DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PNL-SA-26092

THE ROLE OF METADATA IN MANAGING LARGE ENVIRONMENTAL SCIENCE DATASETS

Proceedings of SDM-92
A Planning Workshop
November 3-5, 1992
Salt Lake City, Utah

June 1995

Edited by:
Ronald B. Melton and D. Michael DeVaney
Pacific Northwest Laboratory
and
James C. French
University of Virginia

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Abstract

The purpose of this workshop was to bring together computer science researchers and environmental sciences data management practitioners to consider the role of metadata in managing large environmental sciences datasets. The objectives included:

- establishing a common definition of metadata,
- identifying categories of metadata,
- defining problems in managing metadata, and
- defining problems related to linking metadata with primary data.

Acknowledgements

We would like to acknowledge Judy Cushing, who took notes at the workshop, and Stan Moon, who served as the discussion facilitator. We would also like to thank Dave Maier for his help in planning and organizing the workshop.

Funding to support the workshop was provided by the U.S. Department of Energy's Atmospheric Radiation Measurement program and the U.S. Environmental Protection Agency's Environmental Monitoring And Assessment Program.

Table of Contents

Introduction

These proceedings are a result of SDM-92, a three day work shop held in November 1992 to discuss metadata issues that arise in the management of large environmental datasets. Metadata, i.e. data about data, is an important element of scientific data sets. Traditionally metadata might be thought of as the notes in the lab notebook that accompany the record of the primary data, i.e., the data values observed by the scientist. The automated collection or generation of large datasets, i.e., gigabytes/day and up, requires formal means of capturing and managing metadata. Without integrated metadata these large datasets are difficult to use because scientists cannot establish the context within which the data were collected, generated or analyzed. It is the metadata that establishes that context. Metadata is especially important when the user does not collect the data personally, as is often the case with large science projects, and when one user is sharing another's data.

Current activities and practices in the environmental sciences provide specific examples of the importance of metadata. In atmospheric science, for example, modern field programs are typically large, multi-discipline, multi-agency, and multi-investigator efforts with land-, sea-, air- and space- based instrumentation operating under a wide range of conditions. Data which document the platform and instrument conditions, the operational environment and interfering sources of noise are critical to primary data analysis. These data are all metadata and underscore the importance of developing standard approaches to recording and handling metadata as critical data in its own right, and in linking metadata with the primary data.

The metadata also play an important role in integration of datasets from programmatically unrelated activities. The contextual information captured in the metadata provide the basis for the integration. Likewise, the useful life of a dataset is extended if the primary data are well documented with relevant metadata. In both situations appropriate inclusion and management of metadata provides an economic benefit by making the primary data more useful.

At the time of this workshop the literature contained little discussion of metadata in the context of scientific data and its management. Metadata were primarily discussed as information about the structure of a database (denotative metadata), rather than as information about the contents of the database (annotative metadata). This situation is also reflected in the tools available for dealing with metadata. They are primarily focussed on helping with management of denotative metadata, e.g., data dictionary tools.

Data management practitioners have addressed scientific metadata problems in an ad hoc manner as they have solved specific problems. Some common data formats developed for distributing or exchanging scientific data, for example the Network Common Data Format - netCDF, are self-describing and include facilities for storage of and access to metadata. There are few, if any, tools that have been developed that allow users to access and analyze metadata.

In general, there has not been any work done to establish a "theory" of metadata or a body of generic tools. One purpose of this workshop was to raise the level of awareness of the problem of managing scientific data and metadata within the computer science research community. Our hope is to stimulate the Computer Science research community to examine problems of dealing with scientific metadata and its integration with the primary data. New approaches to managing large scientific databases with integrated primary data and metadata will enablethe scientists to make more effective use of the data.

The workshop was organized around three elements: formal presentations, focussed discussions in small working groups, and large group discussions of the working group results. These elements were applied to three topics:

- Characterization of the environmental sciences data management "problem",
- Identification of metadata standards, terms and terminology, and
- Tools and requirements for managing metadata.

Prior to the working group discussion of each topic, the entire group reviewed and added to a list of questions relevant to the topic. Each working group leader then chose one or more questions for their group to discuss.

This proceedings include the position papers prepared by workshop participants. These papers are organized to roughly match the three topics that were used to focus the workshop. Finally several appendices are included that present related material such as a glossary and list of workshop participants.

What Is Metadata?

