to explicitly navigate through the data (record by record, essentially) based on its logical and/or physical organization.

Relational DBMSs are the current focus in the commercial DBMS world. They offer a very simple logical model for data (tables) and a more or less declarative query language (SQL) which hides the physical details of the data. In terms of the data types and operations discussed earlier, all three types of DBMSs basically provide support for records, where the fields are individual values, and for sets of records. Hierarchical and network DBMSs provide some support for dealing with data which has complex structure, e.g., a department that ‘‘contains’’ a set of employees, but (as just mentioned) provide only low-level programming facilities for manipulating such structures. Relational DBMSs, in contrast, can actually make it somewhat harder to represent such structured objects due to the simplicity of the table abstraction. However, they provide large benefits in terms of programmer productivity and software maintenance by not letting users/programmers see the physical structure of the data.

None of the three basic types of DBMSs provide any inherent support for new/specialized data types or operations, e.g., no support for images, text, arrays, time series, etc. However, relational DBMS vendors are now beginning to offer support for blobs and for user-defined or ‘‘abstract’’ data types (ADTs) and associated functions. Clearly, this is an important step in the direction of handling such data types as images or text, and will enable relational DBMSs to have such data as field values in records. However, as will be discussed in the section on emerging technology, more research is needed (and is currently going on) in this area, as efficiently handling such data requires a fair amount of support at the levels of indexing, query processing, and query optimization. It should also be noted that ADT support does not address the need for a richer set of type constructors (like arrays, time series, sequences, etc.), so ADTs, while important, do not solve the problems of handling the complex objects anticipated in the scientific data management area.

Information retrieval systems, which have developed and matured separately from DBMSs, provide support for text-intensive applications. For example, they are widely used in application areas such as library science, allowing on-line catalogs to be searched based on attribute data (e.g., author) or on textual attributes (e.g., title or abstract searches). In the latter case, searches are specified as predicates involving keywords of interest to the searcher. Information retrieval systems are not DBMSs, however, so they do not provide support for much of what a DBMS provides, e.g., they do not handle non-text data, range queries, etc. In order to provide effective support for scientific meta-data and browsing, it appears that information retrieval technology must be integrated with more general (e.g., relational and beyond) database management technology.

In addition to general-purpose DBMSs and the more specialized information retrieval systems, there are also special-purpose data managers capable of handling particular types of data such as image data or spatial data (e.g., GIS packages). While such data managers are certainly useful for managing their particular data types, which are indeed related to the kinds of data that some of the sciences do need to manage, they are too specialized to be useful as general-purpose solutions to scientific data management problems. They are perhaps most important as good sources of examples of the sorts of functions that are likely to be needed in scientific data managers for handling images and various spatial data types.

The final entry on our current technology list, hypertext and hypermedia technology, is relatively new (at least in a commercial sense). Products such as Hypercard exist, but research efforts are underway to provide systems with much richer function (e.g., the Intermedia project at Brown). What hypertext/hypermedia systems seem to do best is support nonlinear or browsing style access to data. These systems provide support for data objects that have links into other data objects, and appear very promising as a technology for applications such as on-line encyclopedias, document editing, etc. They generally provide support for interaction-oriented data types, e.g., text, graphics, and perhaps sound. As for information retrieval systems, they seem to hold promise in terms of dealing with meta-data, but do not offer ‘‘core’’ database functionality such as managing large data sets, querying them, etc.

3.5.2. Emerging Technology

There are four basic approaches that are being taken by database researchers today to extend current relational technology. Some of the basic concepts that appear relevant to scientific databases (e.g., new data types) are handled by most; however, no single approach fully embraces all of the requirements we have identified in our discussions. We briefly explain these approaches below, citing their potential advantages and disadvantages.

Extended relational. Relational DBMSs have provided an enormously valuable tool for handling conventional database problems; the treatment of concurrent transactions, recovery, nonprocedural query languages, and query optimization have found wide spread acceptance in the business community. The extended-relational approach is to build upon this success, by extending the relational model (and relational DBMSs) to incorporate new features. Among the features discussed include: inheritance, user-defined functions and data types, procedures as data types, nested relations, and support for triggers.

One of the advantages of extended relational DBMSs is a more likely acceptance by the database community; the advantages of the relational model (e.g., its basis in set theory) are retained and only a minimal set of changes need to be provided. The disadvantages of the approach is that the set of ’minimal’ changes may not be sufficient for, in our case scientific databases, or that the relational model itself is not adequate despite modifications.

Object-oriented. Among the principle advantages of object-oriented languages are the features of encapsulation and inheritance. They have been found to significantly simplify the development of complex software systems. For this reason, the languages of choice in developing specialized database applications (e.g., VLSI layout editors) are object-oriented. It turns out that a significant part of these applications is a data management component to make data that is internally generated into persistent data (e.g., CIF files) that can live across multiple application executions. The object-oriented DBMS approach offers an integration of object-oriented programming technology with (relational) database technology.

There are many benefits of this approach. Persistent data types can be declared, thereby removing the burden of data persistency from application programmers. New data types and their operators can be defined by users and be understood by the OO-DBMS in a clean and seamless way. Yet another benefit is the practical realization of the useful ideas of entity (i.e., object) and generalization (i.e., inheritance) that has been the hallmark of semantic data model research. Perhaps the primary disadvantage of the approach is its immaturity. Presently, there is little first-hand experience with these systems.

Extensible DBMSs. No DBMS can satisfy all applications equally well. The extensible DBMS approach, sometimes called a DBMS toolkit approach, is aimed at the tailorization of DBMS internals (e.g., storage structures, query optimizers) to meet the requirements of applications that are not adequately served by existing DBMSs. Extensible DBMSs is really a study of the architectural issues of DBMS construction to make them more ’open’, that is, to make customization of DBMSs an easier task.

There is no single approach that can accomplish this goal; advances in a variety of fronts have been recognized. The admission of new relational operators (e.g., sampling) into a DBMS requires, for example, modification of the query optimizer to know how to optimize queries that reference the

operator. One of the advances in extensible database research is the use of rule-based optimizers, where the optimization strategies associated with an operator can be precisely explained in terms of compact rules.

Another advance is the idea of composing DBMSs from prefabricated software building-blocks. There seems to be a central core of ideas that are found in many DBMSs - from network to object-oriented - that are constantly being reinvented. By providing tools to write generic modules once and to compose them, and by employing carefully engineered component interfaces that are designed to simplify the integration of new modules, some of the more difficult aspects of DBMS construction can be simplified.

The primary disadvantage of this approach is that DBMS customizations will have to be done 'at the factory', requiring a designer/implementor with expertise in DBMS implementations. That is, the toolkits are unlikely to be targeted for the end-user (i.e., scientific database community).

Logic based. Conventional relational query languages, such as SQL and its clones, are examples of limited first-order logic data languages. A natural generalization of these languages is to use prolog/datalog as DBMS query languages, thereby providing a significant increase in query expressibility. Typically, complex queries - even those involving recursion - are expressed elegantly and compactly. Queries are processed via sophisticated inference engines which provide a wide range of optimization possibilities for complex rule sets.

Among the advantages of this approach is its solid theoretical basis. For expressing and studying recursive queries, there is no better formalism. At the same time, the approach addresses only the recursive query problem, and provides yet another language that scientific database users must learn (and that must somehow be coupled with existing programming and query languages employed by the scientific database user community).

Summary. The following is a compilation of some DBMS prototypes that are available, along with their features:

System	Extended Relational	Object Oriented	Extensible	Logic-Based	New Data Types
Exodus		x	x		x
Genesis	x		x		x
LDL	x			x	
Postgres	x		x		x
Starburst	x		x		x

3.6. Recommendations on Various Issues

3.6.1. Core Data Types and Operators

Scientific database systems should support a set of core scientific data types, compositions constructors, and manipulation operators. The selection of these core capabilities should be determined by their usefulness to a large number of scientific applications. They should be part of an optimization procedure, and should be implemented to execute efficiently. These core capabilities should be powerful enough for defining additional complex structures and operators in terms of the core capabilities.

3.6.2. Extendibility

Scientific database systems should be extendible, because of the variety of complex structures required in scientific applications. The systems should facilitate the addition of new data types, new data structures, and new manipulation operators. It should be possible to describe these additions in terms of

the core data types and operators described above when appropriate.

3.6.3. Conceptual Modeling

The user models for scientific databases should be at the conceptual level (e.g. objects, relationships) rather than at the logical level (e.g. relations, record types). The models should at a minimum support various scientific data types (spatial, temporal, sequence, etc.), the aggregation and generalization constructors, and facilities for integrity constraint specification.

3.6.4. Heterogeneous Databases

Support for heterogeneous databases is essential for most scientific applications. The interface and transformations of databases should be specified at the conceptual level, i.e. in terms of objects and relationships between objects. Data reformatting between systems should be done through data format standards, so that each subsystem needs to have only two translators: to and from the standard.

3.6.5. Interoperability of Software

A scientific database environment should facilitate the interoperability of a variety of software components, such as statistical analysis software, graphical display software, data management software, metadata browsing software, etc.

3.6.6. Exchange Standards

Exchange standards have been proven to be extremely useful in specific scientific disciplines. The development of standards between multiple disciplines is essential to the efficient interaction between such disciplines. It is recommended that these standards should concentrate on data exchange formats only, since trying to achieve agreements in other areas (e.g. query languages) may be unworkable. Data exchange standards should be self-describing. Also, these standards should be extendible, so that arbitrary data streams can be included as ‘‘uninterpreted data.’’

3.6.7. New Technologies

Emerging new technologies, such as Object-Oriented Database systems, Extensible Database systems, and Logic Database systems seem promising, but it is believed that each has different advantages. Thus, an integrated technology that enjoys the joint benefits of such technologies should be encouraged.

3.6.8. Metadata

Scientific database systems should have powerful metadata modeling and access capabilities, because of the complex nature and quantity of metadata in scientific applications. These capabilities should include: support for subject hierarchies, taxonomy hierarchies, keyword search, text search, and browsing.

3.6.9. User Interfaces

Flexible user interface development tools should be made part of a scientific database environment. The specialization of user interfaces for scientific applications is necessary for the efficient learning and use of such systems. User interface development tools should permits easy development of browsing, query, and manipulation capabilities, by using menus, icons, multiple windows, graphical displays, etc.

3.6.10. Emphasis on Applied Problems

Research and development projects should be encouraged to use and cooperate with real scientific applications. Realistic scientific problems can best be abstracted from practical scientific database needs.

4. Core Tools

Panel members:

Vernon Derr, NOAA
Nancy Flournoy, American Univ.
Greg Hamm, Rutgers Univ.
Anita Jones, Univ. of Virginia
Bob McPherron, UCLA
Frank Olken, Lawrence Berkeley Lab
Peter Shames, Space Telescope Science Institute (chair)
Maria Zemankova, NSF

4.1. Objectives

The goal of this panel was to assess unfulfilled requirements for ‘‘core tools’’ in scientific data management, i.e., tools not available, or not easily usable, in present database systems. Considerable differences in tool requirements were noted across the different scientific disciplines considered; even more significant differences were seen between large facility projects and smaller, single-laboratory efforts. Even so, common objectives span all levels, including:

- Enable data management — Some scientific data will be managed in complex DBMS environments, others with simple flat file archives. A spectrum of tools spanning the range of environments is needed to encourage and facilitate good data management. Unless the intellectual energy required to archive and manage data can be significantly lowered, scientists will be reluctant to invest in these tools beyond their own immediate needs.

- Enable communication and standardization of data — Interdisciplinary work will only be possible if considerable standardization of data and nomenclature are achieved, first within disciplines and then beyond. Database tools which support the construction of thesauri, controlled vocabularies, and easy exchange of data will be essential to facilitate this process.

- Support computational science — Most scientific database systems are built to support the needs of the investigators who designed the experiments involved. As data resources grow, considerable science will be done using these resources as experimental material. This use may generate requirements beyond those obvious to the original generators of the data, and implies a need for tools which support the painless incorporation of annotation, audit trails and other metadata needed for later, unanticipated re-interpretation of data sets.

- Improve retrieval and analysis capabilities — Classical DBMSs provide significant tools for the systematic organization of data, but are restricted in terms of their representational power for abstract data types. This limits possibilities for ‘‘semantic’’ retrieval of data, and for analyses which require embedded knowledge about data.

- Provide integrated data management and analysis environment — a variety of data analysis tools are in use in the different disciplines. Some of these are user built, others are created and distributed by discipline centers to a wide community of users. Ideally, access to data management facilities from within these analysis tools or environments should be possible, so that the tools can be readily applied to the data.

The panel thought these objectives implied a need for improving the core tools at three levels:

- DBMS/OS — Considerable improvement is needed in terms of the representational power, manipulative capability, and interface definition of present DBMSs and operating systems. Alternative data models may be required to support the specific needs of scientific, as opposed to commercial, databases.

• Utilities and other tools — Much of the friction in scientific databases is generated by conceptually simple activities such as data laundry, reformatting, interpolation, sampling, scale matching, etc. There is also a direct need for tools to help scientists take more responsible action in archiving, annotating, and depositing their data for public use without a large increase in the effort required. Considerable improvement could be made by making available well-conceived sets of small tools needed for these sorts of tasks.

• User interfaces — All of these activities will be useless if they are not actually done, and they will not be done if they are not easy to do. No scientist will spend six months learning to use data management tools solely to improve the general state of the science. This implies a need for great improvement in user interfaces to enable facile use of both of the other sets of tools mentioned above.

Achieving a system that will support multi-disciplinary research across a variety of databases and archives will only be possible if a number of system infrastructure elements are in place. The panel made certain assumptions about these elements and they are best stated up front. If all of these elements are not provided as part of the distributed system then many of the innovative science data access goals may not be met.

The specific infrastructure elements are:

• Wide Area Networks - networks which are widely installed and supported, and are of sufficient bandwidth and reliability are essential for the successful deployment of a distributed system.

• System Interfaces and Tool Kits - the distributed system must run on a variety of different hardware platforms that will evolve over time. Integrating new tools and new sites must be easily performed. An interface toolkit that can accommodate the integration of this heterogeneous and evolving mix of systems must be provided.

• Data Exchange Formats - standard formats for data exchange must be identified or developed. Some discipline specific models, such as FITS in astronomy, already exist. Others must be identified or adapted from various standards efforts if easy data exchange is to be possible.

• Locality of Control - it is essential that the system infrastructure support local control over data and computational resources, while at the same time providing for remote access to these resources. The infrastructure should support this access and also help preserve site integrity.

4.2. Recommendations

The panel identified several specific recommendations for research which are discussed in more detail in the following sections. These recommendations are:

Support the creation (or expansion) of one or more data analysis environments for science data that includes database and data archival processing. This should include a sophisticated user interface that can unify existing tools, yet be flexible enough to be:

- domain specific
- handle differing levels of user sophistication (novice/expert)
- browse across different, distributed DBMS's
- provide ''hooks'' for special application programs
- access a hierarchy of storage in a user-transparent fashion
- support tracking of data accessed across multiple DBMS to maintain an audit trail of transformations applied to create each dataset.

Support the development of an integrated set of user analysis tools, to include:

- data search, location, and access
- file and directory manipulation
- file and data editing

- data transformation and processing
- data display and visualization
- statistical and other analyses

Fund research to design, prototype, and apply alternatives to the relational data model. The resulting model will require a theoretical underpining and sufficient richness to support some or all of:

- temporal data
- spatial data
- sequences
- graphs
- structures formed of these objects
- associated meta-data and descriptive information

Encourage long-lived facility class organizations to develop and distribute, free to their community, software tools that are engineered to promote:

- standard interchange formats
- standard application interfaces
- standard analysis packages
- hooks for extensions

Support the creation of more and better location and retrieval capabilities including:

- A high level directory of sites, databases, and services to facilitate access to the system by new users and experienced users alike.

- Discipline specific data directories and repositories for those fields where such facilities do not already exist.

For all these recommendations, due consideration must be given to the support and operation of the required network, repositories, directories, interfaces and other infrastructure elements. Moreover, research efforts should give priority to the production of prototypes which support actual applications with reasonable efficiency.

4.3. Data Modelling

The relational data model is built on finite set theory. Many types of scientific data do not fit this model very well. Attempts to shoehorn scientific data into the relational model lead to obscure schemas and queries, and (often) the inability to effectively ask or answer certain types of queries.

Investigation of alternative data models and their implementation is appropriate.

In particular, we note that much scientific data consists of observations of functions defined over continuous domains (e.g., space and time). Thus interpolation, for example, is a natural operator. We should have data models which can readily accommodate these kinds of data and operators.

We also need data models which include the notion of sequences of records, not simply sets of records. Sequences are ubiquitous in molecular biology (e.g., DNA sequences), econometrics, statistics, and signal processing (e.g., time series).

4.3.1. Implementation of alternative data models

Implementations of such alternative data models are required, because we do not expect that these models can be effectively implemented with existing relational DBMSs. Practical experience with applications on real DBMSs is the ultimate arbiter of the utility of new data models.

Object oriented database systems offer one possible mechanism for supporting scientific data models. However, providing a mechanism for constructing a scientific DBMS is not sufficient for most applications. We believe that scientific DBMSs must provide support for functions on continuous domains, and sequence data, etc. This should not be left as an exercise for the domain scientist designing the database.

4.4. Integration of Analysis Tools

The scientific enterprise is characterized by generality. Operations and procedures are infrequently performed. Their order of application is highly variable. It is difficult to predict in advance what will be done. The information and data required may come from a wide variety of sources. These characteristics impose constraints on the DBMS used to organize the data required by a scientist. Commercial data base management systems available today do not work well for this purpose.

The state of the art in scientific DBMS is epitomized by the relational data base such as Ingres or Oracle. These systems have a firm foundation for operating on data sets organized as flat files. It is relatively easy to carry out operations such as joins, projections, concatenation, and selection via constraints. These systems are good at maintaining information on the state of the data base as a function of time. They provide a high degree of security and quality control over the data entered into the system. Relational systems are primarily used to manage data at a single site. Management of distributed data is done though the addition of networking and distributed processing capabilities added to the basic system.

These DBMS's do not support scientific analysis well. Sequential access to a large number of records can be very slow. Modification of existing relations, or the creation of new relations is difficult to do because of system security. The sequence of operations which modify the data base are not easily tracked, even though the state of the data base is accurately known. It is virtually impossible to enter annotations onto files, or include non alphanumeric data as part of the data base.

Some panel members felt that the operation of the DBMS should be kept distinct from the analysis and display system. The scientific enterprise is so general and broad that it is difficult to believe that one system can provide all potential applications for all scientific disciplines. It seems far more profitable to provide straightforward links to the DBMS which can be used by application programmers in developing separate, and quite specific modules. However, simplifying the scientists interface to eliminate the appearance of separate systems is mandatory to improve ease of use.

In this view there are four types of DBMS tools: inventory tools, management tools, access tools and logging tools. Inventory tools prove reports to users about the kinds of data sets, their structure, their location, their format and so on. Management tools are more specific to the system managers. They allow system administrators to move, copy, reformat files as complete entities. Access tools provide programmers and users with the ability to read and modify individual records in managed data sets. Logging tools provide a means for keeping track of the sequence of operations by which data originating in data sets in the system is modified to produce new data sets.

Future scientific DBMS should evolve towards a truly distributed system in which, from a single location, it is possible to apply all available tools to all managed data sets whatever their location. To do this in a convenient way for the scientist, and to do it efficiently the entities which the relational model supports should be generalized to include objects of greater complexity than presently supported. This necessarily implies that the relational operators used to create queries can be generalized to include user-defined operators.

Future DBMS should not try to do central management of distributed data. The people most familiar with the data and its limitations should maintain control over the data. Also future DBMS's should not try to centralize the data at one location and allow distributed management by the various data set owners. Instead, the future scientific DBMS should provide a means for doing distributed tracking of data. Necessary functions of such a high level system include:

- Inventory reports

- Access to data as whole data sets or distinct subsets
- Graphical displays for browsing data
- Textual displays for data and meta-data
- Data order and delivery service at macroscopic level

Ordinarily the data at various sites would reside in some specific DBMS which provides access to data sets at a microscopic level of individual records. The high-level distributed data tracking system should not depend on specific DBMS architecture at individual sites. It should, through appropriate standards and interfaces, make use of local system functions to accomplish its tasks in a manner transparent to the user.

5. Ozone Hole Case Study

Panel members:

Francis Bretherton, Univ. of Wisconsin
Jim French, Univ. of Virginia
Hector Garcia-Molina, Princeton Univ.
Tom Marr, Cold Spring Harbor Lab
Steve Murray, Harvard-Smithsonian Center for Astrophysics (chair)
Larry Rosenberg, NSF
Don Wells, NRAO
Greg Withee, NOAA

5.1. Objective

The objective of this case study is to consider one situation in which scientific data was a crucial component in a scientific inquiry. In this section we describe how the Ozone Hole was discovered and how scientists used data to investigate the seriousness of this threat to mankind. Then we point out the issues, the problems and the successes which involved the presence, absence and handling of various kinds of data. We think that this story particularly well illustrates both the importance of scientific data and the difficulties — technical and political — in applying it most effectively.

5.2. Problem Discovery

5.2.1. Early History

The Supersonic Transport project, that was suggested in late 1960's, raised concern about the impact of exhaust gases on stratospheric chemistry. Of particular interest was the production of ozone, the Chapman process. Little was known then about atmospheric chemistry. It was not a distinct scientific discipline. However, there was a belief that oxides of nitrogen would disrupt the Chapman process, depleting the ozone layer. And ozone concentrations were known to be intrinsically highly variable.

NASA organized a program to bring chemists together to work on the problem of upper atmospheric chemistry, particularly the chemistry of the lower stratosphere and of ozone production. Satellite experiments were planned and carried out to monitor the ozone concentration, and to measure the rudimentary composition of the upper atmosphere.

A large activity in atmospheric chemistry resulted. Upper atmospheric modelling increased, supported by fundamental laboratory research in determining the reaction rates of various processes and other necessary basic chemistry. Observations from a variety of sources were available, including atmospheric observations from ground based spectroscopy which had commenced in the early 1900's, remote satellite sensors in particular the Total Ozone Measurement System (TOMS) project, and in-situ lower stratosphere balloon measurements.

By the mid-1970's the main outcome of this activity was the discovery of the presence of high concentrations of chlorofluorocarbons (CFC's) in the upper atmosphere, and that free Chlorine — resulting from ultraviolet photo-dissociation — was the dominant mechanism which disrupts the ozone producing Chapman process.

5.2.2. Data Systems Comments

During this period when the problem was first recognized, the NASA sponsored activity was not widely known, particularly outside their community of atmospheric scientists. This lack of interdisciplinary communications hampered the rate of progress in recognizing the importance of the high

CFC concentrations.

The satellite sensor data was being saved within the Earth Science discipline, but it was not well organized, nor well documented. Of particular importance was the fact that the description of the data and how it was collected and filtered — the appropriate ‘‘meta-data’’ was not generated and saved.

5.2.3. Discovery

British meteorology teams had regularly monitored ozone concentration at Halley Bay, Antarctica on and off since the International Geophysical Year 1957. By about 1984, enough data were analyzed to produce a ‘‘hand’’ generated plot of the ozone concentration versus time from 1980. That plot showed a trend of reduced minimum ozone concentration each October, which is the Antarctic spring. This led to the conclusion that the ozone layer was being depleted. That is, a hole in the ozone layer over Antarctica was discovered.

5.2.4. Data System Comments

Sufficient data were available to see this effect in 1982, and even earlier in the satellite TOMS data. Delay in processing the British data might be attributed to the lack of awareness that there was anything unusual to be expected and a lack of a data policy that stressed rapid analysis of data. In actuality had the TOMS data been scrutinized in a straight-forward way, it would not have shown the depletion effect. The cause for this is discussed below.

5.3. Data Calibration

Retroactive review of the satellite data did not indicate a concentration decrease. The reason for this is that the raw sensor data was filtered by the ground processing system; a threshold was applied to remove ‘‘noise’’. This threshold was set high enough that it did not permit detection of the concentration decrease. Why?

The stratospheric models never predicted such a decrease. They had been used to help determine where to set the noise filter in the TOMS data. Once filtered the data were used to check the model. This circularity allowed the modellers and experimenters to believe that they had made the correct models and filter threshold choices.

Other NOAA data from radio sonde balloons were collected but not used in the analysis and model testing.

5.3.1. Data System Comments

Circular dependencies in modeling and data calibration led to a lack of correct data to test the models, and lack of recognition of proper constraints on the data they were using to test their models. Data analysis policies, or lack thereof, allowed useful data to be missed.

5.4. Understanding

5.4.1. Confirmation

Unfiltered satellite data was still in existence. It was re-analyzed, and this confirmed the existence of an ozone hole. It was unquestionably a real phenomenon.

Within the science community the question arose of what level of resources to invest in discovering what the danger to society and to the global climate was. A major initiative to investigate and understand this problem was begun. It involved multidisciplinary teams of chemists, atmospheric physicists, meteorologists, climatologists, etc. The teams needed access to data to facilitate improvement of models with more detailed testing. New data was collected from ground, air and space to continue to monitor the situation.

The result of this effort was the discovery that ice clouds in the lower stratosphere over Antarctica were the ‘‘villain’’. Ice and/or water phase chemistry leads to large quantities of free Chlorine being release into the stratosphere in the early spring (October) which then acts as a catalyst to deplete ozone concentrations.

The Antarctic is unique in two respects that make the ozone depletion most serious there. It is the only place where the lower stratosphere is cold enough during the winter to make ice clouds, and air circulation in the south polar region produces a wind vortex that prevents mixing of the stored Chlorine.

5.4.2. Data System Comments

The use of a variety of data sources from chemistry to global atmospheric monitoring networks, and ground, air, and remote space sensing were ultimately successfully. But this only occurred after a crisis had been generated. Routine combination of multi-source data is not ordinary operating procedure.

5.4.3. Consideration

Once the problem was thoroughly documented, the question arose of whether the Ozone Hole is a natural or man-made occurrence. There are multiple possible sources of Chlorine. Numerous industrial corporations across the world create chlorofluorocarbons. It was also recognized that volcanic emission of HCl from Mt. Erebus could be another source.

Again sources of information spanned disciplines. For example, ice core samples had long been taken to monitor long term temperatures through Carbon Dioxide analysis. But the samples might yield data on acidity, and Chlorine content if properly analyzed. A stratospheric aerosol and gas experiment was not initially intended to identify HCl, but re-analysis of data for this purpose might prove to be useful.

5.4.4. Data System Comments

The need for access to relevant data is severe. Much of the information is contained in data collected for other purposes or owned by investigators who have not published raw data in a form suitable for re-analysis. Even awareness of potential data for this purpose is limited, no master directory of data related to Antarctica in general or to the upper stratosphere existed.

5.5. Assessment of the Impact of Ozone Depletion

There are local effects. Increased ultraviolet flux occurs at ground level. It harms penguins. In addition, it kills plankton and krill (crustaceans) which are essential in the food chain of whales. Certainly the Antarctic stratosphere is connected to the general ecosystem and could over time influence global change.

Where else might ozone concentration depletions occur? In the Arctic and mid-latitudes it is not cold enough for ice clouds to form, but the atmosphere contains aerosols. If they include HCl, will there be the potential for ozone depletion on a global scale?

Further study continues. The data resources include:

- SAGE - monitor aerosols,
- LIDAR measurements of aerosols at selected sites, and
- models and theory - integrate data (fusion).

An international panel, the Inter-Governmental Panel on Climate Change, is meeting at regular intervals to review the status and impact of ozone depletion in the context of global change.

5.6. Mitigation

The saga of the ozone hole is not played out. The problem continues. Risk assessment on the potential magnitude of problem is needed. There is a trade-off needed to measure cost of reaction versus effect. And the data to support the decision process is needed.

This science story is representative of questions that will arise with more frequency. A problem will possibly afflict a population or environment segment of our world, or outside it. The question arises of what level of resource should be allocated to understanding that problem and assessing its risk. Data is only one portion of the puzzle, but it plays an increasingly important part.

5.7. Remarks

The saga of the ozone hole illustrates a variety of issues related to scientific data, its collection, maintenance, archiving and processing. We rehearse these issues here.

- Information sharing and communications across disciplines. The paradigms and the terminology of different disciplines are different. And perhaps specialization will remain so necessary that these differences will increase, not decrease. Automation support for scientific data is quite new. It has grown up independently in the various disciplines so that accidental problems abound. For example, networks are not readily connected, incompatible database systems are used with no format interchange capability between disciplines, and no automated directory system supports inquiries about the mere existence of data. None of these problems are fundamental; all have technology solutions but they have not been found and applied.

- Data Exchange Formats. Transfer of data across disciplines is complicated by the differences in formats used by the disciplines. Assuming that it is inappropriate to work towards a single format, it will be necessary to have exchange formats so that data can not only be transferred between laboratories in one discipline, but between laboratories in different disciplines.

- Data - Errors, Calibration, Quality. Discipline specific standards for the processing, archiving and presentation of data are just now being proposed and approved by science discipline organizations. As a result the quality of data is directly controlled by the scientists responsible for that data. Scientists differ in the care they take to perform and record calibration and scrubbing metadata.

- Timely Analysis. Unless locally dictated, there is no data policy ensuring timely analysis of new data. This has even led to the invention of a term ‘‘pre-discovery’’ in the astronomy community to describe plates which record new events, such as the supernova detected in 1988, which were not studied until after another scientist announced the event.

- Complete, Careful Retention of quality Metadata. More and more frequently data is being scrutinized by a scientist who is different from the scientist who collected that data with a particular intent in mind. The new viewer of the data requires accurate and precise information of the form of the data as well as how it was collected and calibrated. In some cases it would be ideal to have access to all notes that the collecting scientist may have recorded in a laboratory notebook. First, the existing database systems are not particularly helpful in ensuring that quality metadata is retained. Second, acquisition of precise and accurate metadata is hard work.

- Re-analysis of data. There is increasing need and opportunity to use data for multiple purposes. Success requires re-analysis of the data. And re-analysis requires that support for that re-analysis be considered at the outset. Quality of the data has to be assured. Known errors must be removed and calibration performed. Re-analysis calls for application of tools not anticipated at data collection. Automated support for the processing of scientific data is immature

enough to make this difficult even though there are no fundamental reasons for it to be so.

• Timely distribution of data versus proprietary rights. The scientific disciplines need to adopt policies which mediate the tension between ensuring the timely distribution of data to all comers and the protection of the investment of the scientists who invested the intellectual effort and the resources to acquire the data. Different disciplines may address this issue differently.

• Locating and retrieving data. The question ‘‘What data exist in the world that are related to my problem?’’ cannot be answered today. Directories and catalogs are woefully absent. Where they exist, support for efficient and effective query processing is sometimes lacking.

• Data Retention/Lifetime. It has repeatedly been said that the scientific community is about to be inundated with data. Economic considerations will force decisions about what data to retain and for how long. Intelligent judgments must be made when future data needs are not all predictable.

• Scientists and Data Systems Computer Scientists need to work together from the onset of a project to define system requirements and interfaces. There has been poor communication of the problems and needs of the individual sciences to the computer scientists who could meet those needs, because the computer scientists are not listening hard enough. And in the opposite direction individual collections of discipline scientists sometimes elect to invent ad hoc data system solutions, which after the fact are poorer than those which the computer science specialists could have provided. Neither field is at fault; communication between disciplines with different attitudes, paradigms and skills is just difficult. Currently, there is a question of whether there is a rewarding career path for a database creator in the sciences. However, the objectives are important and resources are limited. Solving this problem case by case is important.

Jurists have a saying that ‘‘hard cases make bad law.’’ In a similar vein, extreme cases can lead to bad policy. It is not completely clear whether the problems illustrated in this case study represent poor science (e.g., failure to analyze available data, failure to calibrate properly, or failure to ask the right questions) or represent poor data management. In many cases, the data was actually acquired and used once the scientists had focused on the right questions. However, the case study did illustrate in very concrete terms that good science is interdependent with good data management — an interdependence that is likely to grow in the 90’s.

APPENDIX A

FITS — A Self-Describing Table Interchange Format†

Donald C. Wells[1]
National Radio Astronomy Observatory[2]

Basic FITS, the ‘‘Flexible Image Transport System’’, is a data format which was designed by astronomers in 1979[3] to support interchange of n-dimensional integer and floating point matrices using a self-describing notation. FITS is the de facto interchange format used by astronomers everywhere since 1980. The rules of the format are controlled by the FITS Working Group of Commission 5 (Astronomical Data) of the International Astronomical Union, and there are North American and European FITS standards Committees as well. FITS is also the official interchange and archive format for NASA astrophysics missions, and NASA operates a FITS Support Office, including a hot-line service.[4] There is an anonymous-guest archive for FITS matters[5] and an E-mail exploder.[6]

The architecture of Basic FITS is extensible; the meta-rules for extensions are also a part of the standard.[7] In particular, extensions to transmit tables have been designed, and the ASCII tables extension is also a part of the FITS standard.[8] Numerous CDROMs containing databases in the FITS ASCII tables format have been published by NASA projects during the past two years. A binary tables extension to FITS has been proposed;[9] prototype implementations have demonstrated interoperability and this extension is currently being considered by the FITS committees for adoption.

The FITS ASCII tables extension is capable of conveying a set of tables as a self-documenting machine-independent and OS-independent bytestream. The logical record size is 2880 bytes;[10] record

---

†The workshop organizers asked Don Wells to write a short tutorial on FITS, a self-describing data interchange format, which has been used effectively in the astronomy community. FITS has been a very successful catalyst for the exchange of data. Two reasons are: (1) FITS is self-describing — the astronomer structures data as suits the project needs; (2) NRAO developed the support software tools and has both maintained those tools and distributed them free to users.

[1]**dwells@nrao.edu**; 804-296-0277; Donald C. Wells, National Radio Astronomy Observatory, Edgemont Road, Charlottesville, VA 22903-2475.

[2]NRAO is operated by Associated Universities, Inc., under agreement with the National Science Foundation.

[3]Wells, D.C. and Greisen, E.W., 1981, Astron. Astrophys. Suppl. Ser. **44**, 363-370, ‘‘FITS: A Flexible Image Transport System’’.

[4]Barry Schlesinger, 301-794-4246, **bschlesinger@ncf.span.nasa.gov**

[5]**fits.cx.nrao.edu**, 192.33.115.8, in directory **/FITS**; this text is **/FITS/doc/fitsdbmsapp.tex**

[6]send requests to be added to the mailing list to **fitsbits-request@fits.cx.nrao.edu**

[7]Grosbol, P., Harten, R.H., Greisen, E.W. and Wells, D.C., 1988, Astron. Astrophys. Suppl. Ser. **73**, 359-364, ‘‘Generalized Extensions and Blocking Factors for FITS’’.

[8]Harten, R.H., Grosbol, P., Greisen, E.W. and Wells, D.C., 1988, Astron. Astrophys. Suppl. Ser. **73**, 365-372, ‘‘The FITS Tables Extension’’.

[9]Cotton, W.D., 1990, ‘‘FITS Binary Tables’’, draft available from D. Wells.

[10]This (peculiar) size is rich in prime factors; it is commensurate with the word and byte sizes of all computers that have ever been sold in the commercial market. In 1979, when FITS was designed, machines with 6-bit bytes and 24-, 36-, and 60-bit word sizes and ones-complement arithmetic were still commonly used by astronomers. Indeed, the first FITS file was written by an IBM 360 (32-bit, twos-complement, EBCDIC codes, PL/I program) and was read by a CDC 6400 (60-bit, ones-complement, 6-bit ‘‘Display’’ codes, Fortran program); that interchange worked on the first try and the file is still readable today by all FITS readers, long after both original environments have become irrelevant to astronomical computing. Obviously a new format design today would use record lengths of 2^n but most astronomers believe that the principle of protecting the older bits is still important.

blocking by integer factors from one to some limit (typically ten) is allowed on media for which it is a relevant concept.[7] The data structures are preceded by ‘‘headers’’, which are 80-character lines in keyword-equals-value format. There are 36 header lines per logical record, and records are padded with blanks. FITS headers and tables extensions do not contain carriage returns or line feeds or other non-printing ASCII codes.

The FITS ASCII tables extensions are appended to the Basic FITS binary matrix. The matrix dimensions are allowed to be zero, but the minimum Basic FITS header must still be present. There are two reasons for this convention: (1) FITS is a *family* of formats which have internal consistency, and this simplifies documentation, shortens learning time and made standards negotiations easier, and (2) in many scientific applications auxiliary tabular data structures need to be associated with the main binary matrix data structures.

In this appendix we will display a typical FITS table, a single table in a FITS file. The table has 2268 rows and 22 columns encoded in 80 ASCII characters. The data were produced by automatic software which searched images of the Northern sky produced from scans made by the 300-foot telescope at Green Bank, WV (the 300-foot collapsed in November 1988, about a year after these data were recorded), and were given to D. Wells by James J. Condon of NRAO for use in this appendix. In the verbatim listings shown below two extra lines have been prefixed to the listing of each logical record to show the column alignments in the file, and the record number and line number are shown for each line (these are not a part of the FITS bytestream, of course). First, we show the minimum Basic FITS header:

```
                1         2         3         4         5         6         7         8
 r/l    12345678901234567890123456789012345678901234567890123456789012345678901234567890

01/01:  SIMPLE  =                    T / Standard FITS format (AA Suppl. 73, 365)
01/02:  BITPIX  =                    8 / 8-bit characters
01/03:  NAXIS   =                    0 / No image data array present
01/04:  EXTEND  =                    T / There may be standard extensions
01/05:  BLOCKED =                    T / Tape may be blocked (2880 byte records)
01/06:  TELESCOP= 'NRAO91M '           / 91m = 300-ft telescope (r.i.p.)
01/07:  INSTRUME= '7BEAM6CM'           / 7-beam receiver
01/08:  OBJECT  = '87GB CAT'           / The 87GB 4.85 GHz source catalog
01/09:  EPOCH   =               1950.0 / Equinox (yr) of RA, dec values in table
01/10:  DATE-OBS= '01/10/87'           / Observation start date (dd/mm/yy)
01/11:  OBSERVER= 'CBS     '           / Condon, Broderick, and Seielstad
01/12:  ORIGIN  = 'NRAOCV  '           / Written at NRAO, Charlottesville
01/13:  DATE    = '11/06/90'           / Date file written (dd/mm/yy)
01/14:  HISTORY  AIPS IMNAME = 'B1950.11H'
01/15:
01/16:  COMMENT  This table contains all sources from the 87GB catalog
01/17:  COMMENT  with hours of right ascension = 11 (equinox B1950)
01/18:  COMMENT  derived from the Green Bank 4.85 GHz sky survey made in 1987
01/19:  COMMENT  with the 91-m telescope  (J. J. Condon, J. J. Broderick, and
01/20:  COMMENT  G. A. Seielstad 1989, A. J. 97, 1064),
01/21:  COMMENT  in standard FITS table format (see Astr. Ap. Suppl. 73, 365).
01/22:  COMMENT  Catalog reference:  P. C. Gregory and J. J. Condon,
01/23:  COMMENT  Ap. J. Suppl., submitted May 1990.
01/24:  END
01/25:
01/26:
01/35:
01/36:
```

The **NAXIS** line declares that the binary matrix does not exist (it’s dimensionality, the number of axes, is zero); the type of the matrix elements (**BITPIX=8**) does not matter in this case. The following keyword-value pairs of this header are optional (except the **END**). The next logical record begins the header of the ASCII table:

```
               1         2         3         4         5         6         7         8
 r/l    12345678901234567890123456789012345678901234567890123456789012345678901234567890

02/01:  XTENSION= 'TABLE   '           / Table extension
02/02:  BITPIX  =                    8 / 8-bit characters
02/03:  NAXIS   =                    2 / Simple 2-D matrix
02/04:  NAXIS1  =                   80 / Number of characters per row
02/05:  NAXIS2  =                 2268 / Number of rows = number of sources
02/06:  PCOUNT  =                    0 / No random parameters
02/07:  GCOUNT  =                    1 / Only one group
02/08:  TFIELDS =                   22 / Number of fields per row
02/09:  EXTNAME = 'B1950.11H'          / Name (Epoch.hours of right ascension)
02/10:  EXTVER  =                    1 / Version number
02/11:  EXTLEVEL=                    1 / Hierarchical level
02/12:
02/13:  TTYPE1  = 'RAH     '           / right ascension (hours)
02/14:  TBCOL1  =                    1 / start in column 1
02/15:  TFORM1  = 'I2      '           / 2-digit integer
02/16:  TUNIT1  = 'HR      '           / units are hours
02/17:  TNULL1  = '99      '
02/18:
02/19:  TTYPE2  = 'RAM     '           / right ascension (minutes)
02/20:  TBCOL2  =                    3 / start in column 3
02/21:  TFORM2  = 'I2      '           / 2-digit integer
02/22:  TUNIT2  = 'MIN     '           / minutes of time
02/23:  TNULL2  = '99      '
02/24:
02/25:  TTYPE3  = 'RAS     '           / right ascension (seconds)
02/26:  TBCOL3  =                    5 / start in column 5
02/27:  TFORM3  = 'E4.1    '           / xx.x SP floating point
02/28:  TUNIT3  = 'S       '           / seconds of time
02/29:  TNULL3  = '99.9    '
02/30:
02/31:  TTYPE4  = 'URAS    '           / rms uncertainty in RAS (seconds)
02/32:  TBCOL4  =                   10 / start in column 10
02/33:  TFORM4  = 'E3.1    '           / x.x SP floating point
02/34:  TUNIT4  = 'S       '           / seconds of time
02/35:
02/36:  TTYPE5  = 'DECDSIGN'           / declination sign
```

This is an extension of type ‘‘**TABLE**’’; it is a 2-dimensional matrix of 8-bit bytes, with 80 bytes per row and 2268 rows in the matrix. The keyword **TFIELDS** on line 8 tells us that there are 22 columns in the table. Keyword **EXTNAME** specifies a name for this extension (multiple extension structures can be concatenated within a single FITS file, and can be distinguished by their names). Each of the table columns is documented by a set of five keywords. **TTYPE***ii* specifies the column label for the *ii*-th column. **TBCOL***ii* specifies the ordinal in the matrix of the first character of the data field of the column, and **TFORM***ii* specifies the format (and the field width) in Fortran style. **TUNIT***ii* specifies the physical units of the column and **TNULL***ii* specifies the field value that signifies nulls. Here is the last header record:

```
                1         2         3         4         5         6         7         8
 r/l    12345678901234567890123456789012345678901234567890123456789012345678901234567890

05/01:
05/02:  TTYPE20 = 'ZERO    '           / zero-level of fit (Jy)
05/03:  TBCOL20 =                   67 / start in column 67
05/04:  TFORM20 = 'E3.3    '           / (.)xxx SP floating point
05/05:  TUNIT20 = 'JY      '           / Jansky
05/06:
05/07:  TTYPE21 = 'PIXX    '           / x-coordinate pixel number
05/08:  TBCOL21 =                   71 / start in column 71
05/09:  TFORM21 = 'I4      '           / 4-digit integer
05/10:
05/11:  TTYPE22 = 'PIXY    '           / y-coordinate pixel number
05/12:  TBCOL22 =                   76 / start in column 76
05/13:  TFORM22 = 'I4      '           / 4-digit integer
05/14:
05/15:  AUTHOR  = 'P. C. Gregory and J. J. Condon'
05/16:  REFERENC= 'Ap. J. Suppl., submitted 1990 May'
05/17:  DATE    = '11/06/90'           / file generation data (dd/mm/yy)
05/18:
05/19:  END
05/20:
05/35:
05/36:
```