James C. French
Department of Computer Science
University of Virginia
Charlottesville, VA 22903

1. Introduction

The notion of metadata in scientific datasets has always had an intuitive feel about it. Practitioners have tended to accept expansive definitions that admit anything describing who, what, where, when, why, or how data was collected. Even without a formal definition, it has always been felt that data and its associated metadata must be regarded as an immutable whole. However, achieving that end has not always been satisfactorily accomplished.

I have had the opportunity to collaborate with environmental scientists on a variety of projects including small mammal field studies, detection of coastal change, and measurement of aerosol densities in atmospheric studies. The volume and complexity of data varied enormously across the projects with human field studies collecting a relatively small amount of data while other projects involving remotely sensed data and imagery generated extremely large amounts of data. These associations together with collaborations with other science disciplines have simultaneously increased my awareness and my confusion over exactly what constitutes metadata.

We can distinguish two broad classes of metadata. The metadata capturing the physical characteristics or structure of the data and the metadata associated with the logical interpretation of the data. The physical metadata allows us to decode the raw bits into integers, reals, and other structures. For example, if we are given a binary file we can think of this as the data necessary to print the file (i.e., convert to ASCII) without loss of information. The logical metadata, at least as the term is intended here, is necessary to place the data in context and allows interpretation of the above mentionedintegers and reals within in this context. This is the usual meaning associated with the term metadata [1].

2. Adding to the Confusion

The following sections draw on various collaborations to add perspective to some unique aspects of metadata. Several examples of scientific data collection activities are discussed. The intention is to suggest that the notion of metadata may have to be broadened somewhat to accommodate situations that occur in practice.

2.1. Markup as Metadata

The Global Backscatter Experiment (GLOBE) project underway at the Marshal Space Flight Center (MSFC) is an example of a large multisensor, multinational, multidiscipline effort conducted by multiple investigators. Data is collected from both airborne (aircraft and satellites) and ground-based field programs as well as from other projects initially unrelated to GLOBE but which have collected relevant data.

The role of the GLOBE team at MSFC is two-fold: (1) to receive data from PI's, validate and co-register it; and (2) maintain archives and distribute the data to end users. They have a strong editorial role in guaranteeing the quality of the data. The goals of the GLOBE database project are to: preserve data integrity and ability to interpret data; provide seamless access to data and subsets of data; and to co-register data to facilitate analysis of subsets of data from different sources. The project intends to make raw data and derived data products available to a wide community.

There are many different software packages in use today by scientists seeking to manage their experimental data. Although these packages have common goals, they differ in many important respects [2]. The MSFC group looked at several of the packages commonly used to specify transport formats and representations for scientific data including: Common Data Format (CDF) [3]; Flexible Image Transport System (FITS) [4]; network CDF (netCDF) [5,6]; and Hierarchical Data Format (HDF) [7].

For a variety of reasons,[1] the MSFC group finally chose HDF developed by the National Center for Supercomputing Applications (NCSA) at the University of Illinois, Urbana-Champaign. They added their own header information to each data set to describe details such as PI, type of sensor, start/stop times of data collection, and region covered. This could easily be adapted to produce an ``inventory" display describing the GLOBE database holdings. They have also incorporated HDF tags for identifying subsets of data.

This approach seems to be analogous to descriptive markup [8] in text processing systems. SGML [9] is probably the best known example of such a descriptive markup language. The idea of tagging data descriptively can be used to provide context in a data independent manner. The question arises as to how to interpret the tags. The definition of the tags may be site or processing platform dependent and the definition might change over time. The definition of the tags in effect at the time the data are tagged must be preserved along with the data.

2.2. Provenance as Metadata

The GLOBE project is expected to produce several refined data products. This may involve multiple datasets and significant processing. It seems reasonable to ask what information an end-user of the refined product would need to replicate it from the original data. This is in the spirit of the scientific enterprise where sufficient

[1] Besides the usual advantages of a standardized format and portability provided by most of these formats, HDF is one of the Earth Observing System Data Information System (EOSDIS) preferred formats. In addition, NCSA provides a useful tool set well suited to the GLOBE project needs and they will soon support a netCDF interface.

information should be provided for independent verification of results. Taking a lead from the art world, we call this the data provenance.[2]

This is an aggressive approach to preserving the integrity and interpretability of the data by documenting each processing step in the same file as the data itself. To ensure compliance with this approach it seems necessary to arrange to have all processing steps log themselves (i.e., program name, parameters, platform executed on, etc.) directly into the data file. This has been done previously in analysis environments, but does not become a permanent record. Furthermore, when special processing steps are performed it may be necessary to include source code as part of the documentation. Two examples are: (1) multiple edits batched within a single routine; or (2) a special function applied to correct some flaw in the raw data.

An additional benefit to logging is the possibility of extracting the processing steps from the log and re-executing them to regenerate the data or some earlier version of the data. In the most general case, this might involve recompilations of source code and the execution of processing steps at remote sites. But, theoretically at least, it is possible with today's technology.

The steps involved to incorporate historical processing information are:

1. Start with a base data set, say D_0, (We assume the ``name'' D_0, is known forever.)

2. Apply n processing steps $P_1, P_2, \bullet\bullet\bullet, P_n$

3. When releasing a a derived data product, D_n , append the P_i to D_0 . That is, $D_n = (P_1, P_2, \bullet\bullet\bullet, P_n; D_0)$.

When to save the P_i is a policy decision. Thus, the provenance of $D_i = \left(P_{i_1}, P_{i_2}, \bullet\bullet\bullet, P_{i_n}; D_j\right)$ plus the provenance of D_j.

In some sense, provenance is the backbone metadata providing everything necessary to recreate a derived product from some earlier starting point. Automatic logging can easily be incorporated into analysis programs. An interpreter that re-executes processing steps can easily be built when custom code is not involved and when all steps are intended to be run on a single platform. From a computer science perspective, the problem becomes more interesting when custom code and multiple execution platforms are involved.

2.3. Uncertainty as Metadata

This example arises from a data correlation and fusion application for remotely sensed data. Sensor reports, e.g. imagery, are often the object of correlation and fusion algorithms. A sensor report consists of a measurement vector, **x**, which is the sensor's estimate

[2] Provenance is the history or pedigree of a work of art, manuscript, or rare book. It is a record of the ultimate derivation and passage of an item from the original producer through its various owners. (At the workshop it became evident that this concept, or some variant of it, was known by several names, for example: data archeology, pedigree, lineage, and audit trail.)

of the observed entity's attributes (e.g., location), and a covariance matrix, Σ, which characterizes the measurement error for **x**.

Correlation decides whether two reports constitute sensor observations for the same underlying entity. Fusion combines two correlated reports into a single report and stores the new report in a database.

Although in the actual application the data has much higher dimensionality, a two dimensional location vector is sufficient to make the point. The two dimensional location vector, **x**, is comprised of the pair $(\bar{x}_1, \bar{x}_2)$ and an associated covariance matrix, Σ, defined by

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{bmatrix}$$

where $\sigma_{ii} \equiv \sigma_i^2$ is the variance of X_i and $\sigma_{12} = \sigma_{21}$ is the covariance of X_1 and X_2.

The correlation and fusion problem proceeds as follows. Given an observation X and a correlated database, we first determine whether X is sufficiently correlated with some element Y of the database to declare that X and Y denote the same underlying entity and therefore, that X and Y should be fused into a combined observation, say X'. This is typically achieved by minimizing some statistical distance metric, for example,

$$[X_X - X_Y]^T[\Sigma_X + \Sigma_Y]^{-1}[X_X - X_Y]$$

The question that arises here is whether Σ is data or metadata. One could argue that since Σ is necessary to properly interpret **x**, the location vector, that it falls into the realm of metadata. Σ captures uncertainty just as tolerances in measurement (e.g., 2.374 ± .002*cm*).

Note that Σ requires more storage (3 reals in this case[3]) than **x** (2 reals). This implies that metadata may be more numerous than data.

2.4. Constraints as Metadata

The next example considers how a physical constraint on data may be considered as metadata. We consider a field study of small mammals, in this case mice. The experiment involved monitoring a fixed geographic region to obtain data on the mouse population. These data were to be compared with other available environmental observations (vegetation, rain fall, etc.) to better understand the mouse ecology.

The study involved trapping mice and recording location where trapped as well as the physical characteristics of the mice. When a mouse was trapped for the first time, it was tagged with a small plastic tag for subsequent identification. If a tagged mouse was retrapped, its physical characteristics were rerecorded. Among other things, the following field data were collected: date of observation, location of trap, ID tag number, sex, weight, estimated age, and if female, whether the mouse was pregnant or not.

[3] In general the covariance matrix will require $n(n+1)/2$ reals since it is a symmetric matrix.