The next FITS logical record begins the table itself, which extends for 63 (=2268/36) FITS records. Here are the first and last rows of the table:

```
                1         2         3         4         5         6         7         8
 r/l    12345678901234567890123456789012345678901234567890123456789012345678901234567890

06/01:  11 0 1.5 1.0 +181916 19  63.2 226.9    67   10      1.13 0.96 -64  -3  513  362
06/02:  11 0 3.2 1.5 +322917 23  65.7 194.2    35    7      1.38 0.68  55  -1  512  735
06/03:  11 0 5.6 1.6 +27 6 9 26  65.6 207.3    30    7      1.58 0.75  44   0  511  253
06/04:  11 0 9.7 2.0 +451913 25  61.8 166.3    27    6      1.28 0.69 -55  -2  510   93
06/05:  11 011.0 1.1 +105913 20  59.4 240.1    67   11      1.15 0.67  -4  -6  509  601
06/06:  11 011.7 0.9 + 515 9 18  55.8 248.3   135   20      1.22 0.66 -10   1  509   86

68/04:  115859.9 2.1 + 917 1 44  68.3 267.0    40    9   W  1.94 1.32 -32  -4   92  450
68/05:  1159 5.1 1.5 +472035 19  67.7 146.2    50    8      0.87 0.69 -80   1  222  283
68/06:  1159 6.2 1.6 +231311 27  77.8 230.2    35    7      1.20 0.73 -71  -1  119  804
68/07:  1159 6.9 0.9 +113140 19  70.1 263.6   146   21      0.92 0.74 -51  -2   92  651
68/08:  115911.4 0.9 +144814 19  72.7 257.3   211   29      1.01 0.81 -59  -1   96  942
68/09:  115914.0 0.9 +105335 19  69.6 264.7   153   22      1.12 0.68   5   0   89  594
68/10:  115915.7 2.0 +502524 24  65.0 142.4    32    7      1.66 0.98 -65  -6  237  559
68/11:  115916.7 1.0 +393541 16  73.9 160.6   261   32      1.15 0.85   8   1  179  484
68/12:  115919.2 1.1 +3313 5 19  77.6 181.9    73   11      1.03 0.72  -8   1  150  806
68/13:  115920.1 1.2 +445634 17  69.8 149.6    90   11      0.87 0.71   7   0  205  960
68/14:  115920.7 1.1 +3651 2 18  75.7 168.4    86   12      1.24 0.96  88  -3  165  238
68/15:  115929.5 1.4 +3130 0 23  78.3 189.5    43    8      0.99 0.67 -84   0  140  653
68/16:  115930.2 1.2 +581838 14  57.9 135.5   268   27      1.09 0.81  -4   0  283  369
68/17:  115933.9 1.9 +504917 22  64.7 141.8    36    6      0.94 0.79 -13  -2  235  595
68/18:  115934.4 1.0 +13 228 20  71.4 261.1   107   16      1.15 0.84 -25   1   85  786
68/19:  115935.0 2.3 +595826 22  56.3 134.4    35    6      1.77 0.99  89  -5  293  518
68/20:  115936.1 1.2 +45 122 17  69.7 149.4    78   10      0.96 0.81 -78  -5  202   75
68/21:  115937.6 1.6 +342824 25  77.1 176.7    36    7      1.34 0.70 -43   0  149  918
68/22:  115939.8 1.0 +214047 19  77.2 236.8   116   16      1.04 0.72  25  -1  102  667
68/23:  115941.5 2.5 +552123 26  60.6 137.6    27    6      1.25 0.83 -48  -5  262  104
68/24:  115941.8 2.2 +622451 19  54.0 132.9    43    6      1.12 0.78  48  -2  308  736
68/25:  115942.0 1.4 +182235 25  75.3 248.3    44    8      1.33 0.94 -85  -3   93  372
68/26:  115942.6 1.8 +721656 11  44.6 128.3   140   12      1.07 0.79 -20  -1  378  722
68/27:  115946.2 1.4 + 85619 26  68.1 267.9    48   10      1.19 0.61 -22   0   74  419
68/28:  115946.4 1.2 + 83314 24  67.7 268.4    61   11      0.95 0.61 -41   1   74  384
68/29:  115951.3 2.4 +272523 41  78.9 210.0    32    8   W  2.04 1.31  37  -8  117  288
68/30:  115951.6 2.3 +534640 25  62.1 138.9    29    6      1.05 0.76 -41  -1  249  858
68/31:  115952.1 1.4 +141830 26  72.5 258.8    43    9      1.35 0.66 -70  -2   81  898
68/32:  115952.1 1.3 +133037 25  71.8 260.4    48    9   W  1.05 0.56   4  -3   79  827
68/33:  115954.1 2.2 +514810 26  63.8 140.7    28    6      1.40 0.87  18   0  236  682
68/34:  115956.7 1.9 +482732 25  66.8 144.4    42    7  E   1.49 1.14  32  -5  216  384
68/35:  115957.3 2.2 +142322 57  72.5 258.6    37    8   W  2.83 0.80  29  -5   79  905
68/36:  115957.7 1.9 +585831 19  57.3 134.9    44    7      1.35 0.92 -40   1  282  429
```

This FITS bytestream consists of 68 logical records: 1 Basic FITS header record, 4 header records for the table header and 63 records for the table itself. The fact that the last row of the table *exactly* fills

the 36^{th} line of the 63^{rd} record is accidental; normally the last record is padded with blanks. Also, the fact that the rows are 80 characters long, commensurate with 2880, is peculiar to this table; other tables might have other row lengths. If the row length is not commensurate the rows of the FITS matrix are written as a contiguous stream without regard to logical record boundaries. The total stream is $195840\ (=68\times2880)$ bytes long.

This file is number 11 of 24, covering the eleventh hour of Right Ascension, the celestial longitude coordinate, and the total survey contains about 50000 sources. Other analogous radio surveys at different frequencies can be compared and composite tables containing source strengths or non-detections as a function of frequency can be constructed. Similar surveys in other frequency ranges (X-ray, ultraviolet, optical, infrared) can also be compared with this source list, and valuable astrophysical insight comes from such ‘‘panchromatic’’ astronomy.

APPENDIX B

Position Papers

Table of Contents

1. Introduction 1
2. Panel 1: Multidisciplinary Interfaces 2
3. Panel 2: Emerging and New Technologies 13
4. Panel 3: Core Tools 22
5. Panel 4: Ozone Hole Case Study 27
Appendix A: FITS — A Self-Describing Table Interchange Format 32
Appendix B: Position Papers 37

On the Relationship of Extensible DBMSs to Scientific Databases†

Don Batory
Department of Computer Sciences
The University of Texas

DBMSs are complex software systems that are notoriously difficult to build. Extensible database systems were conceived to ease the burden of DBMS construction. A number of different approaches to extensibility have been proposed and realized within the last year [Car88, Haa89, Sto86]. Among them, the Genesis approach is distinguished as a software building-blocks technology [Bat88a-b, Bat89]. Its premise is that complex software systems can be constructed from prefabricated components in minutes at virtually no cost.

Genesis 2.0 became operational in November 1989. Our objectives were achieved: customized relational DBMSs in excess of 50K lines of C could be specified and their executables produced within a half hour. The process of specification relies on a graphical layout editor which enables target systems to be defined as compositions of available software components. A top-down design methodology is embodied in the editor, where implementation details, ranging from the selection of nonprocedural data language(s) to the packaging of records in physical blocks, are captured by component assemblies. File structures, storage systems, network DBMSs, and relational DBMSs can be expressed by the editor.

Scientific databases (SDBs) pose a unique opportunity for evaluating extensible DBMSs. Extensible DBMSs are new and relatively untested. They provide features and capabilities that their designers felt would be critical for tailoring DBMSs to new applications. Are these features sufficient for SDBs? If not, what is lacking? Research in SDBs can answer important questions that could significantly influence the course of future extensible DBMS research.

At the same time, extensible DBMSs provide a unique opportunity for helping SDB researchers. Given the algorithms, data languages, etc. that are believed necessary for storing and processing SDBs effectively, extensible DBMS technologies should be able to produce database systems customized for SDB applications. Experimentation with such systems should provide invaluable feedback on proposed algorithms, etc., ultimately reducing the time in which SDB research impacts SDB end-users.

The research communities for scientific databases and extensible DBMSs have the opportunity of advancing their respective domains in important ways through cooperation.

References

[Bat88] D.S. Batory, 'Concepts for a Database System Synthesizer', **ACM PODS 1988**.

[Bat89a] D.S. Batory, J.R. Barnett, J. Roy, B.C. Twichell, and J. Garza, 'Construction of File Management Systems From Software Components', **COMPSAC 1989**.

[Bat88b] D.S. Batory, 'On the Reusability of Query Optimization Algorithms', **Information Systems**, 1989.

[Car88] M. Carey, et al., 'A Data Model and Query Language for EXODUS', **ACM SIGMOD**, 1988.

[Haa89] L.M. Haas, J.C. Freytag, G.M. Lohman, and H. Pirahesh, 'Extensible Query Processing in Starburst', **ACM SIGMOD**, 1989.

[Sto86] M. Stonebraker and L. Rowe, 'The Design of POSTGRES', **ACM SIGMOD**, 1986.

† This work was supported by the National Science Foundation under grant DCR-86-00738.

Database Needs for Global Change

Francis Bretherton
University of Wisconsin

This papers starts from the needs of Earth System Science, and emerging perspective on the functioning of our global environment. It requires understanding the workings of an interconnections between the atmosphere, ocean, cryosphere, solid earth and biosphere, with particular attention to the changes over decades to centuries due to human activities and natural variability. As the Global Change Research Program, it is a major U.S. initiative, linked to similar efforts in other countries around the world. It is multi-agency and multi-disciplinary, requiring global measurements and their synthesis over many decades. The volume of satellite data is expected to exceed 1 TByte/day by the end of this decade. Associated in situ data will be very complex and diverse, ranging from automated instrument systems measuring temperature in the ocean to descriptions of the annual cycle of crop planting and harvest in Africa. The management of this information will be crucial, and a substantial fraction of available resources are being earmarked for this task.

Types of scientific activity driving the information management activity include:

Preparation of global measurement products
Syntheses using numerical models
Process studies
Data archeology
Assessments

The information system will be distributed, multi-disciplinary, and must be able to evolve with time. Control will largely be decentralized, in the hands of many different scientific groups cooperating with national data centers according to acceptable information integrity and exchange standards. Data systems will have to cope not just with raw data, but with metadata, and with added products and assessment information. Derived products are likely to be the most commonly retrieved information, but an archive adequate for re-derivation of those products is also required.

Specific issues for database systems include:

(i) The relational model is probably inadequate. Proximity in space and time is an essential attribute for geophysical data, that must be efficiently recorded and handled in queries. Also graphical and loosely structured text such as log books and quality assessments must remain effectively associated with the data sets to which they refer. An object oriented approach that includes access methods with the data is needed.

(ii) Methods for browsing and classifying high volume image data need to be developed. Random access to raw archives is prohibitively expensive.

(iii) For quasi-autonomous, distributed databases to work together, a common information exchange language (SQL++ ?) is essential. Prototypes for such a language (or languages) need to be identified as soon as possible.

(iv) Given the open-ended multi-disciplinary nature of this enterprise, no standardized lexicon or data-dictionary is likely to be generally accepted by all parties. Instead, we will need an evolving system of local aliases relatable to a centralized corporate name that uniquely identifies each distinct construct. How can this lexicon itself be managed?

Extensible Database Systems and Scientific Data

Michael J. Carey
Computer Sciences Department
University of Wisconsin

Background

Until the past few years, research and development efforts in the database system area were focused primarily on the support of traditional business applications. The design of database systems capable of providing effective support for less traditional data, such as scientific data, geographic data, image data, and computer-aided design and manufacturing (CAD/CAM) data, is now widely recognized as an important and pressing problem in the database community. Significant challenges await next-generation database systems, as the needs of emerging database applications can differ quite widely from one another. For example, some applications involve managing complex objects with many small subobjects; some depend heavily on support for data with rich spatial relationships, requiring spatial operations; many require support for temporal data, but various notions of time exist; some involve handling very large objects such as images or large arrays. Even the "restricted" domain of scientific data promises to be very challenging, given the rich range of scientific data of interest nowadays; examples include sequences of Earth images produced by weather satellites, genetic sequence data, measurements from high energy physics experiments, NMR data, and descriptive data about scientific instruments and the structure of complex scientific experiments.

Potential Solutions

Despite the apparently wide range of database requirements posed by emerging applications such as scientific data management, certain general themes are clear: the need for new data types (simple and/or structured) and their associated operations, new storage and index structures to support their storage and manipulation, and optimization support to ensure that queries involving such new data types can be processed efficiently. In response to these requirements, a number of researchers have undertaken the design and implementation of so-called *extensible database systems*. Prominent examples of ongoing extensible database system projects include POSTGRES at the University of California-Berkeley, Starburst at the IBM Almaden Research Center, EXODUS at the University of Wisconsin, and GENESIS at the University of Texas [1]. These projects can be loosely classified as being either "complete DBMS" projects (POSTGRES and Starburst) or "DBMS toolkit" projects (EXODUS and GENESIS). The complete DBMS projects aim to provide a full DBMS, based on a single data model (an extended relational model), to satisfy all new applications. In contrast, the goal of the DBMS toolkit projects is to provide tools that will enable new DBMSs to be easily constructed to handle new and different applications as they arise.

The EXODUS project at the University of Wisconsin [2] is an extensible database system project that falls into the DBMS toolkit class. The project's goal is to reduce the effort needed to develop a new, application-specific database system such as a scientific data manager. EXODUS provides certain kernel facilities, including a versatile storage manager. In addition, it provides an architectural framework for developing new database systems; powerful tools to automate certain aspects of the development process, including a rule-based query optimizer generator and a new programming language; and libraries of generic software components (e.g., access methods) that should be useful for many application domains. We recently designed a data model and query language that we view as a solid foundation for many next-generation database applications, and we are now putting the EXODUS toolkit to the test by using it to build a DBMS based on this design. It is our hope that the EXODUS toolkit will prove to be useful for addressing the challenges posed by the many scientific data management problems that appear to be emerging.

My Workshop Interests

I am associated with the EXODUS project, but I also recently spent three months working on the Starburst project [3] as a visiting scientist at IBM. In addition, I have a strong interest in object-oriented database research, a related direction in database technology that offers solutions to problems posed by certain classes of emerging database applications. As a result, I am hoping to get at least two things out of the workshop. First, I am hoping to gain a much better understanding of the current and anticipated data

management problems faced by the scientific research community. Second, I am hoping to gain insight into how the various solutions being developed in the database research community — full extensible DBMSs, extensible DBMS toolkits, and object-oriented DBMSs — may help in addressing scientific data management problems. Of particular interest are the questions of (i) how different the data management requirements are across various scientific disciplines, and (ii) whether or not it appears feasible for a single DBMS to provide an effective solution across many disciplines.

References

[1] *Database Engineering*, Vol. 10, No. 2, Special Issue on Extensible Database Systems, M. Carey, ed., June 1987.

[2] Carey, M., DeWitt, D., et al, "The EXODUS Extensible DBMS Project: An Overview," in *Readings in Object-Oriented Databases*, S. Zdonik and D. Maier, eds., Morgan-Kaufman Publishing Company, 1989.

[3] Haas, L., et al, "Starburst Mid-Flight: As the Dust Clears," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 2, No. 1, March 1990 (to appear).

Are Recent Advances in Database Management Technology Relevant to Scientific Databases?

Umeshwar Dayal
Digital Equipment Corporation
Cambridge Research Laboratory

For applications that require access to large, shared repositories of data, database management systems (DBMSs) offer many advantages over ordinary file systems. DBMSs provide data independence by separating the conceptual model of data from the physical storage structures or access methods; this permits modification of the storage organization and access methods to improve performance, without requiring that the applications be rewritten. DBMSs allow different applications or users to have different views of the same data. Modern DBMSs provide a high level query language in which users can express what data is to be retrieved, without specifying in detail how to retrieve the data; the DBMS optimizes the execution of the request. DBMSs control the integrity of the database and protect it from unauthorized access. DBMSs provide backup and crash recovery facilities, and mechanisms for synchronizing concurrent accesses to shared data.

Traditionally, DBMSs have been aimed at business data processing applications. Their data models and type systems are limited to modelling files (or relations), which are collections of structured records (tuples) composed of fairly short alphanumeric fields (attributes). Their query languages are limited to expressing first order queries, plus some arithmetic and aggregation operators, over these collections of structured records. Access methods typically are limited to sequential files, B-trees, and hashing. Only simple integrity constraints such as unique keys, value-in-range, and referential integrity are supported. The atomic unit of work for purposes of integrity control, concurrency control, and recovery is the transaction, which is assumed to be of short duration.

Recent advances in database management technology have been aimed at extending the benefits of DBMSs to other application domains. These advances include enhanced data models, query languages and user interfaces, access methods, integrity control, and transaction management. Much of this work has been done in the context of extensible, object-oriented DBMSs (OODBMSs) and active DBMSs (ADBMSs), and has been strongly influenced by the requirements of CAD/CAM, computer-aided software engineering, and geographic information systems.[1]

My thesis is that many of these advanced capabilities will also be useful for managing scientific databases. Scientific databases typically consist of vast quantities of unstructured ‘‘raw’’ data (e.g., observations, measurements, photographs, images), stored in many files (often on removable media); and structured data (e.g., the results of analysis or processing performed on the raw data; metadata that describes the configuration or environment under which the raw data was obtained, instrument characteristics, etc; and directory or catalogue data to help locate the files in which the raw data is stored). OODBMSs can bring all of these different types of data under centralized control. Object models can directly model conceptual entities such as experiment runs, observations, instruments, and observers; and the complex relationships among these entities. For example, the model can be used to associate the structured data resulting from analysis with the experimental data on which the analysis was done; or to link together data from different runs of an experiment. Object models permit the modelling of highly structured ‘‘complex’’ objects that are composed of other objects (e.g., an experiment may consist of several steps, each of which is itself an object with associated information that must be recorded); and unstructured or ‘‘one of a kind’’ objects (e.g., images, text, speech, annotations). They provide a wide range of collection types such as arrays, vectors, and sequences, in addition to the conventional sets, tuples, and relations. Their generalization and inheritance mechanisms are useful for organizing large databases with many different classes and instances of objects. OODBMSs are extensible: new abstract data types, together with their representations, access methods, and operators, can be added. This makes it possible, for example, to define an image or document data type, and to include operators of these (and other) types in queries. Extensible query optimization techniques are being developed for optimizing queries that include user-defined operators. The use of a high level query language that provides operators for searching collections of objects and detailed

[1] There is also a small, but significant, body of work that has more directly targeted statistical and scientific applications.

manipulations of individual objects facilitates exploration. Ideally, the query language is seamlessly integrated with a programming language, so that complex computations can be invoked within queries. Also, ideally, a graphical user interface is provided so that the scientist does not have to learn an unfamiliar query language or programming language.

In addition, there has been work on modelling, storing, and accessing dimensional (e.g., spatial and temporal) data, which clearly is very important for scientific databases. The spatial and temporal operators can be combined with operators for searching over collections of objects to formulate queries such as ''Find observation data for the past week for all objects located within a distance of x units from object y.''

The development of active DBMSs was motivated by the need to provide timely, and customizable, response to critical events or situations. ADBMSs support the event- or data-triggered invocation of actions, in addition to the conventional, imperative, procedure call mechanism. This capability has many uses. An example is flexible integrity control. Conventional DBMSs evaluate integrity constraints immediately on updating the database, or at the end of a transaction; in an ADBMS, constraint evaluation can be triggered when certain specified events or situations are detected in the database. Instead of simple predicates that are evaluated against the database, arbitrary integrity procedures can be invoked. This feature could be useful in a scientific database where integrity of the data might depend on the experimental configuration, instrument calibration, environmental factors, etc. Also, when an integrity violation is detected, rather than roll back the transaction in its entirety, some less drastic action might be specified (e.g., notify an operator) or a procedure might be called to ''patch'' the database (e.g., adjust the data to correct for errors in instrument calibration). The active capabilities can also be used to alert scientists when certain patterns of events or data are detected. They can be used to propagate operations from one object to other, related, objects (e.g., to propagate updates from one view to another). They can be used to maintain snapshots of the database on which the scientist may try out various ''what if'' scenarios or hypotheses; the ADBMS can refresh the snapshot as desired (immediately as the underlying database is updated, periodically, after some critical events, or on demand). ADBMSs can be used to support collaboration in a group of scientists by allowing them to share data, and by notifying collaborators when significant events are detected by some members of the group. They can be used as the basis for workflow control; i.e, controlling long-running activities (longer than a single transaction) consisting of several steps that must be correctly sequenced (e.g., the steps of an experiment, a simulation, or an analysis procedure). Finally, there is ongoing work on time-constrained active DBMSs, which guarantee real-time response.

At the workshop, I would like to assess the relevance of these emerging technologies to scientific databases. There are no commercially available DBMSs today that can fully meet the performance needs of scientific database applications. However, the metadata, analysis data, directories and catalogues could be brought on-line; also, there are clear advantages to bringing the raw data files under the control of a DBMS. However, many challenging problems remain in dealing with the size, complexity, and rate of growth of scientific databases. I would like to understand these problems, and to identify directions in which DBMS technology should evolve to solve them: for instance, the development of more sophisticated models, query languages, or user interfaces to facilitate ease of use by scientists; the use of optical devices to meet the on-line storage needs; and the exploitation of parallelism and large main memory caches to reduce computation and retrieval costs.

Ensuring Quality Data by Fostering Teamwork Between Researchers and Data Specialists

Vernon E. Derr
National Oceanic and Atmospheric Administration

As the nation's scientists become more concerned with interdisciplinary data extending around the globe in climate and global change studies, the necessity for extensive data sets useful for correlation studies and other analysis increases steadily. A new breed of scientist is appearing. They often get their principal data from data archives rather than by their own efforts in the field or laboratory. Their demands on data archivists for documentation attesting to quality, accuracy, precision, resolution and completeness cause a strain on services from data centers.

Recognizing that the quality of data is an evolutionary process, that requires updating and reassessment, is vital to improving the health of data sets and providing researchers with trustworthy information. An example of reassessment in satellite data is the development of new algorithms for calculating physical parameters from basic irradiances that would perhaps remove some of the noise and provide greater accuracy. Often such improvements are discovered by the scientists analyzing the data.

The extensive interdisciplinary data sets needed must be comparable in coverage and accuracy and resolution to be useful in analysis. They may come from many data sources, perhaps even foreign sources, and with varying quality. The researchers who need such data sets may be hard pressed to assemble them. On the other hand, the data clerks in large centers may not normally have the scientific knowledge to provide the data as required by the researcher.

This impasse occurs often enough to require a solution. One solution might be to require the scientist to delve deeply into the details of data archives. (It seems impractical to hope that data clerks would be able to become scientifically sophisticated.) An optimum way to solve the problem would be to engage qualified scientists in the data centers to work directly with the researchers in developing data sets, directing the activities of less knowledgeable (and less expensive) aides. A number of data centers have begun to move in that direction by hiring well-qualified scientists. Why would a well-qualified scientist want to work in a data center? For the same reasons a safe-cracker would like a job in a safe manufacturing company — the information available is invaluable.

The teamwork that is developing between computationally-oriented researcher and data specialist scientists is especially rewarding when they join in writing papers and reports. The incentives for each as joint authors is stimulating to a high degree and results in excellent science. That is, of course, the goal.

Scientific Database Workshop Position Paper

Glen Flierl
Massachusetts Institute of Technology

The global scale observational programs in oceanography and in the area of "Global Change" will gather volumes of biological, chemical, and physical data far beyond our present holdings. The data sets will be highly diverse, not only in that many different processes will be observed, but also in that many different techniques will be used by numerous groups all over the world. Microcomputers and workstations have dramatically altered the gathering and analysis of oceanic data. While the convenience and ease of use of these machines make them ideal for the individual working on his or her data, the process of exchanging data or collecting relevant information from archived data sets has become even more difficult. Everyone uses different formats with different procedures for manipulating data; while many commercial products permit importing/exporting data from other systems, few of them deal with data stored remotely in large archives. In oceanography, the agencies archiving data are accustomed to supplying it on magnetic tape media in their own specialized, limited, and not always convenient format.

Commercial data bases (at least those with which I am familiar) are not entirely appropriate for scientific data for a number of reasons:

(1) The relational model is often not adequate. Scientific data often represents sampling of a continuous process; ordering therefore is meaningful. Interpolation, smoothing, curve-fitting, Fourier transforms, et cet. all rely on the sequence of data. Data is often grouped and such groupings again have meaning beyond coincidence of the values of one variable. Efficiency in storage and processing requires recognition of the grouping; many operations likewise need to handle grouping properly. Data is frequently multi-dimensional, for example, a variable being known on a three- dimensional grid.

(2) In the geophysical/chemical/biological sciences, data can often not be precisely interpreted without much other information: how was the data gathered, what instrument was used, what processing was done? For some types of data, such as trace metal distributions, even more detail is required. This "metadata" is almost invariably separated from the data when it is entered into archives; this can make the data virtually useless. Systems need to carry metadata as well as information about units and quality flags throughout the processing and make them readily accessible to the user.

(3) Many data base systems are greatly concerned with the update process but provide relatively few data manipulation tools. Most scientific work involves a relatively stable data base (or one with a single point where data is entered) but extensive manipulations. The ability to share tools is as important as the ability to share data.

(4) Commercial systems are usually "closed." The available operations are fixed by the designer. Some permit combining these operations into more complex operations, but this may involve a substantial speed penalty. A system with hooks permitting new operations to be added at any time, along with sufficient support so that a reasonable programmer can construct a program for a new operation.

(5) Access from programs to the data base is important. While this is possible on mainframes, it is not always inherent in micro systems.

(6) Scientific data users need to be able to access data stored in many forms (not known a priori) on many different machines. While there are perfectly valid reasons for using different data storage and access methods to optimize local processing, synthesis of various data sets often falters because of the work involved in data conversion. Packaging a standard converter with the data can alleviate these problems (an object-oriented approach).

(7) Scientists work in different modes at different times and these modes require different capabilities of a database. In exploration or synthesis, flexibility in gathering and manipulating data is paramount. In other cases, such as reducing satellite imagery, efficiency is essential. Whereas scientists may be willing in the latter case to spend considerable effort in designing a database schema, they will not in the former case.

(8) Visualization is an important part of scientific data analysis. Some databases have quite sophisticated graphics packages; however, these are rarely able to cope with the kind of output required for meteorological data (for example). Again, because these systems are closed, it is virtually impossible

to add a new kind of plot.

I suggest, then, that a scientific database for a large, multi-investigator project should combine

(1) A definition of data objects and catalog objects: what inputs and outputs are expected? The substantial impact of such protocols on the way a community thinks is illustrated by the X Windows experience.

(2) Construction of programs which handle these interchanges and interface with data bases and collections of data in use.

(3) Packages of "filters" or functions which transform data and which can be added to at any time. The common relational operations (extended to groups, lists, multidimensional arrays,...) form the basic set.

(4) Modular display (both tabular and plotting) facilities, which again are extensible by the user.

(5) A macro/shell script facility for building more complex operations from existing functions.

While it seems possible to carry out these various steps, the process requires a special openness--availability of tools from many different groups, access to source codes, etc.-- in all aspects. Such openness is part of the scientific endeavor (but not common to commercial vendors) and is the only way to encourage users to develop new functions and display modules. Such user participation is important in ensuring that scientists are not impeded by the data system and can easily find, adapt, or develop tools to do the jobs they need done.

Database Statistics

Nancy Flournoy
American University

Statistics are procedures for summarizing information, drawing inference, and assessing error probabilities and loss functions associated with the summaries and inferences. Statistical methods are designed to accommodate specific data types (e.g., discrete or continuous; qualitative or quantitative; independent or dependent). Each statistical method takes into account the protocol by which data are generated (i.e., the sampling scheme) and depends upon a specific objective (e.g., test a hypothesis or estimate parameters) which may include a utility function as well.

Database structures are new ''data types'' to statistics and a theory of database statistics is needed to support decision-making based upon them. We might begin with the relational model, but there are scientific environments better suited to the hierarchical model, the pedigree model, or networked models in which common keys are absent. Thus for each type of database structure, a database statistic is defined to be a set of mathematical and/or graphical procedures that operate on database structures incorporating data accrual protocols to permit statements of confidence regarding an objective.

Many issues arise when considering various sampling schemes and their impact on statistical assessments. Classical statistical assumptions, such as that data are independent, identically distributed (e.g., as normal random variables) are laughably simplistic. However, at the forefront of work in statistical theory these assumptions are being relaxed and sampling protocols are being generalized and analyzed mathematically.

For example, data accrues sequentially, not independently, in databases. Sequential methods comprise a growing statistical subfield, with more recent focus on methods that adapt as information is added. Also, computational power, numerical approximations, and Monte Carlo techniques are being used to make Bayesian theory tractable.

Whereas work on sequential and adaptive methods is yielding results that improve the match between database structures and statistical techniques, database sampling schemes are often much more complex than the generalizations now being developed. For example, databases may contain mixtures of administrative data together with data from more than one experiment performed on the same experimental units (e.g., subjects, organisms, etc.) and from sequential experiments on different units. A retrospective investigation may utilize data in various combinations. A growing collection of statistical methods that deals with the synthesis of data over different experiments is being called meta-analysis. The connections between meta-analysis, sequential statistics, and database structures need to be made explicit and developed in concert with further developments in both structures and statistics.

Statistical methods should be defined in terms of operations that conceptually act directly on the database structures. It is necessary to have analyses that use the actual error structure in the data so that conclusions are based on more accurate appraisals. It also would be a conceptual and an operational simplification to have statistical methods that act directly on data structures. From an operational point of view, such methods would eliminate the intermediate process of coding the creation of a flat file before doing a statistical analysis.

There are many challenges inherent in the development of database statistics. These include the development of statistical protocols for sampling from large databases in conjunction with valid inferential techniques. Database statistics also need a new calculus by which to operate on database structures. Such a calculus would extend the operations of projection, join and selection to higher dimensions and incorporate the concept of matrix multiplication to produce higher order cross-products.

In addition to the development of database statistics, statisticians need to increase current thrusts in the areas of disseminated quality control, data sharing, meta-data, confidentiality and data sharing. These issues require multidisciplinary cooperation and exchange.

An Electronic Publishing / Information Storage and Retrieval Perspective on the Management of Scientific Databases

Edward A. Fox
Dept. of Computer Science
Virginia Polytechnic Inst. & State Univ.

Scope

Scientific databases are at the heart of the investigations of many researchers in a variety of disciplines. "Database" as commonly used by scientists refers to collections that support a particular type of investigation. What is notable about those databases from the computer science perspective, however, is their heterogeneity. One may classify these databases using various contrasts: bibliographic vs. full-text, graphic vs. image, speech vs. sound, simple vs. compound document, still vs. motion video, hypertext vs. hypermedia, analog vs. digital, unstructured vs. normalized, relational vs. deductive, hierarchical vs. network vs. tabular, local vs. distributed, geographic vs. spatial, etc. In addition to these matters of form and structure, there are others relating to level of utility. One of the key distinctions here is the difference between raw data, understandable information, and meaningful or usable knowledge. For the purposes of this workshop, we propose that "scientific databases" be defined to include all of the levels, forms, and structures mentioned above. That means that data, information, and knowledge must be considered, and that databases can include multimedia objects of any useful type.

Prior Experience

To properly give a position, it is important to convey the basis for the statements made. As background, then, a few words are in order regarding previous research, development, and service activities. This covers the broad areas of information storage and retrieval (IS&R) where the goal is to develop methods, build systems, and conduct experimental investigations regarding making information readily usable. Computer algorithms, data and file structures, query languages, knowledge representation schemes, user interfaces, storage methods, networking, and a variety of other areas of computer science are involved in IS&R. We have worked with two systems, SMART, designed for bibliographic collection handling using statistical processing methods, and CODER (COmposite Document Expert/extended/effective Retrieval), designed to support distributed information access to full-text collections through the use of AI methods. These have been applied or are being applied to databases in computer science, medicine, libraries, offices, and naval intelligence. Other work with digital video has focused on tool kits for developers and on demonstrations of new technology.

In the area of electronic publishing, a series of CD-ROMs have been prepared, collecting dozens of databases and making them accessible with a wide variety of search and hypermedia tools so that proper contrast and usage is supported. In connection with ACM (the Association for Computing Machinery), there has been involvement with hypertext publishing of scientific literature, a videotape documentary on interactive digital video, extracting of book publications from review and bibliographic databases, development of online searchable databases, and planning for construction of an electronic library in part through standards-based work in handling electronic submissions.

Problems

Scientific databases should be usable by all those whose work would benefit from prior results, especially when produced as the result of governmental support. Key problems are: locating usable databases, finding useful data in those databases, being able to extract that data in a usable form, and maintaining the quality of the databases. These problems in turn lead to a host of related problems: how to effect proper editorial control, how to develop and properly use standards, how to prepare searchable lists of databases, how to develop network access software, how to handle electronic publishing on optical media along with proper retrieval software, etc.

Technology

The technology involved in this field is rapidly developing. Even the long-time standard medium for interchange, 9 track tapes, is no longer suitable for many purposes. The dramatic revolution in storage,

where the production costs are now roughly one cent per four megabytes (for CD-ROM), make it very clear that the real value is in the content and usability of databases. As CD-ROM or similar drives become standard in workstations and PCs, and as storage technologies improve even further, individuals will have local access to gigabytes of data. This has in part been made possible by electronic publishing methods, whereby text, graphics, and in some cases images are directly prepared and processed with computers.

The other key technology, which has received much more attention by NSF, is networking. With 56KB links moving to T1 and T3 links, and as fiber networks proliferate in universities, business, and eventually reach the home, there is tremendous opportunity for accessing vast amounts (terabytes) of remotely stored data.

Methodology

To solve our needs to manage electronic databases, it would be wise to turn to the work that has gone on since the 1950's in the IS&R field. While there is widespread familiarity with relational database methods and with certain tools relating to hypermedia, there is a great deal of largely ignored work that could easily be applied to current problems in database access if proper initiatives were launched. Bringing together investigators involved in — the classic field of IS&R, in database management systems, in object oriented bases, in hypertext and hypermedia, in knowledge representation, and user interface design — a coherent approach could emerge that should lead to standards, prototype development efforts, and practical tools for the nation's scientists. Wherever possible, such teams should work directly with creators and users of scientific databases.

Plan

NSF should embark on a multi-pronged initiative as early as possible. This requires separate administration from current programs, for several reasons. First, the computer science issues are often viewed as more applied than basic from the vantage point of computer scientists, and so would receive low peer review ratings. Second, funding in the knowledge and database area is at present inadequate to support the quality work that is being proposed. Third, funding by programs in the scientific disciplines that produce the databases is also unlikely since support of work more central to those sciences would inevitably be recommended by peer reviews in those disciplines.

In addition, there should be line items allowed in NSF project proposals to support not only page charges, but also electronic publishing storage and production costs so that results are widely disseminated. These efforts should be publicized, encouraged, and adequately supported.

The work required should deal with all of the problems discussed above. It should handle issues of inventorying, cataloging, publishing, making accessible, constructing management tools, and developing standards (with import/export software that is validated for all common environments) for key scientific databases.

In launching this initiative to support making scientific databases usable, NSF should work with other groups. NASA has recently called for proposals on an endeavor that relates, and NSF might choose to collaborate. Other agencies are supporting standards work and research endeavors in medicine and other areas that could be tied in. Outside government circles, too, there are many professional associations that publish literature or databases — a coordinated effort in this regard would be of great value so that researchers in universities and other laboratories can easily have access to useful data. One particularly important initiative today is that of facilitating access to scientific publications at the workstation, so that the myriad of authors, reviewers, and publishers stay coordinated enough so that the Library of Congress and others need not retrospectively convert publications that are today produced on computer, published, and then converted back to computer form in a way that looses most of the usefulness of the original.

NSF should and can take a bold stance on this matter, and would certainly effect considerable savings as a result of avoiding replication of investigations and wasted efforts of leading scientists writing database management software or hunting around on the very expensive high-speed networks of tomorrow. Proper support, followed up by encouragement of training in use of these new capabilities, will help our nation move forward as a leader in the Information Age.

The Challenge of Scientific Database Management

James C. French
Department of Computer Science
University of Virginia

Interest in scientific database management is heating up. A year ago a front page story in the *Wall Street Journal*[1] chronicled the woes of scientists facing the deluge of new data being made available by increasingly sophisticated instruments. The same article noted that only 10 percent of existing space data is ever analyzed; most of this data, gathered at considerable expense, may never be analyzed. More recently, the director of the national data center at NASA Goddard Space Flight Center has conceded that existing funding is inadequate to handle the impending increase in data.[2] Scientists are becoming increasingly dissatisfied with the situation. National data centers are being referred to as ''data cemeteries.'' Our ability to collect data has far outstripped our ability to manage it effectively.

It is estimated that EOS (Earth Observing System) will create the equivalent of the entire Landsat archive every two weeks; that is, as much data in two weeks as Landsat created in its 17 year lifetime.[3] The situation is no different in the life sciences. According to a 1988 report of the National Research Council,[4]

> The mapping and sequencing effort will generate more data than any other single project in the history of biology. For example, just to record the 3 billion nucleotides [of the] human genome will require nearly 1 million pages of printed text.

It has been estimated that the projected sequencing activity will cause GenBank to grow anywhere from 1000 to 1 million times its current size in the next 10 to 20 years.[5]

A recent *Science* editorial[6] has described the situation as ''a kind of electronic chaos that limits the value of the vast store of information for the average user.'' But, this is not a new refrain. Forty five years ago, as World War II was drawing to a close, Vannevar Bush urged that, in his editor's words, ''men of science should then turn to the massive task of making more accessible our bewildering store of knowledge.''[7] Bush himself wrote that:

> There is a growing mountain of research. But there is increased evidence that we are being bogged down today as specialization extends. The investigator is staggered by the findings and conclusions of thousands of other workers — conclusions which he cannot find time to grasp, much less to remember, as they appear. Yet specialization becomes increasingly necessary for progress, and the effort to bridge between disciplines is correspondingly superficial.

This sentiment was echoed 25 years later in 1970 by Alvin Toffler who wrote in *Future Shock*,[8]

> Today change is so swift and relentless in the techno-societies that yesterday's truths suddenly become today's fictions, and the most highly skilled and intelligent members of society admit difficulty in keeping up with the deluge of new knowledge — even in extremely narrow fields.
>
> Ideas come and go at a frenetic rate. (A rate, that, in science at least, has been estimated to be twenty to one hundred times faster that a mere century ago.)

Clearly, our shortcomings in the management of scientific data have been recognized for many years.

---

[1] *Wall Street Journal*, January 12, 1988.

[2] "Jukeboxes for Scientists," *Scientific American*, **261**,1 (July 1989), pp. 24-25.

[3] News & Comment, *Science*, **244** (June 16, 1989), p. 1250.

[4] *Mapping and Sequencing the Human Genome*, Report of the Committee on Mapping and Sequencing the Human Genome, National Research Council, Washington, D.C., 1988, p. 75.

[5] J. W. Fickett and C. Burks, ''Development of a Database for Nucleotide Sequences,'' in *Mathematical Methods for DNA Sequences*, M. S. Waterman (editor), CRC Press, Boca Raton, Florida, 1989, 1-34.

[6] Retrieval of Scientific and Technical Data, *Science*, **245** (July 7, 1989), p. 9.

[7] V. Bush, ''As We May Think,'' *The Atlantic Monthly*, **176**,1 (July 1945), pp. 101-108.

[8] Alvin Toffler, *Future Shock*, Random House, 1970, p. 157, 177.

Today, we are attempting to address the same problems.

Our inadequate handling of our acquired knowledge has led to one particularly alarming situation. As Bush noted, ''Mendel's concept of the laws of genetics was lost to the world for a generation because his publication did not reach the few who were capable of grasping and extending it.'' More recently, it has been noted that the celebrated hole in the ozone layer should have been detected ten years earlier — evidence of its existence was present in the Nimbus-7 satellite data at that time.

Why does this situation persist? Why have we not solved our problems with scientific data? We have not lacked vision. In 1945, Bush noted that, ''The world has arrived at an age of cheap complex devices of great reliability; and something is bound to come of it.'' He then went on to posit a machine, *memex*, which correctly anticipated the desktop workstation, electronic mail, and hypertext. He also envisioned an environment of harmonious data and knowledge sharing in which practioners were able to make immediate productive use of the work of others. The current availability of ''cheap complex devices of great reliability'' far exceeds anything that Bush imagined. Why then has this ultimate goal of his vision not been realized?

The reasons are many and complex. Some examples include: insufficient funding; inadequate access to technology; and too few individuals trained to analyze data.[9] The root cause, however, is that we have just not had the commitment. Until the management of scientific data is accorded priority on the national research agenda, we will continue to lack the funding, equipment, and people necessary to solve the technical problems. Fortunately, evidence of the last several years suggests that the problems are gaining visibility and the resolution of the problems is beginning to be given due attention.

The scientific community abounds with data that has been collected by individual investigators and has, in some sense, become the property of the investigators. The maintenance of the acquired data is a necessary evil in the conduct of any scientific investigation. Sharing data, however, requires much more effort on the part of the investigator (e.g., standards of quality must be maintained and guaranteed, documentation must be provided, etc.) and, from the point of view of the investigator, may not be worth the effort. It is the availability of data to individual researchers, that is, *sharing of data*, which should be the focal point of this workshop.

There are already two well known and very ambitious projects for widespread data sharing underway: the Global Change program and the Human Genome Project. These are concrete areas in which computer science database researchers can become involved. They are also sufficiently diverse so that they encompass most if not all the extant problems. Both initiatives involve intradisciplinary and, to one degree or another, interdisciplinary sharing of heterogeneous data. While sharing data within a discipline is nontrivial, sharing across disciplines is a much more difficult undertaking. But the heterogeneous distributed database problem facing scientists is much less difficult than, for example, the problems that would be faced by the European Economic Community.

The impression given by many scientists is that the computer science community has been singularly unhelpful in solving the problems of data management. Within the database research community, scientific databases have been seen by many as largely uninteresting and unglamorous. (This attitude is prevalent in the sciences too. After all, few careers are advanced through the stewardship of data.) A relatively small number of researchers work in this area. This workshop can serve as an educational opportunity and help to attract researchers to this area.

Although we can not address all the problems with data management facing the scientific community, we can make some strides toward identifying the technical issues involved and foster a cooperative interdisciplinary atmosphere for solving these problems. Only then will we be able to achieve Bush's goal, that ''The applications of science ... may yet allow [man] truly to encompass the great record and to grow in the wisdom of race experience.'' Our challenge is to make the ''great record'' conveniently available to the scientific community.

---

[9] According to the Association of American Universities, there will be a shortage of some 7500 Ph.D.'s in science and engineering by early in the next century. (*The Ph.D. Shortage: The Federal Role*, AAU, Washington, D.C., January, 1990)

Workshop on Scientific Databases — Position Paper

Hector Garcia-Molina
Department of Computer Science
Princeton University

Two of the key questions to be addressed by the workshop are the following: (1) What current database technology can be used in scientific data processing; and (2) What types of additional tools need to be developed. In a first attempt to answer these questions, let me briefly look at the steps that I think are involved in scientific data processing. (I will probably have a better understanding *after* the workshop; this is my current understanding.)

(a) *Producing*. The first step is obviously producing the data, out of some experiment or sensor.

(b) *Finding.* For many, however, the first step involves finding the data produced by others. The scientist decides he or she needs some particular type of data, and then must figure out who has such data.

(c) *Interpreting*. Once the data is procured, it must be interpreted. This involves figuring out what exactly each datum represents, e.g., what instrument was used to obtain it, how accurate the data is, what the test environment was, and so on.

(d) *Modeling*. Related to the interpretation phase is the modeling phase. This is the traditional database activity of abstracting the entities and relationships that exist. For example, we may determine that a measurement record contains three fields, temperature, pressure, and date. For each sensor object, there corresponds a set of measurement records, and so on.

(e) *Translating.* After the data is interpreted and modeled, it may have to be translated to a different environment for processing. This may involve changing the model and the interpretation.

(f) *Storing.* The data needs to be safely stored, providing such things as protection from failures, concurrency control, and security.

(g) *Processing.* Finally, the data must be processed. This is very application dependent, but there may be some common functions among scientific applications. For example are traditional database query functions, statistical analysis (e.g., sampling, graphing, aggregation, interpolation), and time analysis (e.g., time series analysis, time based queries).

The three steps where current database technology can provide the most help are the finding, modeling, and storing ones. The problem of finding scientific data is very similar to that of finding any resource in a large distributed system (e.g., finding where machine "foo" is located) or in an information retrieval system (e.g., finding the right book in a large library). For modeling, database researchers have provided a wealth of models, from relational to object oriented to semantic. The scientific applications I have seen tend to have very simple models (e.g., a file of records with 10 fields each). The file may be huge, but the structure is simple. So in my opinion current models are more than adequate. The hard part is interpreting and translating the data (steps (c) and (e)). Finally, there also exist many known techniques for reliably storing data. There may, however, still be a few open questions such as management of very large objects and data compression.

For the interpreting, translating, and processing steps I feel that current database technology is nowhere near adequate. The main problem is that these steps are very application dependent. As stated earlier, there are some common functions that databases may provide, but most of the hard problems are beyond this. To translate, say, polar coordinates into cartesian ones one must actually "understand" the application. This coordinate translation is different and unrelated to say, translating a reactor temperature reading into an energy yield reading.

One possibility is to take a "black box" approach, where we view the database system as being a data repository plus a collection of black boxes that "connect" one datum to another. The black boxes encode the necessary translation and processing functions. Although this may be advantageous in some cases, in others I think it may just push the hard problems into the black boxes, for someone else to worry about.

Data Management for Genome Research

Nathan Goodman
Codd and Date International

To a biologist, the term data management means more than it does to a computer scientist. It covers all aspects of computing except data analysis. The main issues are graphical user interfaces (GUI), data visualization, and data management per se. I will limit my comments to the latter.

A data management system has four parts. The core is the data model, e.g., relational. On top is the language and programming model, e.g., SQL embedded in C. On the bottom are the architecture and implementation. On the side are operational controls. All four parts must be suitable in order for a data management system to work well in a specific application area.

Relational systems are the mainstream, best developed, data management systems today. Object oriented systems are starting to come out. A lot of new (and not so new) ideas are in research, including spatial, temporal, semantic, and non-first normal form (NF2) models.

Relational systems are not particularly good for genome work. The data model is not well suited for many important kinds of genome data, including sequences, maps, and images. SQL is not well suited for sequence searches and other important genome queries. The architecture and implementation cause performance problems. Nonetheless, in the near term, relational systems are the only game in town and we have to live their weaknesses.