Now suppose the field logs contained the observations shown in Figure 1. The implicit constraint on mammals that only the females can become pregnant implies that there is some recording error in the starred data of Figure 1. It must be the case that either: (1) mouse #207 is a female and there is a sex recording error in the first observation; or (2) there has been a tagging error and one of the mice is not mouse #207, or neither is.

Date	Trap	Tag No.	Sex	Weight	Age	Pregnant
			•••			
* 03/04	29	207	M	147	2	n/a
03/04	63	198	F	120	1	N
			•••			
* 04/22	42	207	F	152	2	Y
			•••			

Figure 1

This sort of situation is easily resolved by humans, but if the constraint has not been recorded explicitly, it would not be possible for an automated process to detect it. The point is that some constraints, whether stated or not, are essential metadata.

3. Summary

This paper does not answer the question posed in the title. The foregoing examples were intended to suggest that the nature of metadata may be more varied and elusive than suggested by the usual partition into structured (physical) and descriptive (logical) metadata. It is certainly the case that the appropriate metadata for a dataset might be a Fortran program together with the parameters used to run it. Descriptive data, uncertainty, and constraints are all candidates for consideration as metadata. An appropriate definition of metadata must accommodate all the examples discussed here.

Before we can be successful managing metadata, we must be more precise defining it. An answer to the question ``What is metadata?" is a necessary first step.

References:

1. A. Shoshani and H. K. T. Wong, Statistical and Scientific Database Issues, *IEEE Trans. on Software Eng. SE-11*,10 (Oct.1985), 1040-1047.

2. L. A. Treinish, *Data Structures and Access Software for Scientific Visualization*, Summary of workshop held at SIGGRAPH'90, Jan. 1991.

3. L. A. Treinish and M. L. Gough, *A Software Package for the Data-Independent Management of Multi-Dimensional Data*, National Space Science Data Center, NASA Goddard Space Flight Center, Greenbelt, Maryland, Apr. 1987.

4. D. C. Wells, FITS - A Self-Describing Table Interchange Format, in *Scientific Database Management (Panel Reports and Supporting*

Material) (Tech. Rep. CS-90-22), J. C. French, A. K. Jones and J. L. Pfaltz (editors), Dpt. of Computer Science, University of Virginia, Charlottesville, VA, Aug. 1990.

5. *NetCDF User's Guide: An Interface for Data Access*, Unidata Program Center, Mar. 1991.

6. R. Rew and G. Davis, NetCDF: An Interface for Scientific Data Access, *IEEE Computer Graphics and Applications 10*(July 1990), 76-82.

7. *NCSA HDF Calling Interfaces and Utilities*, University of Illinois at Urbana-Champaign, July 1990.

8. J. H. Coombs, A. H. Renear and S. J. DeRose, "Markup Systems and the Future of Scholarly Text Processing", *Comm. of the ACM 30,* 11 (Nov. 1987), 933-947.

9. SGML: Standard Generalized Markup Language, ISO 8879, ISO, 1986.

Metadata in the Atmospheric Radiation Measurement Program

R.B. Melton
Pacific Northwest Laboratory
Richland, WA 99352

Introduction

The Atmospheric Radiation Measurement (ARM) Program is a major U.S. Department of Energy program intended to develop improved general circulation and related models. The Clouds and Radiation Testbed (CART) is the operational infrastructure of the ARM Program. CART will instrument and plans to operate five field sites over a ten year period to acquire and process atmospheric and radiometric data. Members of the ARM Science Team will use the data to test, diagnose, and improve their models. The ARM Program Plan provides a complete description of the program [1].

Unlike many atmospheric sciences field programs, the ARM instruments are owned and operated by the program. The program's principal investigators, the ARM Science Team, are a minimum of two steps removed from the instruments. This is in contrast to typical field campaigns where the principal investigators field and operate the instruments. This difference makes metadata particularly important. Because the scientists do not directly operate the instruments they do not have first-hand knowledge of the context within which the data are collected. The primary source of contextual information is the metadata collected with the principal data[1] or generated during processing and analysis of the data.

This paper summarizes our experience with metadata. First, categories of metadata are described, then our approach to managing the metadata is discussed.

ARM Data Structures

In ARM, sources of data are referred to as platforms. A platform may be an algorithm, a single instrument, a network of instruments (referred to as sub-platforms) that is geographically distributed, or some other source of data. Any platform or sub-platform may have multiple sensors. The individual sensors produce a variety of data types including images, vector time-series, scalar time-series, and regular grids.