The key near term requirements are training and database design tools so that biologists can use relational systems more productively.

The next step is to try out object oriented systems. These systems are new and incomplete. I believe in object oriented, but it's too early to tell how useful it will really be for genome databases.

The third step is to move new data models from research into practice. I would like to see the development of a usable prototype system that incorporates these new data models. This is likely to be an expensive undertaking unless we can find a way to channel existing research efforts into the project. Database integration is an important theme at all steps. Many individual researchers and labs collect genome databases. A key problem is integrating these databases into a cohesive whole.

NSF Workshop on Scientific Databases Position Paper

Gregory H. Hamm
Molecular Biology Computing Lab
Rutgers University

The biological sciences, and molecular biology in particular, present significant challenges for data management and database technology. These challenges stem from the nature of the science and the characteristics of the data that are generated. Both the amounts of data involved and the potential for their analysis are in a period of unprecedented growth. Present approaches to data management will soon fail to support the science, and indeed have never allowed full exploitation of the data already available.

A diverse science

Molecular biology derives largely from genetics and biochemistry, and many of its paradigms reflect these origins. The science is a "cottage industry", and its practitioners tend to group themselves according to model organisms and methods. As the science has become more data-intensive, these characteristics have led to the separate development of data collections, tools, and analysis methods which are used within, but rarely across domain boundaries. As a result, little effort has been devoted to designing these resources in a common way, or in ways which might facilitate their use together. Even relatively simple questions requiring coordinated analysis of data collections related in obvious ways (e.g., DNA sequences and three-dimensional protein structure coordinates) can often be approached only through the laborious construction of special-purpose software, a task rarely undertaken by biologists. As a result, the many data collections now available do not easily support investigations into the broad range of biological questions they might, in principle, help to answer.

Integrated management of biological data is even more difficult than integrated use. The data are very diverse, and their validation, maintenance and interpretation requires specific expertise in each case. This defeats immediately any vision of creating a single global repository for all biological data: even if all the requisite expertise were gathered in one place today, new methods would shortly provide data for which new expertise would be required. For the same reason, any single database design would not long remain adequate for the representation of all biological data, even if such a design could be completed once.

These problems have recently been approached through the construction of controlled-vocabulary cross-references and "backbone" databases. These efforts are intended to allow local experts to maintain the factual content of portions of a larger database, while permitting common central management of the relationships among the parts. These efforts will greatly facilitate management and exploration of data across existing boundaries, at least for closely related data sets. In the longer term, however, even these efforts seem unlikely to provide an environment in which all biological data can be easily explored, and into which broad new insights can easily be integrated.

Fuzzy data

Unlike many measurements in the physical sciences, biological data often have a fundamentally qualitative flavor, even when they are numerical. Primary data are usually about observables which represent the aggregate results of very complex behavior (e.g., the position of a band on an electrophoresis gel). These data must be interpreted in terms of specific experience to reveal information about the system under study. For this reason, present databases store interpretations rather than primary data (which are rarely published). Re-interpretation of results, often necessary because of the extensive interrelatedness of biological data, is therefore quite difficult. Future databases will need to reflect these complex relationships more explicitly if the situation is to improve.

The qualitative (or better, integrative) nature of many of the stored data also generate interesting analysis requirements. Biology often proceeds by analogy, through comparison of experimental results from different methods, or of results in different organisms or gene systems. These analogies are almost always inexact. There is thus a need to search the body of known results for patterns at a very high semantic level. For example, it is rarely helpful to search the database of known protein structure coordinates for the absolute positions of specific atoms, but it is immensely helpful to search for known structural motifs,

such as a certain juxtaposition of several alpha helices. This sort of search has only recently been possible, and serves to illustrate the kind of capability that will be needed across biology. Other examples include searches for similar biochemical "circuits", searches for patterns of evolutionary distance among groups of organisms, or perhaps for patterns of organization of genes within a genome or chromosome. None of these searches can be performed at the level of the data actually reported and stored, and none are easily possible with present approaches.

Prospects

Fortunately, many of these problems are receiving unprecedented attention, in part due to the widespread attention given the Human Genome Initiative. Data management problems which were perceived by only a few workers in specialized areas are made obvious by a project whose information needs are several orders of magnitude beyond anything yet attempted in the biological domain. As a result, elements of a solution to the database problems in biology are emerging.

First, computational science has begun to deliver technologies which have tremendous promise for biological databases. Examples include the object-oriented design paradigm, frame-based representational systems, special-purpose database hardware, and semantic search algorithms. Advances in computing hardware have resulted in nearly universal availability and familiarity with some level of computer, and wide availability of very high-performance computers. Second, biologists are becoming increasingly sophisticated in their use of computers, and in their abilities to state their research requirements in terms of computational strategies. Third, biological science is now posing questions which provide problems of fundamental interest to computer scientists, creating the possibility of effective interdisciplinary work. Finally, there is a growing community of "trans-disciplinary" workers whose expertise is centered at the interfaces of biology and computing, and who can provide much of the insight into how the two fields can productively interact.

One of the principal challenges in scientific computing in the next decade will be the development of database systems which can handle the inherent complexity of biological information. The existence and availability of such database will transform biological research. Meeting this challenge will require the construction of databases in fields where none are available, significant research and development in database and knowledge base technology, and the provision of a robust and widely available computational infrastructure for biological science.

Data Management for Climate and Global Change

Roy Jenne
Data Support Section, Scientific Computing Division
National Center for Atmospheric Research

We have a small data section (six people) at the National Center for Atmospheric Research (NCAR), whose job is to ease the problems of data access for research in Meteorology, Physical Oceanography, and overall climate studies and modeling. The group has one of the largest archives of data (over 350 datasets, and a volume over 16 trillion bits). Users with access to the supercomputers at NCAR (400 users at NCAR, 650 at universities) can also readily access the data online. Simple data access programs are provided to unpack the archived data, and present arrays ready for calculations. Learning time is very low.

Other users (from around the world) obtain data on magnetic tape. Most of our data access software is transportable to computers of various types. We now have an activity to provide more transportable access software along with the data (especially for the packed binary data).

1. Design for Evolution of the Data Systems

It is essential that data systems be conceived so that they can gradually accommodate good new ideas and ignore poor ones. This means that good systems are designed to cope with diversity, even though the amount of diversity is kept limited. The access to data and information about data should not be dependent on particular hardware or software systems.

DISCUSSION: When we consider the history of data systems, it is noted that there have been many changes on time scales of two to five years. There have been changes in data storage systems, concepts to define formats, and in computing capability. It is unlikely that a data system defined today will remain constant for the next decade. The design problem is to develop data systems that are flexible enough that the data and also information about the data (metadata) can be readily saved for the next 100 years even though changes occur. Thus, the hardware to store data will change and the software to help access data will also change. The basic data archive should enjoy relative stability so that errors are not introduced by frequent conversion and handling.

It is bad practice to change all formats in an archive every time a new idea comes along. Also, the metadata (information about data) must be structured so that it can be readily used by future systems as well as present ones. Thus, a ‘‘data system’’ should be conceived of as an entity that operates on data and provides easy access, not as an entity that owns the data.

2. Problems of Data System Design

It is far too common that data systems are only designed to accomplish a narrow range of functions that the designer thought of, or had time to accomplish. Then they are sold as being good for all functions of meteorology, astronomy, curing rheumatism, and climbing mountains. They are often conceived in the model of what is needed for an airline reservation system. They often are not flexible enough to handle input formats received from elsewhere in the world. This is not the model of what is most needed for scientific computing. For this, there are several needs:

- Prepare the necessary data in reasonable structures (not necessarily ‘‘common’’ structures). For high volume data, emphasize tightly-packed, efficient structures that control costs. Emphasize technical developments to decrease the actual cost of data transfer and access.
- Provide simple data access routines that handle the task of obtaining data from storage, and unpacking it so that it is ready for calculations. These must be efficient, and have a short learning time (from 5 minutes to 2 hours).
- Optimize data input methods so that the user can easily do his/her own computing using a higher level language (most commonly Fortran).
- Provide tools for graphical displays so that the user can easily prepare displays the way he wants them. This includes tables, plots, and VCR motion film.

- Encourage the preparation of a few ‘‘canned displays’’ that are commonly used in part of a discipline area.
- The world will have a number of data manipulation tools that are focused on certain tasks. Examples are: Lotus 1-2-3; a DBMS; and Unidata (meteorology). It must be easy to take an array of data from Fortran calculations and move it to one of these data manipulation software packages.

3. Relevance of Data Management Activities

It is sad that too often meetings about data management are not relevant for the main data needs of the users. In the Data Support Section at NCAR, we listen to the needs and troubles of users, day after day. Some examples are: (1) For mesoscale models, we need higher resolution datasets of earth vegetation cover, lake extent, and elevation. Can you help us? (2) We want to use modern analysis methods to reanalyze all of the world’s surface land, ocean, and upper air meteorological data from 1958- on. Can you help us obtain the necessary data? (Yes, but for minimum standards of data completeness, this will be a very large task). (3) Can you help us obtain a lot more data over Africa? (4) You are just emphasizing access to data at NCAR, or delivery of data on half-inch tape (and some by communications). We want to have easy access to large amounts of your data at our university department, at low cost, using technology such as CD-ROM disks, DAT tapes, or EXAbyte. We want data access software that makes it easy to make calculations using Fortran.

In many cases the focus of data management discussions is on data browse and display, not on the central problems that we and users face.

4. Complaints About Data Access

Some complaints about data access and data systems can be paraphrased as follows. The last CD-ROM we got has about 200,000 grid analyses showing meteorological fields over the Northern Hemisphere for the past 30 years. The access software was set up so that we had to interactively access each field. We wanted to do serious computing using thousands of fields. We did not want to waste months of our personal time to accomplish what was most appropriately done in an easy batch mode (This is now fixed). We want to make calculations without fighting a system, and without paying a huge cost of system overhead. Then, we want to prepare displays of the results the way we want them to look, not the way a system decides they should look. We welcome tools to prepare these displays, but don’t tell us that we should spend two weeks to learn your system and memorize hundreds of cryptic commands before we can do useful work.

5. Catalog Systems for Information About Data Availability

At the national and local levels users must have easy access to information about major datasets and where they are held, but only the most necessary information is needed at the national level. More detailed information has to be readily available at the data centers. A national data catalog system, coordinated by NASA, is under development. Nearly 1,000 datasets are already described in the system.

A catalog system will often give the user more details than can be assimilated. For a given subdiscipline area, we also need brief overview documents (online and text) that present the most important information. Examples are: satellite sounder data, data on US energy statistics, etc.

The preparation of these overview documents will also encourage people to think of what data should be prepared into datasets.

6. Data Structure

Over the years people have found it very useful to organize a given dataset using one to several files, each composed of physical and logical records. To efficiently handle many types of data, we must retain the ability to easily store and retrieve variable length records. Given this need, why do people periodically talk as if their new ‘‘systems’’ will no longer support these structures?

Data structures that we use for data processing make it possible to reorganize data at very high speeds, and to quickly select data subsets. We are not willing to give up this capability. Example: we have a dataset with weather reports from the world’s ships 1854 - 1980. It has 72 million reports (each with about 40 numbers). We can completely invert this file (volume five gigabytes) in four hours of

CRAY-1A time. Most people now agree that to handle large datasets, it is important to keep the data in a simple file structure, and to keep the data flow fast.

7. Problems of Bulk Data Packaging

One of our key services at NCAR is to be aware of the problems of data availability (or access) that people are having, and to help solve these problems. In many cases, problems arise because low volume data and high volume data are together in one file. Some examples are: (1) Analyses and observations (now on 50 tapes) were hidden in 2,000 tapes of data; (2) One set has monthly averages integrated with 22 tapes of daily world station data, such that the small volume of monthly data is hard to access; (3) A satellite dataset has 800 tapes, yet the main data that people need could be put onto about two tapes. We are always solving a few problems like these, and sometimes encourage other groups to help.

In addition, for the larger datasets, we maintain very fast software to select data subsets based on time, region of world, variable needed, etc.

8. System Cost and System Needs

We need to strive for cost effective data systems. An analogy is useful. In a family, it is usually easy to justify the purchase of an automobile. We need it for work, shopping, pleasure, etc. How many of us, therefore, purchase a Mercedes? Building data systems often present us with similar choices. We need storage, communication, media for data transfer, access software, display software, etc. We need to choose these resources to maximize the main goals:

- Prepare the data necessary for research.
- Operate observing systems to obtain necessary observations.
- Provide easy access to required data for research (at both local and remote sites).
- Decrease the time needed to access data, and to learn systems. We want results here, not slogans.

9. Data Activities That are Needed

The following national data activities are needed:

- Prepare (and gather) the needed datasets and selected subsets. This is the main problem retarding the sciences.
- Promote the use of low-cost technology (eg CD-ROM and DAT) to distribute data. Provide portable data access software with the data. Promote other technology to reduce the real cost of giving access to high volume datasets.
- Encourage the development of lower cost mass storage systems that have one-time cost categories of (1) $700 (use CD-ROMs or DAT-manual mount), (2) $3,000, system with small carousel, (3) $20K, (4) $125K.
- Develop better solutions to the ‘‘data packaging problem.’’ Let users define their own formats. But develop a few variable length blocking structures to ‘‘package’’ the data. These would be efficient, have checksum protection, and could be used for transfer of data between computers as well as for local storage.
- Prepare translators to change data from ‘‘simple array structure’’ to a structure needed by one of the interactive data manipulation packages (an image analysis package, a PC publisher, Lotus 1-2-3, etc.).
- Many scientific groups prepare a battery of software to help work with their type of data and display it. Promote activities to gather selections of this software.

10. Papers of Possible Interest

- **Data Management Methods; Data for Europe**
  This paper discusses data strategy, data progress, and gives a history of selected data ideas over the last 20 years.
- **What Types of Data Access do Users Need?**
  A few short papers review how scientists use data for both calculations and browse functions. This

includes examples, a paper on "Levels of Data Support," etc.

- **Planning Guidance for the World Climate Data System, 1982**
  (by Roy Jenne) Surveys data requirements. Lists the variables. Discusses data management strategy. It is document WCP-19, available from WMO, Geneva. The planning for climate includes most of the variables needed for "Global Change."

- **Other Papers**
  A list of documents that discuss selected data types, available inventories, data management, etc. is available.

The Global Laboratory

Anita K. Jones
Department of Computer Science
University of Virginia

The computational mode of pursuing science has come of age. In some fields of science it depends heavily on the availability of data. Important as it is, the data fits into a larger context. Scientists in many fields are joining in a loosely federated, global laboratory in which a scientist is electronically connected to all other sites *of interest*. A site may be of interest because of the presence of other scientists, or the presence of data collections or an instrument (control).

This trend toward a global laboratory in which data will play a large part will only increase. And *economics* plays a part in determining what the computational support in this global laboratory will evolve to be. So it is worthwhile to consider what will be expensive and what will be cheap.

- Raw compute power was expensive in times past, but is much cheaper now.
- Memory, particularly secondary storage which has mechanical parts, remains slow and cost per bit is on a modest, but not a dramatic, decline. Dense packed, slow-access, removable media are reliable for a short number of years and readily transported.
- Data is accumulating at multiplicative rates — much faster than the cost to store it is going down.
- Network connectivity is almost universal and is cheap, having been greatly subsidized by the U.S. government.
- Dramatically increased network transmission speeds are predicted to remain inexpensive, relatively speaking.
- Visualization graphics have proved useful. Physical constants, like the speed of light, constrain the graphics-related processing to be very close to the display screen.

It leads me to believe that the global laboratory will involve multiple, but few, large data collections in each science. Economies of scale and maintenance will favor large sites. Scientists will routinely have high-end workstations with some graphics capabilities. They will be connected to the rest of the laboratory by wide- and high-bandwidth data pipelines. Scientist workstations — say, the visualization intense Stardents — and large compute engines — for example, the Crays — will be caches for the data flowing through the data pipelines.

Such a cache — holding for example a few 10s to 100s of gigabytes on a high end workstation — will hold the data of interest for seconds, minutes, hours or a few days. But economics will make it less costly to re-retrieve the data rather than to store it. Bits will routinely fall on the floor".

If this vision is only partly correct, it will put a premium on an efficient and humane solution to a number of computer science problems:

- browsing — Effectively, all data will be remote. With data, and derived information, being inserted into the data collections daily, browsing needs to be cheap, focused to serve the scientist's objectives, and fast. The browsing activity should result in returning modest amounts of relevant data to the scientist as well!
- transmission — When wanted, the data must arrive in a timely way. Users strongly prefer computing services to be predictable. The network will influence such predictability more than the compute engine in some situations. Predictable transmission latency will be prized.
- metadata — Metadata will have to be richly descriptive and accurate in order to service both browsing and the computation algorithm needs.
- knowing when to throw data away — Because local scientist memory will frequently be used as a cache, that memory will saturate much more rapidly than today's file systems — so rapidly that automated purging will be required. The smart, efficient, automated garbage scavenger will be highly prized as will a data compressor demon that repaces the actual data with the set of actions needed to re-retrieve the data.
- interfaces — Data format and systems are and will remain heterogeneous. Interfacing them better will ease the scientist's difficult task of dealing with unfamiliar data and systems.
- fusion — New science will result from the merging of data from traditionally unrelated sources.

The basic paradigms for fusion of data will be stressed and should be enriched.

- tailored data management systems — The data collections are large enough and costly enough to maintain, that they warrant having data management systems whose interfaces, and possibly internals, are tailored to the application and its unique attributes.

Biological Databases Contain More than DNA Sequences

David T. Kingsbury
Department of Microbiology
George Washington University Medical Center

Biology is inherently information-rich because of the complexity and variety of living systems. Understanding these systems requires information about their organization, structure and function at a multitude of levels from the macroscopic to the molecular. Moreover, each species, and in many species each organism, represents the potential for a unique solution to the problems of life processes and the organism's interaction with the environment. The full understanding of biology requires extending organismic complexity to include the relationships of species and organisms in their ecological niches, and the evolution of the biosphere over time.

Until recently, much of this information was accessible to scientific inquiry only at relatively high levels of abstraction: the inherent richness of information was not reflected in the volume of data available. This situation has changed rapidly over the last decade in a number of biological fields, among them molecular biology, neurobiology, ecology and taxonomy. Emerging scientific paradigms will require substantial bodies of data organized into large and dynamic databases to support ongoing biological research. Indeed, some aspects of modern biology (e.g., the Human Genome Project, protein structure-function studies) are now utterly dependent upon database and computer technology. In many cases current data collections are not well organized for ease of retrieval but remain central to work in a given field. For example, the major source of data about the genetic map and much of the information about specific markers of the widely studied bacterium *Escherichia coli* , is a printed review. Because of the heavy reliance of the research community on these data this review was the most widely cited paper in the biological sciences for the two years following its publication. Further, the reliance of modern molecular biology on databases is exemplified by the heavy dependence on GenBank, the DNA sequence database, over the past 7 - 8 years. At the present time it is impossible to publish new information about a gene sequence without providing evidence of prior examination of the gene sequence database and reporting the percent homology with existing sequences, if any exist.

As as result of these changes, a substantial number of people now devote their careers to data management and computational analysis in biological disciplines. However, despite significant and robust efforts, the present generation of biological databases will fail in the next decade because they were not (and could not have been) designed to deal with the volume, complexity, and diversity of the data which will need to be accessible for biological research. For example:

- a single Purkinje cell is thought to have in excess of 100,000 independent inputs;
- the human central nervous system is thought to contain 10^{10} cells, and 10^{18} synapses;
- the human genome contains some 10^{9} bases, for a "genome space" of 4×10^{9} states;
- a stock center for *E. coli* needs to describe 20,000 strains, each with 3×10^{6} nucleotides;
- taxonomy and systematics need to organize 10^{7} leaves (i.e., species) into a multi-versioned tree.

This "explosion" of data is obviously severe in a quantitative sense, but is equally daunting in terms of its diversity and the interrelationships which must be represented, organized and maintained. These problems are further complicated by the nature of biological research: the data are generated by geographically dispersed researchers with relatively little standardization. Taken together, these problems pose unique transdisciplinary challenges for database design. Indeed, it is important to note that the technology is not yet in hand which can support the design of adequate databases for much of the biology of the next decade. In fact, the application of the term "database" itself tends to trivialize the problem and is misleading. Database technology and theory as it currently exists is an inadequate paradigm for what is needed now and in the future to represent and organize biological information.

Scientific Database Workshop Position Paper

Barry F. Madore
Infrared Processing and Analysis Center
California Institute of Technology

The NASA/IPAC EXTRAGALACTIC DATABASE (= NED) is funded by NASA and run by astronomers at the INFRARED PROCESSING AND ANALYSIS CENTER on the CALTECH campus. NED is now in beta-test and will become operational in the summer of 1990. We are populating a relational database so as to provide members of the astronomical community with free access to data on extragalactic objects (quasars, galaxies, etc.) as published in catalogs and in the literature. These data will be linked by extensive name cross-referencing and unique positions on the sky. In addition to the data, extensive pointers to the literature will be made available by object and by attribute. With the literature we are also providing on-line access to the published abstracts to contemporary articles (including a knowledge-based software package which will allow the use of plain English queries to retrieve relevant papers).

Astronomers at universities and research observatories around the world will be using and contributing to this effort. International collaboration is exceedingly important to us and we would hope that the NSF will be able to provide guidelines for the community on how best to co-ordinate these efforts with the publishers of journals, the private and public operators of observatories and research laboratories, and the many cognizant governmental funding agencies. Our own small efforts on all of these above issues have so far met with very encouraging results.

On building a National Human Genome Database

Thomas G. Marr
Computational Biology Group
Cold Spring Harbor Laboratory

The goal of the Human Genome Initiative is to completely characterize the genome at the molecular level by mapping and ultimately sequencing the entire genome. The Human Genome Initiative presents scientists and engineers with problems that truly will push the state-of-the-art in computer science in the areas of database, algorithms, networking, and visualization.

There are many existing technical problems associated with building a national human genome database:

(1) The data are generated nationally and internationally in a distributed and mostly uncoordinated way.

(2) There is mostly undetermined and/or unspecified imprecision and uncertainty in all mapping data at all levels of resolution, including the sequence level.

(3) We will have multiple concurrent versions, through errors and polymorphisms, of what are, in principle, the same underlying entities.

(4) As the rate of nucleotide sequencing increases, the need for automated biological feature extraction will become critical. Sequence analysis will become difficult within a typical relational system as the amount of data increases, because of system constraints.

(5) Physical maps are derived, stochastic entities and multiple versions often exist. Ordering clones and restriction fragment mapping can be shown to be NP-complete or hard, and thus combinatorially explosive, if data are generated only in a random fashion.

(6) The community of involved scientists is expecting maximum logical connectivity with all relevant data. This can only be accomplished with richly annotated metadata, thus adding to the complexity of the problem because we will need expert annotators from diverse fields connected directly and remotely to the database; computer tools for browsing and linking metadata are not well developed.

(7) The most technically mature database systems available now and for the near future unquestionably are relational systems, but there is a large semantic gap between the type of data we want to manage and the relational model. Object oriented database systems appear to be more appropriate, but they are not technically mature.

(8) The existing scientific literature, to which the sequence needs to be connected, is not well organized, is electronically inaccessible and yet is essential to the correct interpretation of the genome sequence.

Data Validation - A Major Problem in Scientific Data Management

Robert McPherron
Institute of Geophysics and Planetary Physics
UCLA

Modern instruments and computers make it possible to acquire enormous files of digital data. These data are an immensely valuable resource but their management poses some significant problems. One of the most significant is data validation. We propose that the current system of research funding is not conducive to the creation of validated data that can be used with assurance by secondary users. We advocate several changes that could help alleviate this problem.

A major source of data is the principal investigator or PI. The PI is an individual scientist, or group of scientists led by one individual who build an instrument, operate it for long intervals of time, process the data, and write scientific papers with the data they acquire. Another important source of data is operational systems. Magnetic observatories, weather stations and satellites, nuclear surveillance spacecraft, remote sensing satellites. All contain instruments that routinely gather data used by the scientific community.

State of the art instruments which advance the frontiers of knowledge rarely perform as the designers expect. The data are often contaminated by a variety of effects within the instrument and in the immediate surroundings. These effects must be removed from the data by appropriate computer processing, or taken into consideration when the data are used for scientific studies. Data from operational observatories are often processed by technical staff who are disconnected from the instrument builder, and are not familiar with the eventual application of the data. Not infrequently they unknowingly introduce errors into the data, or fail to remove some that they should.

The principal investigator system is a sociological mechanism for producing good data and validated scientific results. The central feature of the system is that in return for producing important scientific results, it gives the PI both financial resources and exclusive access to data from his instrument. These resources allow him to carry out the operations on his data that make them useful in scientific studies. The system rewards the individual for quality control by making it easier to win future projects if he has done well in the past. Doing well depends on successful publication of scientific results. This requires peer review of his papers. Often errors in data are caught in this process by the PI's competitors. Doing well also requires recognition in the form of citation. Again, others may discover errors in data through inconsistencies with previous reports.

Frequently, operational systems do not have the feedback of the peer review process to guarantee quality data. The success of individuals in these projects is measured by criteria other than scientific results. Was a functional instrument delivered on time? Were the data transformed and deposited in the archive in a timely manner? Often, it may be years before the data are used by some scientific investigator. Such a system is prone to hidden errors that can catch the unwary.

Today there are fewer missions or projects with far more expensive instruments. Fewer individuals are able to succeed as PI's of instrument projects because of the scarcity of resources. Individuals who are not the original instrument builder must work with data placed in archives by PI's and operational projects. Today, also, the nature of many disciplines is changing. The exploratory phase is over. Current research depends on studies which make use of data from many instruments on many platforms. The fundamental question is how can the secondary user be certain of the quality of the data he or she uses in their work?

In response to this need for access to high quality data by users other than the data producer, government agencies are developing scientific data centers. Thus far, most scientific data centers are subsidized by the government. Commercialization is unlikely to be successful because the scientific users has no paying customer other than the government to which they can pass the costs of data purchase. Very few scientific data centers could survive if they had to function on the basis of data sales because the scientific users have very small data budgets. Examples of data centers include the NOAA National Geophysical Data Center in Boulder, CO and the attached World Data Center for Solar Terrestrial Physics. Another example is the NASA National Space Science Data Center in Greenbelt, MD and its attached world data center for Rockets and Satellites. Today these data centers are the final repositories of data contributed by PI's, Instrument Teams, and Operational Systems. An important question that any user must ask is "Are these data of sufficient quality to answer the questions I am asking?"

Today the quality of data in national data centers is frequently unknown! Much of the data is in formats so poorly documented it is impossible to retrieve without an effort exceeding the resources available to a secondary user. Even when the data are retrievable, there is often no documentation describing the problems in the data. The secondary user is required to go to the original investigator for the validation of the data, and in many cases, access to the data he needs. Sometimes the PI is no longer available, or is a competitor of the secondary user, or has no resources himself to support the project.

In our opinion the PI system fails to deal properly with the problem of long-term archiving of data. In its current form some PI's do not perceive any advantage in placing their data in the data center. It takes time and resources that are to their advantage to use to produce papers that make them more competitive. It is often better from the PI's point of view because he can gain partial credit if he supplies the data directly. But almost none if it is provided by a data center. Also, it makes it possible for others to do work that they might eventually do themselves. Finally, it exposes the problems of their instruments to the world. Users may discover that their instrument was seriously flawed. This fact can be hidden by appropriate choices of published examples, but not by open access to their data.

Given that a PI has done everything right and wishes to submit his data in a useable form it may not be possible. Often the resources of a single PI exceed the total resources of a data center. It is simply not possible for the data center to handle all of the data potentially available.

The NASA Planetary Data System is attempting to solve some of these problems. The PI and Instrument Teams are required by contract to submit their data to the data center. Funds are set aside in advance to assure that this happens. The various Nodes of the distributed data system are given incentives and resources to acquire the data from PI's. Additional resources are available to PI's from the project to submit properly validated data. Data submission requires a variety of tables including instrument description, data set definitions, parameter characterization, and standard volume labels that make the data readable in the future. Once submitted the data are validated by a peer review committee. Data are accessed according to the information provided. They are compared to examples of published data. Browse data sets are created and scanned to see if all of the data specified in the catalog are actually present. Discrepancies are identified and measures taken by the data center or the data supplier to eliminate them.

Such a system can not help but provide better data to secondary users. However, the system is expensive. It is not viewed favorably by all PI's. Many feel that the money required to manage this data would be better spent by individual scientists doing research.

New incentives must be developed to guarantee that high quality data are produced by instrument and data processing teams. These data and metadata must be captured quickly and simply in machine readable format at the time they are generated. All of these must be stored in a location easily accessible to qualified researchers, without the intervention of the data producer. Some mechanism must be developed so that credit can accrue to the data supplier without his intimate involvement in every project. There must be changes in the level of funding that are provided to instrument builders in comparison with data analysts. Additional resources must be made available to those who manage data after missions and projects are terminated.

We believe that the NASA Planetary Data System has developed some unique methods for obtaining and validating data from its planetary missions. In particular the peer review process for data submission is one that guarantees much higher quality data than has previously been submitted to data centers. This process should be considered by all data centers as a means of obtaining validated data sets. If this process were combined with the new publication format of the American Geophysical Union, the Brief Data Report, it might become more advantageous for PI's to submit their data to data centers since they could receive citation to their efforts by other investigators. Such citations would constitute proof to funding agencies that their work had been worthwhile.

The Astrophysics Data System

Stephen S. Murray
Harvard-Smithsonian Center for Astrophysics

Introduction

From the point of view of an active research scientist, the explosive growth in the amount of data available on a subject, and the diversity of data sources is often overwhelming. Some of the fundamental problems the researcher has in dealing with this situation are:

What data are available?
Where is this data?
What is the form and content of this data?
How can this data be browsed?
How can the data be obtained?

Beyond these issues are the more familiar questions of what can be done scientifically with these data and how were the data generated, reduced or otherwise processed before being distributed.

In the discipline of space astronomy and astrophysics, these problems become especially significant as the era of the ''Great Observatories'' approaches. These are a series of large scale national facility missions of long duration in several sub-disciplines:

The Hubble Space Telescope (HST): Now scheduled for launch mid-April 1990. It will return in raw telemetry over 1 terabyte per year, and operate for at least 10 years.

The Gamma Ray Observatory (GRO): Scheduled to be launched in June 1990. It will also be a long lived observatory returning large quantities of raw data for analysis.

The Advanced X-Ray Astrophysics Facility (AXAF): Current schedules call for a mid-1997 launch and a 15 year lifetime. Several hundred gigabyte per year raw data rate.

The Space Infrared Telescope Facility (SIRTF): Probable launch by the end of the decade (century), and a 5 to 10 year lifetime with comparable data rates of AXAF and HST.

In 1986 NASA organized the first in a series of two workshops to develop requirements for an Astrophysics Data System (ADS) that would be in place to support the scientific user community in learning about and accessing the data from these missions as well as the many other space astronomy missions being planned for the 1990s and beyond. The ADS workshop report (the ''Squibb'' report) called for setting up a data system that would connect the various mission centers where data are collected, processed and archived, with the users who are distributed throughout the country. The report stressed the importance that data archives be ''active'' research sites where the expertise associated with the source of these data is also maintained. Additionally, the report recognized the need to have migration paths for data and information from mission sites to less active archival centers as the operational phase of a space project ended. While these points seem obvious once stated, making them a matter of NASA policy was a major step forward for the field. The first priority of this data system was to establish a ''Master Directory'' that could be accessed by the community to identify and locate the data holdings from NASA supported missions. Following these community wide workshops, and Astrophysics Data System Working Group was formed with the goal of designing a data system to meet the needs of the community and to begin the implementation of this system with a Master Directory function.

The Astrophysics Data System

Among the recommendations of the ADS Workshop report was the desire of the astronomical community to be involved in the process of designing and implementing the ADS. In selecting the membership of the working group, NASA followed this recommendation, and put together a group of scientists and computer systems specialists who were currently involved in data systems for active missions, along with a number of active scientific users of these data. This results in the end users of the system being actively involved in the system, guiding its development from the perspective of the user, with a balance of input from those technically knowledgeable with regard to implementation limitations. The working group expanded during the initial design study to include participants from industry whose products would be included in the system. Again, the guiding principle for the ADS development was to include the users and the developers in the design and specification process. This is a departure from the typical NASA procurement process for systems which involves specifications and contracts.

System Requirements

While formal requirements and specifications of the ADS were avoided in favor of a rapid prototyping development process, the working group did establish certain overall system requirements to guide the detailed development. Among these were the desire to provide useful service to a wide range of potential users of the system. A minimal user, working with a modem and a personal computer needed to be supported. Similarly a scientist connected to a central computer through a ‘‘dumb’’ terminal also requires access to the ADS. Finally, the ‘‘power’’ user with a graphics workstation should not be constrained by the limitations of the less well endowed users. This meant creating a system with varying levels of ‘‘local’’ resource (CPU, memory, storage, display) requirement, and one not overly dependent on network throughput. Another requirement imposed on the system was the ability to handle a wide range of sophistication in the user population. The system must offer a variety of levels of operation from completely menu drive operation to command driven operation. Finally, the system must be responsive to the needs of the user community. The initial system will reflect the best guess of the working group, but only through use by the scientific community will the system requirements become known. Thus the underlying design needs to be capable of evolution and extension without disruption.

System Architecture

Early in the design discussion of the ADS a sample user scenario was developed as a aid in determining the required functions for the system and to help maintain a focus on the goals of the system. This has proven to be an important tool. The scenario itself has evolved over the course of the ADS development, but it serves as a reference to assure that the system design meets user requirements. The development approach for the ADS is to avoid formal requirements and specifications for the system, rather rapid prototyping and mockups are used to test ideas against the scenario and the ‘‘vision’’ of the system that the working group has formed.

During the process of visualizing the Astrophysics Data System, it became clear that a Master Directory function was just a first step in a more ambitious system. This lead to the concept of a server--client architecture for the ADS in which the Master Directory would be an initial service offered to the community. The basic building blocks of the ADS would be a user agent (UA) providing a uniform interface to the system users, a transaction manager (TM) to administer the activities of the system, a message passing system (MPS) to provide the communication protocol between system elements, a data independent access method (DIAM) for connecting heterogeneous data systems to the ADS, and information access method (IAM) for handling non-traditional data, especially textual descriptive information. These five components, along with the backbone provided by a network system (which is not part of the ADS so much as used by the ADS) form the infrastructure out of which the ADS can be built. Each component is implemented as a client or server, held together by the interface specification of the MPS.

Initial Components

The working group spent considerable time in selecting implementations of the various key components of the ADS. The guidelines were to find existing products where possible, and to extend or modify these products to meet the specific ADS needs. These elements must be broadly, avoiding vendor specific restrictions and distribution fees. In practice this goal was not completely met. However, a measure of the level

of success achieved is that the system will be available to all qualified users. A benefit of the client-server or server-service model for the ADS is that it allows for evolution of the system and replacement of any component by a newer version or entirely different implementation provided only that the interface to the ADS system as a whole was not violated.

The initial components of the ADS are:

User Agent: A commercial product adapted specifically for ADS was selected and developed. It is the Knowledge Dictionary System™, which is from Ellery Systems Inc., Boulder CO. This provides the user interface, a test editor, a command interpreter, and a local database system. The KDS™ also has elements of the transaction manager and message passing system embedded in it so that the user agent can interact with the rest of the ADS.

Message Passing System: A ''semi-commercial'' product was selected for ADS use. In this case ESI has obtained the rights to a message passing kernel based on the ANSA TestBench architecture developed at a U.K. consortia of universities and industry. The elements of the MPS are embedded in the various components of the ADS in order to support communications and control of processes.

Transaction Manager: This function has two components. Within the KDS™ user agent there is a local transaction manager that protects the user's local database. As a service for the ADS there is a transaction management and administration service that assures orderly management of the ADS services. The KDS™ function is part of the user agent and is currently working. The system function has a preliminary design and is under development at ESI.

Data Independent Access Method: The ADS was able to benefit from another NASA project, the Distributed Access View Integrated Database (DAVID), being developed at the GSFC/NSSDC. For the purposes of the ADS, DAVID provides a homogeneous interface to the variety of mission specific data collections and their underlying database systems. In addition, DAVID supports the additional operators that are needed for astronomical applications and are not necessarily available within standard commercial data base management systems.

Information Access Method: The underlying technology of the KDS™ consists of a information access method based on classification space. The particular application to ADS is the judgement space (j-space) methodology, wherein textual information can be indexed in a multi-dimensional space for retrieval based on relevancy to the query being made. Furthermore, the query can be posed in natural language, using the peculiar jargon of the field familiar to the scientific users. All descriptive material can be treated in this fashion so that data, data products, images, etc., can be processed by indexing description of these items and referencing the item to its description.

Project Status

The development of the overall system has been underway for about 18 months. The user agent software was initially developed to run on PC class computers and has been a) re-written in *C* and b) ported to UNIX from MS-DOS. This includes all of the included functions of KDS™ such as transaction management, information access, and local database management. Specific changes to the system were also made for the ADS such as enhancements to the user interface, the database system and export functions to pass data to external user functions. The user agent runs on SUN/4 workstations under SUN OS/4, and on PC's under MS-DOS. These are the targeted systems for initial release of the ADS. Future plans include porting the UA to other UNIX based workstations, MAC personnel computers, and perhaps VMS based workstations.

The DAVID system has been installed at a number of ADS data sites and interfaced with local DBMS systems at these sites such as INGRES, IM/DM, and Sybase. These installations of DAVID are designed to appear to the user as a single data system, the distributed nature of the underlying data is managed within DAVID. Modifications to DAVID were made to deal with special data transformation functions required for astronomical work such as the establishment of a canonical coordinate system for objects. Future work is expected in the development of standard gridding the ADS data to improve access times for often repeated searches. Browsing capabilities are needed to give users the opportunity to scan data sets before carrying out major manipulations.

The MPS has several components. The main element is called the ''trader'' which runs in the background on all of the systems where servers and services need to be arbitrated. Within the clients and servers, there

are elements of the MPS which send and receive messages that contain service requests and the data acted upon or returned from the service. At this time the MPS has been installed at one data site, but on several computer systems - SUN/3 and SUN/4 running SUN OS/4, and microVAX III running VMS. At this same site, the message passing kernel elements of the MPS have been integrated into versions of KDS™ and DAVID so that end to end system tests can be run. A successful test of this prototype was carried out in mid-February, where data from the Einstein Observatory observation log and one of the point source catalogs were accessed via queries from the user agent. These were passed via the MPS to DAVID and on to the local (INGRES) DBMS.

The information access component of ADS has several aspects and functional requirements. In the simplest form, the ADS needs to be self-explanatory and to provide novice or infrequent users with enough information to allow simple functions to be carried out quickly and easily. For this purpose, a menu driven interface was developed that was intuitive to use and allowed the user to scan through general information about the ADS and its data holdings in a hierarchical inverted tree structure. An initial set of descriptive documents were generated to populate this tree and test the menu driven system. The user agent will be distributed with this data tree so that users will not need to depend on network connectivity to access top level information.

An alternative to information access via the logical traversal of a data tree is provided by the j-space technology discussed above. As part of the ADS development a preliminary j-space has been created and the information contained in the information tree has been indexed in this space. The user agent provides the tools needed to allow natural queries to be made against this information and returns a list of the most relevant pieces of information (documents) to a query, along with their locations in the data tree, users can then access these items directly from KDS™. Additional information is being added to the data tree providing greater detail for the user. As the amount of information managed using this system increases, services providing indexing, query resolution, and document retrieval will be needed. These are considered part of the ADS and are under study. For example, the processing of queries in j-space is amenable to parallel or vector processing, either by a central parallel computer or by distributed computing, and such an effort is underway.

Schedule

The current schedule for ADS calls for completing the initial development by June 1990 and a beta test phase extending through September 1990. Full release of this system will begin in October 1990 subject to the availability of operational funds for the ADS data sites, and the central administration of ADS. In addition, there is ongoing development of the system. Plans call for resource management services, data processing services, data browsing, and improvements in the interface. It is expected that there will be user contributions to the system in the form of requests for new services, suggestions for improvements, and also in the direct involvement of the community in developing services to be added to the ADS. NASA has already begun funding of grants through its Astrophysics Software and Research Aids Program with the requirement that these activities are coordinated through the ADS program and where appropriate meet the specifications of that system.

Data Management for Genomic Data†

Frank Olken
Information and Computing Sciences Division
Lawrence Berkeley Laboratory

Scope

In this paper we are concerned with genomic data: data having to do with DNA in organisms, its structure and function. Such data include: ‘‘lab notebook data’’ which describe various molecular biology experiments and their results, various types of maps of chromosomes: cytogenetic, genetic (recombination) linkage maps, physical (restriction enzyme cutting site) maps, DNA, RNA, and protein sequences (and the accompanying ‘‘annotations’’ describing the function of various subsequences), protein structure data, protein function, metabolic pathway databases.

In this paper we will briefly mention several of the major problems in storing and retrieving this type of data. We will emphasize mapping and sequence data, due to the author’s familiarity and space limitations.

Sequences

The first observations to be made about genomic data is that it is largely composed of sequences: DNA sequences, RNA sequences and protein (amino acid) sequences.

Conventional DBMS’s are built around set theory, and provide poor support for sequences (you may be able to store a sequence in a record). The research community (DB, and functional programming language) has begun to address this problem with work on ‘‘streams’’.

Commonplace activities with respect to this data are: approximate string matching, approximate pattern recognition, and approximate parsing. There has been extensive research on the first topic, some on the second, and very little on the third. Most of this research has been concerned with random access memory (RAM) based models of computation and has not concerned itself with secondary storage considerations.

Functional Maps

The functional annotation of DNA sequences (e.g., gene complexes) can be thought of as a parse tree over the DNA sequences. (Actually, directed acyclic graphs (not always a tree) since portions of a DNA sequence may have more than one function.) One would like to store these ‘‘parse trees’’ in a database and be able to compare, and query them. Again commercial relational databases do not support such queries very well. There has been some research on parsing DNA sequences (Searls, et al.) There has been some work on approximate matching of trees (Shasha).

Maps

Another ubiquitous type of genomic data are genomic maps: descriptions of the spatial structure of chromosomes. Human chromosomes are linear, those of lower organisms are circular. Maps vary in their completeness, resolution, metrics, etc. Typically, the researcher will perform a set of experiments which increasingly constrain the set of feasible maps until a unique map is obtained. Because maps are one dimensional, map data is similar to the sort of temporal data which arises in scheduling and planning applications. There has been extensive interest in these problems in the AI community (Allen, Ladkin, et al.) Map construction is thus analogous to constructing feasible schedules.

In addition to construction, approximate map alignment (similar to sequence alignment) is required. Recently there has been work on map alignment (Rudd, Webb, Huang, Myers) algorithms.

Existing relational DB systems provide neither the inference mechanisms to support map queries, nor the ability to manipulate maps (ordered sets of intervals), nor the ability to answer approximate map

† This work is supported by the Office of Health and Environmental Research Program of the Office of Energy Research, U.S. Department of Energy under Contract DE-AC03-76SF00098.

alignment queries.

Three Dimensional Data

Molecular biological researchers are interested in the 3D structure of the proteins (esp. of the active sites). Typical queries ask to find either similar or complementary structures — or more generally structures which contain similar or complementary substructures.

Again, relational DBMS's are of little use. Current methods of finding similar structures appear to be very compute intensive, but recent work on n-gram statistics looks promising (Rackovsky).

Experimental Data

There have been attempts recently to construct Lab Notebook databases to record experimental data on relational database systems at LANL, LBL, Baylor, et al.

Such databases are typically multi-media, i.e., they include either image data or time series instrumentation data. Existing DBMS are just beginning to accommodate large objects (e.g., images) and have as yet few facilities for manipulating such data.

Aside from the multi-media aspects, lab notebooks tend to include a lot of *ad hoc* information. Furthermore, the structure of the lab notes change as the experimental protocol changes. This can happen quite quickly in a molecular biology lab.

Existing techniques for designing and restructuring relational databases are hard pressed to cope with such frequent changes, and are quite unable to cope with *ad hoc* data. Proposals to develop hypertext systems to address this problem are interesting, but it is unclear how one processes queries against a hypertext DB.

Another aspect of lab notebook data consists of formal specification of laboratory protocols, both to facilitate interpretation of lab data and to support automated (i.e., robotic) protocol implementation. The problem of specifying molecular biology protocols is not unique — similar problems arise in chemical labs, drug screening, and manufacturing.

Position Paper on Biological Databases

Gary J. Olsen
Department of Microbiology
University of Illinois

My interests are primarily in the field of molecular biology. As areas of biology go, molecular biology includes a substantial amount of "well-structured" data — most notably nucleotide sequences, protein sequences, and data on the three-dimensional structure of molecules. Some other data that are reasonably well-structured, such as bibliographic information, are also of interest in many other fields.

Unfortunately, molecular biology is not a mature field in the sense of having prior knowledge of what information will be useful in the future. Thus, information about function or "interesting features" that might (in principle) be amenable to fixed-vocabulary, keyword descriptions, can change rapidly. Many other types of data are sufficiently interpretive in nature that they are not easily cast into standard formats, and their interpretation may be dependent upon experimental parameters that are not easily summarized in a standard form. For present purposes I will not dwell on data for which there is no clear representation beyond text and graphics.

Even limiting consideration to the structural data (including the sequence data), the items of interest frequently require extensive "manual" intervention for their abstraction from the existing databases. This is a consequence of several factors:

(1) A researcher's interest is frequently for a specific subset of a database "entry." For example, a researcher might want a particular substring of a DNA sequence (preferably based upon a functional description rather than the linear numbering of positions).

(2) There is frequently subtle ambiguity in these requests. However, this is rarely troublesome because neither the databases nor the access software are sufficiency sophisticated to distinguish what might be biologically critical differences.

(3) Much of what a biologist wants could be described as content-based retrieval. Among the most frequent "queries" is, "What sequences are most similar to my newly inferred sequence?" At present there are a large number of programs for searching the entire content of a sequence database in order to answer this question. The critical element is the idea of "similar." There is no consensus on the correct or optimal definition, and the issue is actually decided on performance rather than scientific grounds. This query provides motivation for extensions in two directions. First, what I usually would like is to retrieve the actual sequences of the molecules, rather than just their names. Although this sounds simple, it is not; there will often be ambiguities in resolving this second step in the request (do I want the whole DNA entry, a specific "gene" from the entry, an inferred precursor protein, an inferred mature protein, etc.). The second extension generalizes finding similar items to other aspects of the database: data from similar organisms, from molecules with similar functions, or (among the most insidious problems) things with similar keywords or names. This requires one or more measures of similarity for each type of data in the database, and a mechanism for the user to easily modify the scope of any and all of these in performing a search (or "browsing" for data that might be of value or interest to them).

Scientific Database Workshop Position Paper

Jim Ostell
National Library of Medicine

Molecular biology is generating a host of data which requires specialized tools for analysis and management. Since molecular biology is an infant science, the data themselves are not well understood and their properties and relationships are constantly being revised. However, the flow of these data is expected to increase dramatically in the near future.

The effectiveness of a software tool in molecular biology is measured on two scales: 1) the number of scientists who find the tool useful in their work, and 2) the "correctness" of its algorithm. A great many tools have been created which are ineffective for one or both reasons.

The most widely used tools obviously are effective by criterion 1. Most of these tools were written by individuals who were personally involved in the type of work the tool is meant to do. Often such a tool is targeted at a particular hardware platform or operating environment. As such, it behaves naturally and intuitively both for its particular biological function and for the machine it operates on. Unfortunately, this approach often results in reinventing the same procedure for different machines, or locking scientists out of using a popular procedure if they do not have that machine. It also results in a proliferation of tools to do the same job, implemented in slightly different ways for slightly different applications. It often means that these programs do not use the best underlying algorithms for accomplishing their stated goals.

In the other case, a procedure implemented to solve a biological problem, which on theoretical grounds gives better answers, does not reach the end users because it only runs on a particular machine, or does not have a friendly interface, or does not use standard file formats, or was not written by someone in the business of software distribution. A sadder situation is when an algorithm and implementation are well done, but the biological problem was not well stated to the software designer. Many less than useful tools result when the software engineer is not the same person as the molecular biologist.

Almost all of the best tools currently available come from the private sector. They generally involve a molecular biologist who wrote software which evolved into a product. Some have resulted in small groups with many "cross-over" molecular biologist/software designer types. Generally they have targeted a single hardware platform which they know is available to laboratory workers. These groups are the most flexible at keeping up with the most current "good" algorithms, but the rate of change and growth puts a major development burden on such groups. As the range of possible tools increases, it becomes very unlikely that any single commercial package will contain all the tools a scientist might want, let alone the particular version of that tool that is the best current design. These packages are not usually "state of the art" in terms of software design or targeted hardware platform. Rather, their success stems from using hardware that "is really out there" and addressing the specific needs of particular scientists. Most of the best algorithms have come from an academic or government research group. Often the first versions of these are very limited in terms of user interface or data access. Those which have become widely available generally have done so by a second step of being incorporated into one (or many) more completely developed software packages. Many good algorithms have not been picked up by a broader package, and thus are used relatively little by the research community.

The problem now arises that the biological world has become too complex for a single small group to develop all possible tools to address. Further, if information from a large number of groups is to be integrated in large data acquisition efforts, standards must be set and maintained for how the data are handled. A group should not be using an algorithm for gel assembly which has been shown to make systematic errors when a better algorithm becomes available. However, if this group depends on a particular package, it may be a long time, if ever, before that routine is available to them. In addition, access to data will become more complex. Major biological databases are becoming very large. Relational schemas are being developed to provide access to the many interrelationships they encompass. Yet most groups will not be supporting the particular database manager a database group may be using. Further, they will need intelligent access tools, since they may not know or care about the underlying schema and/or that schema may be changing as knowledge grows.

The need for unity must be balanced by the historical fact that there are no models for success of a large software project in this area. The small entrepreneurs have provided invaluable tools to move this

science ahead, and cutting them out of the equation would be suicidal. Academics must be able to develop new algorithms without getting bogged down in complex user interfaces or elaborate data access tools, yet still easily reach the research community with useful tools. Computer scientists and database designers must now become involved in this area, but it is unlikely they will become expert biologists. The design needs and implementation evaluation must still flow from the biological community, but it is (very) unlikely that this community will become computer scientists. Success can only come from collaboration in terms of tool design, tool use, tool evaluation, data exchange, library functions, and review of quality at all levels, across disciplines, across physical distance, across hardware platforms, across administrative divisions (academic, government, commercial), and most important, across the changing landscape of biological knowledge over time.

Differences Between Commercial and Scientific Data

John L. Pfaltz
Department of Computer Science
University of Virginia

Traditional database management, as epitomized by relational database technology, is largely concerned with "commercial" or "business" data. By this we mean data that is well-structured according to a few accounting or organizational protocols. Many business transactions embody basic accounting procedures—recording amounts transferred, accounts debited and credited, date of transaction, transaction initiator, etc. Computerized order processing and shipping typically emulate manual procedures established by the corporation. Student registration follows procedures set up by the university's registrar. And so it goes. Relational database design is largely concerned with developing data organizations (or schema) which accurately mirror a set of configurations that can be generated by these fixed protocols.

I would argue that scientific data differs from more traditional commercial data in four significant ways:

(1) the potential amount of scientific data,
(2) the temporal value of scientific data.
(3) the way scientific data values are aggregated, and
(4) the meta-data needed to access scientific data.

First, even though commercial databases can be enormous, virtually all commercial data must ultimately be generated by a manual process. An order is received, a banking transaction is initiated at an automatic teller, a student registers for a set of classes. This effectively limits the rate at which commercial databases can grow. Scientific data, on the other hand can be mechanically generated by remote sensors or by automated experimental devices such as nuclear accelerators. To emphasize this, we note that the EOS (Earth Observing System) is expected to beam down in two weeks as much data as Landsat has generated during its lifetime. It seems reasonable to assert that comparable scientific data sets may be one, or two, orders of magnitude larger than their commercial counterparts. (This distinction may become blurred in the future by the automatic generation of commercial transactions, such as computerized trading on the stock market; but for now it seems valid.)

The second characteristic of scientific data is that it has no time window of "value". In commercial databases the value of most entries decreases rapidly after entry; in days, weeks, quarters, or years. Eventually all commercial data ceases to have commercial value; all can be legally thrown away after a fixed period of time, such as 7 years. This is not true if the data is perceived to have scientific significance. The clay cuneiform tablets recording Babylonian business transactions have no [commercial] value today; but recently a colleague used their astronomical data to confirm his analysis of the earth's rotational behavior. Very old scientific data can still have considerable value.

Entries in a commercial database typically involve a limited number of data values (by a data value we mean a single numeric or string value). In the relational model, these are first aggregated as a *tuple* of n-values, which typically represents a single transaction. A number of tuples are then aggregated as a set, or relation. There are effectively two levels of aggregation, the first to create a single object, instance, or tuple corresponding to a basic commercial entity; the second to aggregate similar entities. (This is an evident oversimplification. The relational join operator creates the possibility of different aggregations of data values. Still there is a very real sense of restriction.) Most scientific data sets have at least one more level of aggregation; our raw data values are typically vectors or arrays of numeric values. More importantly, if we visualize our basic scientific entities as being "experiments", we have situations where a single entity, instead of a fixed n-tuple of values, may very well comprise an indeterminate number of components, each of which may be an aggregate of values. It is very much more difficult to describe precisely what collection of data values constitutes a scientific entity.

Finally, the meta-data needed to access scientific data tends to be much more complex. I know of no really good definition for *meta-data.* I am using it in a loose sense to mean that associated data by which we understand the structure of a scientific entity and/or by which we locate entities of interest. In a sense meta-data describes the entity. For example, building on the preceding paragraph, a scientific experiment may consist of a structured collection of results together with a structured description of the context in

which it was conducted. To this might also be added a structured "interpretation" of the experiment. The relational model uses a schema (or simple list of attributes) to describe its entities; the kinds of hierarchical class structures found in semantic or object oriented database models seem more appropriate for scientific data.

It is doubtful if these four characteristics are, by themselves, sufficient to differentiate scientific from commercial data sets. But they do provide a basis for exploring how the management of such data sets must differ.

Assertion 1: The primary problem in mature scientific database will be the elimination of unwanted data.

Characteristics 1 and 2 contribute to what I call the *landfill model of scientific data.* Scientific data tends to accumulate forever, even though some 95% or more will never be used, or even examined, again. The sheer bulk of unwanted data makes the location of, and access to, valuable data problematical. To describe the comparable task of uncovering nuggets of truth in the bible, Thomas Jefferson once used the colorful phrase "searching for diamonds in a dunghill".

Possibly the most important step the scientific community could make would be to enunciate a policy of selective destruction—one which, over time, would minimize the size of stored data sets while maximizing the expected value of the retained data. Such a policy, no matter how wisely formulated, will result in the loss of data of some scientific value. But then again very little that is thrown into traditional landfills is without any value—as dedicated landfill pickers will attest. Nevertheless it is just more economic to bury it. Such a policy of data destruction will emerge; because it must. A rational policy would be preferable to one of sheer necessity.

Two approaches to the landfill problem might be considered by NSF. The first would be the formulation of a theory of the *expected value of data.* If such a theory could be developed, it might serve as the basis of a policy of selective data destruction. An alternative approach could be based on the assumption that the the creator of scientific data is in the best position to estimate its potential value. Individual and institutional contributors to a national database might be charged to initially insert data into the database (landfill) and then charged an annual service charge to keep it there. There would be no charge to mine the database—that would be actively encouraged. Yes — there would be problems with such an administrative policy; but it also has distinct possibilities!

Assertion 2: Most scientific data will be located by examining its meta-data, not the actual data.

Depending on ones sense of what is meant by "meta-data", this assertion might be true of all databases—commercial or scientific. Most often the primary key, say *employee#*, of a relational database is a kind of description of the entity, e.g. "this is data about employee 7312". Retrievals based on employee number are common. Queries, such as "all male employees over 50 earning less than $35,000 per year", that must examine the actual employee data are much less common, but many standard query languages, such as SQL, have features designed to accommodate them. Scientific data, on the other hand, is far less likely to be retrieved with respect to some unique identification assigned to the entity of interest. One envisions retrievals much more of the nature: "all satellite images of <geographic coordinates> taken in February, over the spectrum <spectral bounds>", or "the experiments referenced in <citation>", or "sequences of brain scan images for the same individual where at least two exhibit <condition>" (where this latter <condition> is based on secondary analysis and possibly added to the data set well after the original raw data).

The implications of this assertion, besides the obvious one that retrieval of scientific data is likely to require a much richer access path structure, are two-fold. First, retrieval software will need an enhanced capability of examining a "data dictionary" describing the structure of the data before examining any of the data sets themselves. Second, we will probably have to re-think the kinds of operators and/or comparators that can be applied in searching data. Contemporary data management systems are largely limited to simple equality and ordering operators (e.g. < and >); and the implementation of these in heterogeneous databases can be quite complex. Given the prevalence of aggregate values as basic storage items (e.g. arrays and images) do we allow conditional search clauses of the form "*where* det(<*data_name*>) > 1.0", or "*whose* std_dev(<*data_name*>) < 2.5", or "*where* <*image_name*> *contains* <*pattern*>"? Is allowance for such generalized search operators essential? If the answer is "yes", then the data retrieval systems will have to be extensible to accommodate new operators as their need becomes apparent. Will the added complexity payoff?

Assertion 3: In commercial databases, updates modify the real data; in scientific databases updates will primarily modify the meta-data.

When employee 7312 gets a raise, the value of the *salary* attribute is changed. But the schema of the representing tuple is not. Indeed, the schema of a relation cannot be changed without a massive file reformatting, a task that is not undertaken lightly. In contrast, except to change actual errors, scientific data should never change; it represents a record of observed phenomena which, once captured, should be static. But the interpretation of scientific data, the context in which it is understood, the links between it and other items of information, and the rules by which one manipulates it can change. Even the names by which we know components may change.

The data dictionaries of relational database systems are static. They can only be changed by a database administrator. Scientific data dictionaries and/or data catalogs are structures that must be dynamic.

Straightforward extensions of commercial (relational) database management techniques are unlikely to suffice in the world of science. A different approach to scientific data must first be enunciated and then implemented.

Types of Scientific Databases

Robert J. Robbins
Division of Instrumentation and Resources
National Science Foundation

Some confusion in discussions of scientific databases seems to derive from different individuals having different implicit notions of just what constitutes a scientific database. For this reason, it may be useful to attempt some categorizations of scientific databases and then to examine each type for its individual attributes, management needs, etc.

Classifications may be based upon a number of different (but not necessarily independent) criteria. For example, scientific databases may be characterized by their users, by the type of subject matter (e.g., raw data vs information), by their relationship to scientific literature, or by their role in the scientific process. In addition, some integration of these different classifications may be achieved by a more general analysis of the function performed by the database.

User Classifications:

At the broadest level, scientific databases may be classified according to the breadth of their users as being local or public. Local databases are used to maintain data needed by an individual laboratory to support the work of that laboratory, whereas public databases support an arbitrarily wide set of users. In actuality, of course, these categories represent the end points of a continuum, since intermediate types, supporting the work of a collection of collaborating laboratories can easily be envisioned.

The differences in requirements and attributes for local and public databases bear similarity to the differences between locally produced pieces of laboratory equipment (designed and built by laboratory staff) and manufactured equipment (designed and built by an equipment vendor).

Locally produced lab equipment is often idiosyncratic, cranky, and almost impossible to use by those not intimately familiar with its inner workings. Locally produced lab equipment never comes with a user's manual, and only rarely is associated with the slightest bit of documentation regarding its design. Locally produced lab equipment is usually built to meet an immediate and pressing need, at a time when getting something running is top priority. Conversely, the need to document the design or to plan for maintenance is of low priority. Despite its problems, locally produced lab equipment can be remarkably cost-effective in the laboratory in which it was produced.

Manufactured equipment, especially high quality manufactured equipment, is more predictable, usually easier to use, and certainly better documented. Having been designed to serve many users, manufactured equipment may be less well suited to the precise needs of the local task than locally produced equipment.

If no manufactured equipment exists that meets a complicated need, locally produced equipment can almost always be put together to meet the most pressing needs more quickly than manufactured equipment can be designed, built, tested, produced, ordered, shipped, delivered, and installed. However, locally produced equipment can be notoriously difficult to repair or maintain, especially if the original builder is no longer available to perform the work.

When immediate availability and precise attention to local needs are coupled to a relatively short expected useful life by a limited and highly motivated set of users, locally produced equipment can be far superior to manufactured gear. However, when these conditions are not met, manufactured equipment usually proves superior in the long run.

Local vs public databases show many of these requirements and attributes arrayed along the same continuum.

Subject Matter Classifications: (by levels of analysis)

In one way or another, scientific databases can be described according the level of analysis that their subject matter has undergone prior to entry in the database. Points along this continuum are often described as data, information, and knowledge. Since the words ''data'' through ''knowledge'' are often the source of debate, the continuum can also be described as ranging from unprocessed observations through facts to integrated conclusions.

Granted, one person's information is another's data. However, when attention is restricted to a single point of view, it is usually possible to array database contents along this continuum. In gene mapping using tetrad analysis, for example, the classification of spores from a single meiosis would be data, the calculation of a recombinant frequency between two loci would be information, and the placement of these loci on a map would be knowledge.

Subject Matter Classifications: (by matter of reference)

The contents of scientific databases can also be categorized according to whether or not they are conceived as recording either scientific observations or facts about the real world. In a database of scientific observations, no inconsistencies result from the simultaneous presence of Jones' observation that asbestos causes cancer and Smith's observation that asbestos produces no detectable health effects. However, in a database that purports to contains facts about the real world, these two assertions would represent an inconsistency in the database.

Given that the same observer can make different observations at different times or places or with different methods, we see that records (tuples, whatever) in databases of scientific observations will have a certain minimal structure: <observation, observer, time, place, method>. From one point of view, the observation component might be considered the scientific datum and the other components considered metadata about the scientific datum. With that terminology, we see that two records would be contradictory only if they contain identical metadata, but contradictory data.

Databases of observations differ significantly from those of true facts, particularly in their managerial and integrity checking requirements. To be truly valid, databases of facts about the real world must contain no contradictory assertions either directly in the database as records or indirectly in the database as deducible conclusions. Thus, high quality real-world-fact databases will require a very high overhead of integrity checking. If it were not possible to automate the integrity checking, a significant management work load, that could be performed only by experts in the subject matter of the database, would be required.

In a database of observations, on the other hand, no new observation could ever contradict an older observation in the sense of requiring that one or the other of the observations be disallowed for entry. Furthermore, databases of observations have no integrity requirement that the recorded observations should not be permitted to allow the inference of incorrect statements.

As examples of these types of databases, consider first a database of observations of reaction constants for different chemical reactions. To be valid, such a database would need to contain only accurate records of actual observations, along with accurate records of the appropriate metadata. Once one of these records were entered and checked for accuracy, no further maintenance of that record would be required. As new observations were reported, they would simply be entered into the database, resulting in multiple records for a given reaction. Queries regarding a particular reaction would retrieve all relevant records and the final interpretation of the records would be left to the user.

In contrast, consider a database that claimed to contain the actual real-world reaction constants for different chemical reactions. This database would contain only one record per reaction. As new observations were reported, they would have to be carefully analyzed to see if they should cause the one ''true'' value to be updated. If an update seemed appropriate, consistency with all other records in the database would have to be checked to insure that no impossible situations (such as a chain reactions yielding perpetual energy output) were implied.

Literature Relationship Classifications:

Databases of processed data (i.e., databases of information or knowledge) may be restricted so that they contain only assertions made in the refereed literature or they may be unrestricted and permit the recording of assertions not reported in the literature.

Unrestricted databases have many advantages (speed and timeliness of entry, etc.) but they also incur many costs. If they permit totally unrefereed entries they risk become mere bulletin boards of assertions with uncertain value. If they require their own referee system, they incur a significant management burden.

Functional Classifications:

Experiment management:

These systems are used to store and manage experimental data for immediate analysis and for planning additional experiments. These systems are generally used (often in real time mode) by the researchers who generated the data. ''Lab-notebook'' systems intended to assist in the day-by-day management of data in physical mapping efforts provide a good example.

Because experimental protocols may change significantly and rapidly, and because these systems are used dynamically to plan and interpret on-going experiments, these systems are characterized by a need for flexibility and timeliness. It must be possible to modify these systems quickly without having to resort to system redesign, coding, and compilation. And, it must be possible for users to enter, manipulate, extract, and analyze data quickly without the intermediation of systems analysts, programmers, or other computer gurus. In short, these should be ''spreadsheet-like'' data management systems that provide a wealth of tools to the user in an easy-to-modify, easy-to-manage format.

Since the primary users of these systems should have ready access to the original collectors of the data (indeed, they are likely to be the original collectors), detailed data documentation is likely to be of relatively low priority.

These systems are likely to have little or no need to link data elements to literature citations.

Data archive and retrieval:

These systems are used to store, manage, and retrieve (relatively raw) data for further analysis. In general, these systems manage data for long-term use by many researchers other than those who originally collected the data. Good examples would be the storage and management of telemetry data from space probes or long-term ecological data sets.

The management of such collections of scientific data bears much similarity to the management of scientific specimens. In both cases, there are collections of objects that may be used by scientists in further analysis. And, in both cases, the collections are virtually worthless unless they are accompanied by enough documentation to allow interested scientists to discover that potentially relevant objects exist and to verify that some of the objects are in fact appropriate and reliable subjects for study. In addition, there must be some mechanism to allow scientists to retrieve the objects conveniently and in a form suitable for study.

Identifying and verifying relevant objects can only be done via a good metadata system. Since the museum community has for years been managing scientific objects for further study, much can be learned about metadata needs by observing the highly evolved collection-management systems used in well run, highly respected museums.

Because these archived data sets are intended for long-term use by workers with no opportunity for contact with the original collectors of the data, the existence of a reliable, well-maintained data documentation system is absolutely essential for the usefulness of the system. These systems should have a moderate need to link data elements to literature citations. In some cases, linkage to citations might be used to provide additional documentation regarding the collection and other attributes of the data. In other cases, it might be useful to offer a citation-index function that could be used to retrieve literature in which the data have been used.

Literature retrieval:

Literature retrieval systems are concerned with providing access to the scientific literature. They have more in common with information retrieval systems than they do with database systems. Medline or BIOSIS provide the obvious examples.

Such systems do not pretend to provide information not present in the literature, nor do they pretend to offer significant value-added interpretation of the literature. Their primary (only?) goal is to assist the researcher in locating POTENTIALLY relevant literature. The final decision of the located literature's ACTUAL relevance remains with the user.

In these systems, literature citations ARE the data elements.

Fact retrieval (literature-based):

Literature-based fact retrieval systems offer a significant value-added interpretational feature. In these systems, expert readers comb the literature to extract ''facts'' that are placed into a (reasonably) structured database management system. GenBank, EMBL, etc., provide examples.

Fact retrieval (non-literature-based):

Non-literature-based fact-retrieval systems differ only in that they permit the entry of facts that have not appeared elsewhere in the refereed literature. At issue with these systems is the degree to which the managers of the systems must supply a quality control system that is the equivalent of the peer review system present in refereed literature.

At one extreme, Flora North America (a database project with the goal of producing a flora for all plant species occurring in North America) provides an example of a non-literature-based system that has been organized to mimic the management structure for written literature. ''Authors'' provide entries into the database, which are passed through hierarchically structured editorial (i.e., review) committees before being entered into the actual database. Authorship of entries will be maintained and provided to reviewers, and the identity and roles of the editors will also be made publically available.

At the other extreme, the Authorin system for GenBank allows the immediate entry of non-literature-based, non-reviewed data into a database system.

NSF Workshop on Scientific Databases

Peter Shames
Space Telescope Science Institute

In providing truly useful science data systems, challenges seem to arise in two disjoint, but related, categories: sociological issues and technological issues. We are closer to having solutions to the technological issues than we are to having solutions for the sociological ones of user expectation and modus operandi. However, the success of any systems we produce will depend directly on how well we meet the users' present needs and guide their expectations for the future. Technologically, the challenge is to build an infrastructure that will permit diversity to flourish, the system to evolve gracefully, and user needs to be (and continue to be) fully met.

What is the functional equivalent of walking into the office of the local expert, colleague, or research librarian and asking how to find all there is to know about Q? Much research seems to be conducted by seeking help from the experts, following reference chains, locating the database that someone created for their own purposes, or personal contacts and discussions. This forms some sort of network of information, but in actual practice this process may be more closely analogous to wandering through a forest where there are few well marked trails. The problem of locating and cross-referencing data is well known and tools such as Xanadu, Hypertext, and others have tried to provide suitable constructs to assist the exploration of this complex data space. However, the problem is still with us.

Sociological Issues: Information exchange / finding the data / trusting the data

If locating, accessing, and correlating data is the real problem, and it is as intractable as it still appears to be, what kinds of systems might we build and what portion of the problem space can we hope to deal with? We know that we can construct systems in which all of the data from a given project or experiment can be archived, catalogued, described, and made accessible to users. Such systems, at a variety of levels, have already been developed. Various library systems, abstract searching systems, and systems that correlate research papers with research topics or physical objects are now in use. Several activities are now defining systems that will integrate these project data systems into a discipline data system, offering the possibility of spectral, temporal, and spatial analysis of different physical phenomena.

The integration of all of these existing and planned systems (in even a single discipline) appears to just now be coming into reach, but a number of problems remain. Ensuring the quality of the data that are entered, determining the best categorization of data or documents and seeing that their content is accurately indexed, and identifying cross-references and linkages among data are still very labor intensive and are, as a result, difficult to trust. Solutions may be at hand for some of these problems, but they have yet to be rigorously applied to a large-scale system. Users seeking rigor in their research may still prefer to do the reference work in the old way, following the paper trail. From a user perspective the systems are only as good as the quality of the data that they contain.

Sitting at the boundary between the users and the technology are the interfaces themselves. Interfaces that require carefully constructed queries using some arcane syntax and specific keywords are doomed to failure in any open user context. Some form of natural interface (natural language, pointing, verbal) that permits expression of the problem in the user's own terms is needed. Providing common interfaces that support the exploration of all elements in the data space and tools for displaying the text, data, documents, and images that are returned is essential.

The final issues on the sociology side are those of accessing, operating, and managing the system. Who will be able to use it, how do they get to it, who guarantees the data integrity, how is it kept up to date, who runs it, who pays for it? Many agencies (NSF, NASA, NOAA, USGS, DOE) generate data; some data come from private research or labs, and some from commercial entities. How can this varied input be integrated into a system that functions as smoothly as today's library exchanges do? How do we ensure that such a system is a truly national scientific resource?

Technological Issues: Information interchange / integration of diverse datasets

Among the technological issues, the most obvious are those of information interchange and data standards. There are now many fields where exchange of information is possible, but only if one is willing

to write programs to perform the necessary data restructuring. These may be simple format conversions or more complex re-mappings into different coordinate spaces. The data sets may be sufficiently self-descriptive that this is possible, or the transform may rely upon some human intervention or set of conventions to guide the process.

There are some notable successes in the area of data interchange, the FITS standard in astronomy being one case in point. FITS is used by all major astronomy software packages as the standard format for data transport and ingest. It has been used to transport data taken with single and multiple element radio telescopes, ground-based optical telescopes, and space-based UV and X-ray telescopes. It supports image cubes, spectral line data, and various tabular or database constructs. It is self-descriptive of the data contents so that many transforms on different data types can be performed automatically. FITS has drawbacks in that extensions are permitted but not all standardized, and there are those who complain of its complexity, but it does work.

Designers of some systems are exploring the interchange of descriptive data about catalogs, metadata about inventories, and other information about sites, facilities, and services. However, common standards for data description and transport do not yet exist except in some limited discipline or system contexts. Standards coming from ISO (8211, 8824) and the space community (CCSDS, SFDU) show promise, but are still evolving, have yet to be used in large-scale systems, and are not yet widely accepted. These are critical for data transport and interoperability; they are essential if interoperable systems are to be realized.

Challenge: Heterogeneous, evolvable infrastructure / making it happen

Some distributed systems have succeeded because they are based upon a single vendor hardware platform and network technology. Other systems have supported mixed vendor hardware, but with all components running the same operating system. However, there are few examples of systems that map into the truly heterogeneous world that we all work in. Diversity is a fact of life and technology is continually evolving, to ensure that the status quo is anything but static. Different operating systems, different network protocols, different hardware data formats and interfaces are a fact of life.

If it is not a requirement that different systems interoperate, if users are satisfied with learning the interfaces of each new system they wish to use, and if data transport via bus (as in large, yellow) is acceptable, then we need not concern ourselves with system infrastructure. If interoperability of systems is a part of the strategy of providing powerful, convenient systems that can enhance scientific productivity, then we must be concerned with system architecture and infrastructure.

This infrastructure is not just networks, it is also the protocols for naming systems, locating them, establishing connections, authenticating users and services, accessing the databases and archives, browsing the data, invoking compute servers, and getting the results back to users. It is the mechanisms for handling system heterogeneity, for transporting information over whatever network protocol happens to be suitable, and for gracefully evolving the system with new hardware, services, and facilities during the coming years. And it is the portable tools, utilities, and user interfaces that link to these services and provide users with the ability to manipulate their local view of the environment in the solution of their problems.

Creating this infrastructure is the single greatest challenge our technology must meet if large scale distributed systems are to become a reality. This can be thought of as a system buss into which different modules are plugged, not unlike the backplane buss of a computer. The buss provides the system backbone: the data exchange, the system interconnects, the data transport. It allows individual data systems to be developed with whatever hardware, software, or protocols are appropriate as long as they conform to the interface protocols at the buss. The NREN is to provide the network structures needed. Software building blocks for such distributed systems do exist (CRONUS, ANSA, Kerberos), but work is needed to turn them into a robust infrastructure for the sort of systems being discussed. Once this challenge is met we can start dealing with the sociological issues.

On the Importance of Metadata Management for Scientific Applications

Arie Shoshani
Information and Computing Sciences Division
Lawrence Berkeley Laboratory

1. Introduction

As is well accepted by now, the term metadata refers to information about data. In general, it is the information about the content and meaning of the database. In scientific applications, this information can be quite complex, and are non-trivial to organize. Often data become obsolete because the information about its content and meaning is non-existent or lost.

What distinguishes metadata from data? One point of view is that the distinction is arbitrary. What is metadata for one person is considered as data by another. Consider, for example, the data collected by seismic devices. Is the information on the type of devices, the unit of measure, etc. metadata or data? Another point of view prefers a functional definition, which says that metadata is the information that should be available to users in order to be able to issue queries against the data.

It is an open question whether metadata should be treated with the same tools that manage data. It is certainly an elegant approach that is accepted in current relational technology. The question is whether metadata have unique semantics and unique operations for their manipulation to justify special purpose software.

Rather than answer the above question directly, we will attempt to point out below some of the features that metadata should have in a scientific environment. We start with a brief example of a typical scientific environment that exists in epidemiological studies of low level radiation effects on human beings.

2. An example

The original sources of data for studies of low level radiation effects are employment files, external radiation monitoring files (e.g. radiation sensitive devices on badges), internal radiation monitoring (e.g. urinalysis), air samples for radioactive materials, job assignments, morbidity and mortality data, etc. These data are collected in different ways, such as automatic recording devices, data entry forms, hand written notes, etc.

Before any analysis can proceed, these original data need to be put into computer readable files (called "raw data" files). Because mistakes may be introduced during this step (e.g. interpreting hand-written notes) it is necessary to capture the relationship to the original files and how the data values were determined. The next step is to select a subset of the workers for a study (called cohort). For the cohort selected the raw data files will be "linked" (that is identify all the records that relate to the same individual). This is when errors such as two different Employee Numbers (or Social Security Numbers) are found and corrected. Other inconsistencies (such as values that seem to be too large or too small, and various correlations between data values) are also checked. Finally, a "clean" set of files is created.

The next step is to apply models to the raw data in order to obtain "derived" data of interest. For example, various models are used for determining the actual internal dose from urinalysis, air samples, the disposal capability of the human body, etc. This produces an "analysis file" for the study. Analysis files can further be checked for inconsistencies or modelled further to produce new analysis files.

3. Observations

The above example demonstrates a process that is typical to many scientific applications and involves the following steps: data collection, data validation, data correction, and data derivation. Each of these steps has metadata associated with it, as described below.

3.1. Data collection

This is the process that generates the "raw data". The metadata includes information about the objects (entities) of the database and their relationships. It also includes information about the data attributes including acronyms used, text description of their meaning, units of measure, format for data values, allowable data values (ranges or lists of permissible values), codes used for encoding the data values,

grouping (e.g. age groups), and meaning for exception values, such as nulls). In addition, there is information about the devices used to collect the information. In the above example, the dosimeters used vary over time and from location to location, and usually have different characteristics. Finally, there is information about the data sets, such as who produced them, when and where they were produced, and a text writeup about their content.

3.2. Data Validation

The metadata associated with this process are the conditions and rules that apply in order to validate the data. These integrity conditions may be quite complex, and in general cannot be expressed with existing query languages.

3.3. Data correction

This process is often a continuous one; as new information is found about individuals, data values are corrected. The metadata involved are the history of corrections made to the data, the reasons for such corrections, who made the correction, and when it was made. Often, such data is kept in log books (which may not be in computerized form.)

3.4. Data derivation

The process of data derivation may be as simple as summing up values (e.g. total dose per year is the sum over monthly doses), or as complex as applying a model to calculate derived values (as discussed in the example above). The metadata involved are the statements or programs that are used to generate the derived values. In addition, there is a need to keep track of versions of derived data.

4. Discussion

We discuss briefly below some of the implications concerning the support of metadata. For many of the above aspects of metadata, one can find partial solutions with current technology. For example, in order to support codes, relations (tables) can be defined using relational systems. However, it is up to external software (or the user) to interpret these tables. If instead, there was a data model that supports the notion of a code definition, then a "browsing" capability could automatically display the meaning of codes (e.g. "lung cancer" instead of the code used for it).

Another interesting example exists in expressing and supporting integrity conditions. Simple conditions are supported by commercially available systems, but more complex conditions are typically implemented by special purpose programs. However, there is room to investigate the usefulness of rule based systems for this purpose.

The support of derived data is another important example. The simple cases can be supported by "view" mechanisms of current relational systems. However, there should also be support for arbitrary programs to derive data. Also, the support for version management requires special data structures and operators.

Finally, it is worth noting that text descriptions are used in the different steps mentioned above. This suggests the need to be able to search text. The more precise approach is to organize the text content into categories of information and keywords, but this is not always possible or practical.

Metadata is essential for integration of multiple data sets for further analysis. Integration of existing data is one of the most difficult aspects of scientific databases. Agreed upon standards is the best solution to facilitate data integrity, but are often difficult to achieve. Well organized metadata can at least provide the information necessary for data integration.

Scientific Data Management Issues in the Earth Sciences

Ferris Webster
College of Marine Studies
University of Delaware

The Global Change Research Program is intended to establish an integrated, comprehensive long-term program of documenting the Earth system on a global scale. It proposes a program of studies to improve our understanding of the physical, geological, chemical, biological, and social processes that influence Earth system processes and trends on global and regional scales. The ultimate objective is to develop conceptual and predictive Earth system models.

Data and information management will be a key factor in the success of global change research. It has been recognized in planning as an essential bridge between global change observations and scientific understanding. In planning for a global change data and information management system, a number of data management issues have been identified.

I. Scientific issues (Scientific use of databases)

a) Priorities for data and information management should be driven by and accountable to the scientific program objectives.

b) The preparation of products and analyses should be an integral part of the system. Information, derived from the data, rather than data themselves, will be needed for decision making. What should be the relation between a database and the derived information base?

c) How can the databases needed to answer global change scientific questions be built? The system must accept and archive dissimilar types of data collected from different data collection system, from ground- and space- based collection systems, in different formats and on different media.

II. Technical issues (Computer science theory and application)

a) Meta-databases are critical. There are a legion of examples of datasets that were unusable because appropriate information about the conditions under which the data were obtained were not retained. It is possible that metadata can be treated using many of the same procedures used for databases.

b) The system must be extensible to provide for an evolving data management system and an expanding user base.

c) The system should ensure that a scientist, decades hence, can use the datasets effectively.

d) The system must allow a scientist to find existing data and how good the data are. This will require some kind of global high-level database directory.

III. Sociological issues (The role of databases in science)

a) Unfortunately for the development of effective scientific participation in the development and operation of a global change data and information management system, data management is often viewed as a second-class profession. This workshop probably cannot address this issue directly, but might help alleviate the perception indirectly through improving the scientific basis for database management.

b) Related to the previous point, data and information management must be adequately supported. The existing national data centers are generally not adequately supported to service present users, much less the expanded user community and data needs the global program will involve.

c) In spite of support problems, the existing system in the earth sciences, of national and world data centers and academic and project data units, provides a useful starting point for a distributed data and information management system.

d) For a world-wide cooperative scientific program there must be the ethic that datasets are shared. Databases must be part of the scientific program, and perhaps there must be suitable career rewards for scientific database service providers.

Workshop on Scientific Data Bases

Gregory W. Withee
National Oceanic and Atmospheric Administration

What is available and where is it?

One of the first questions a scientist asks when formulating a hypothesis, is what data are available to confirm it, or contradict it. In the environmental sciences this search can be long and frustrating. A movement is underway to build environmental data directories that answers the question what data are available and where, for some disciplines. These efforts ought to be encouraged and funded as separate high-priority projects instead of "out-of-hide" efforts.

Data services, like library services, start with a catalog system. Once these directory systems are built we need to invest in catalog and inventory systems. They do not need to be connected at first, but they need to be on line so that a scientist can navigate through an inventory list, and even browse through real data. Unlimited data navigation is not available at this time, but technologies for the ingredients are.

Now that I've found the data, can I get it, and will it be what I want?

Data that are not accessible are worthless. Data not documented as to quality, validation technique, principal investigator, instrument and so on, make it difficult for the scientist to use the data even when she or he gets it. Documentation also allows the scientist in advance to determine more accurately the usefulness of the data.

An additional access area is media. It is remarkable how many individual scientists have PCs and workstations, but how few have tape readers in their offices. Tapes, and centralized machines are still necessary for many applications, but data can be captured for the scientists workstation, for example, on CD-ROM.

CD-ROM technology will no doubt be with us for 5 to 10 years. One investment that could improve science access to data is to purchase a CD-ROM reader for all scientists, and, at the same time, master one set after another, on a priority basis for science use. At $600/reader and $2/disk, this cost could be less that $1,000,000 for 1000 scientists. Putting data sets on CD-ROM is also not expensive, in terms of cost per byte. For example, all oceanographic data residing at the National Oceanographic Data Center could be mastered for less than $1 million including some quality and documentation improvements.

Another aspect of access is migration: from paper to tape, from tape to disk, etc. Many data will be lost to further generations if this migration does not occur. Agencies should fund a certain percentage of data migration each year.

Universities, companies, and Federal centers are all working on these problems. Perhaps it is time to accelerate the process by forming a joint Federal/Industry center/partnership dedicated to making accessible the data universe.

Interagency, International, Interdisciplinary

These three "I" words make life in the data management lane quite challenging. These days more and more scientists are tackling problems that can be solved only by using data from other scientists or other agencies. We need to form better data bridges or gateways between agencies. NSF cannot take on these data management challenges by itself. We have in interagency working group on data management for global change; but that is not enough.

Scientific problems requiring international data require even more coordination. An alarming trend in the world today is the increasing tendency for countries to treat data as a commodity, one to be bought and sold. Data fees pose a threat to scientific programs. More and more I hear scientists not obtaining data because they have no money. Amazingly enough even a few thousand dollars can be formidable. Block funding arrangements between agencies and data centers can help alleviate this problem. An international convention on data exchange, signed by all countries, could go a long way toward preserving research access to data.

Interdisciplinary aspects of new research, such as biogeochemistry, and marine ecosystems, pose new challenges to the data system. Scientists trying to use data sets from one discipline and apply them to another

are further removed from their understanding of data. Documentation is of utmost importance to these interdisciplinary scientists.

Common Fallacies

Data management systems are failing because of the increase in data volume.

In general large data volumes produce digital streams with known characteristics. Data handling practices in concept are not more difficult to understand than those for smaller data sets. On the other hand, data sets assembled from thousands of data sources, and with different sensors, algorithms, etc., can pose more challenging problems.

Scientists are good data managers.

In general, scientists are not good data managers. Scientists have not historically done a lot of data management, they look to it as a chore. Data managers, on the other hand, historically have difficulty in keeping up with the science they support; they find themselves being quite good in data management but often working on the wrong problems and unable as a result to help scientists. Certainly the most successful data management centers have involved data managers working with scientists. Indeed many successful science efforts have a solid data management program integrated within them.

The lesson is that agencies should fund data management as an integral part of science programs, and expect a good result. Federal agencies that have National data centers should fund research efforts as a part of these centers' functions.

Data Management as a Profession

It takes a lot talent to build a building. Someone has to perceive a need for it; an architect has to design it; and a contractor and many subcontractors built it, etc. A parallel to scientific data management can be drawn. A scientist may have need for a data base to be constructed. He or she calls upon a data base architect to design it, and contracts with a data engineering firm to build it. Data technicians work on subassembly according to a specific instructions.

In most scientific disciplines this scenario does not happen. More often than not the scientist cannot find a data architect, and most of the time does not recognize they exist. In fact, as a recognized profession they do not. We have computer societies but we do not have a society of data architects and engineers. Present data architects have various university degrees, from physics to computer science, but have learned data management by on the job experience.

NSF could invest in a few universities to develop a data management education program. As a result a new profession could emerge. For example, a future student could major in data architecture with a minor in physics . . . Courses would include data navigation. As a practical matter the Federal, university, industry data navigation laboratory cited above could be located at such a university.

General

Observations in the form of scientific data of this planet and others have been, and will continue to be fundamental to our ability to understand and predict the earth system. However, insufficient funds are available to manage this precious resource.

It should be a requirement that agencies set aside a percentage (e.g., 25%) of their observation budget for data management. We would not think of building a car without wheels; why do we fund observations without proper data handling, quality control, storage, and access facilities.

Biological Databases, Software Development and Computational Biology

John C. Wooley
Instrumentation and Resources Division
Biological, Behavioral and Social Sciences
National Science Foundation

The Instrumentation and Resources Division of NSF has created a Program with the above title to catalyze the development of computational biology, including the development of new information resources for biologists in the short term, and support for the development of new data and information management tools for scientific information, over the longer term. The following summary describes the philosophy behind our establishment of this Program and our participation, in some cases in collaboration with the Information, Robotics, and Intelligent Systems Division, in the support of research on scientific databases, as well as the development of new public databases for the biological, behavioral and social sciences.

Computational Biology is emerging as a discipline in its own right. This is analogous to a revolution in the biological sciences in the late 1950s and early 1960s. During this period a group of scientists began to analyze biological problems in a new way by applying the tools of several disciplines; namely, genetics, microbiology, physics, biochemistry, and biophysics. The power of this approach was so great that it emerged as a discipline itself and is now known as molecular biology. Today, the application of mathematical and computational tools to all areas of biology is producing equally exciting results, providing insights into problems in the life sciences too complex for traditional analysis. To develop most productivity, this new field needs unique resources of information and computation, and support for cross disciplinary training for students and established investigators who are moving into this field.

There is a consensus among all observers that biology, regardless of the subspeciality, is overwhelmed with a large amount of very complex data. What sets biology apart from other data rich fields is the *complexity* rather than the sheer volume of the data produced. However, unlike some data rich fields, biology remains a scientific ‘‘cottage industry,’’ with the data generation done in a highly distributed mode, with no standard format or syntax.

Thus, all areas of the biological sciences have urgent needs for the organized and accessible storage of biological data. Generally this is referred to as biological database development; however, this terminology implies traditional database technology such as transaction oriented relational database systems, and given the common notion of what is meant by a database, fails to convey the difficulties and importance of the challenge. Current database technology is inadequate to serve many areas of the biological sciences; collaborations among computer scientists and biologists will be necessary to design information platforms that accommodate the needs for variation in the representation of biological data, the distributed nature of the data acquisition system, the variable demands placed on different data sets, and the absence of adequate algorithms for data comparison, which forms the basis of biological science.

Scientific Database Management and Conduct of Research

Maria Zemankova
Database and Expert Systems Program
Division of Information, Robotics and Intelligent Systems
National Science Foundation

Database management technology has progressed rapidly in recent years. We have now commercially viable database management systems that provide mechanisms for storage and retrieval of large quantities of data on anything from microcomputers, to mainframes, to specialized database machines. We often hear that we have entered the "information age". However, I classify the current era as "data age". We are collecting staggering volumes of data, but do not have adequate methods for effectively using the data and producing useful information. This lack of suitable information technology is most serious in less traditional scientific application areas where collections of very large volumes of numerical data, image data, geographic data, genome data, and other unstructured data are used in often unforeseen ways. Information technology is changing the conduct of research, and we are faced with challenges of providing the most adequate information systems that enable the scientists to derive the maximum benefit from the available data.

Information technology can offer improvements in the capability to store, access, and analyze much larger volumes of multi-media data, along with the capability to present results in visual form. Research in information modeling can provide necessary structures and operations for storage and manipulation of untraditional data that cannot be represented in the current predominantly relational model based database management systems. Application of artificial intelligence techniques may provide useful approaches to analysis, summarization, and high-level (metadata) descriptions of large volumes of data, especially when trends over long periods of time are necessary. Information and telecommunication technologies can provide the means for more open and efficient collaboration.

In order to build useful scientific database management systems, it is necessary that the database research community understand what the information needs of domain scientists are. I believe that in this workshop we will be able to examine these needs, determine how the current state-of-the art of the information technology can be applied to satisfying these needs and where the technology needs to branch out into new directions. It is clear that providing adequate network of inter-disciplinary scientific information systems will require collaboration of scientists from many disciplines, support from agencies currently supporting research, and involvement of industry in whose domain it is to build widely available systems. This workshop's participants intentionally represent these spheres and it is my hope we can identify the challenges and research directions that will lead to production of effective scientific information systems.