Categories of Metadata

During the design and implementation of the CART Data Environment, we have considered metadata to exist in parallel with the primary data. In other words we have identified the possibility for metadata to exist in association with individual data streams from individual sensors; with individual sources of data,

[1] We use the term "principal data" to refer to the environmental observations produced by the instrument, e.g. temperature.

i.e., platforms; and across multiple platforms. Given this perspective, one way to categorize the metadata is based on its potential frequency of change: fast, slow, and static.

Fast metadata are those that may change from data point to data point. These metadata are parallel data streams to the primary data streams. In ARM the principal examples of this type of metadata are quality assessment results. We apply statistical or other techniques to assess the probable validity of each data point in the data stream. The techniques we are using produce one or more results per data point.

Slow metadata are those that change less frequently than each data point. There are probably several sub-categories. These metadata may be associated with a platform sensor; a platform; or with a temporal chunk of data, i.e., a dataset. They may also be associated with a set of platforms providing, for example, information about the general conditions at the site where the instruments are deployed.

Static metadata are those that do not typically change or change on a very long time scale. These may be associated with instruments or with the operation of instruments. For example, the ARM Central Facility in the Southern Great Plains of the United States is located at a specific location that will not change.

Metadata Management

Our metadata management strategy is to associate the metadata with the principal data as tightly as possible. We store fast metadata as additional data streams in parallel with the sensor data. We store slow metadata associated with sensors, platforms, and datasets in the file containing the dataset. Slow metadata that provide general information not associated with an individual dataset or platform and static metadata are stored in databases.

We have also chosen to use standard tools and data formats for storing data and metadata. All non-imagery data are stored using netCDF [2]. Imagery data are stored using HDF [3]. The slow and static general metadata are acquired and stored using EMPRESS or dbm, a UNIX data management utility. Some static metadata such as maps are maintained in hard copy format.

Figure 1 is a sample header from an ARM netCDF dataset. The first section identifies the file and the dimension of the arrays in the file. In this case the only dimension is time. The second section, "variables", contains the field[2] level attributes. One field is shown - "pres." The attributes give the units; a long-name for the field; and minimum, maximum and rate-of-change limits that can be used for simple data quality checks. These correspond to "slow" metadata associated with a sensor. Three fields, qcmin 1-24, qcmax 1-24, and qcdelta 1-24, are provided to store the results of limit checking in parallel with the principal data fields. The qc results from the individual fields are encoded into a single flag.

The next section is global attributes. These are attributes for the data chunk or the platform that apply to all fields. Note

[2] The primary data are stored in “fields”.

```
netcdf sgpsonde1.a1.930201.1928 {
dimensions:
   time = UNLIMITED ; // (132 currently)

variables:
   long base_time ;
   double time_offset(time) ;
   float pres(time) ;
     pres:units = "hPa" ;
     pres:descrip = "Pressure" ;
     pres:calc = "pres " ;
     pres:min = "0.0" ;
     pres:max = "1100.0" ;
     pres:delta = "50.0" ;
                 •
                 •
                 •
   float qcmin1-24(time) ;
   float qcmax1-24(time) ;
   float qcdelta1-24(time) ;
   float lat(time) ;
   float lon(time) ;
   float alt(time) ;

// global attributes:
     :ingest-software = " sonde_ingest.c,v 2.12 1993/01/25 17:42:38 caraher Release2_1 slater $" ;
     :missing-data = "-9999" ;
     :site-id = "sgp" ;
     :facility-id = "C1 : Central_Facility" ;
     :sds-mode = "production" ;
     :sample-int = "1.2 seconds" ;
     :averaging-int = "10 seconds" ;
     :comment = "Latitude and Longitude are in degrees and are the\n",
   "location of the launch point.\n",
   "Altitude is in meters above MSL" ;
     :sonde_pc_software_version = "7.61" ;
     :phase_fitting_1 = "Phase fitting length is  60 s from   0 min to  10 min\r\n",
   "" ;
     :phase_fitting_2 = "Phase fitting length is 120 s from  10 min to  45 min\r\n",
   "" ;
     :phase_fitting_3 = "Phase fitting length is 240 s from  45 min to 120 min\r\n",
   "" ;
     :sounding_number = "110201931" ;
     :serial-number = "100425145" ;
     :pressure_correction = "0.0" ;
     :temperature_correction = "-0.0" ;
     :humidity_correction = "1.0" ;
     :proc-level = "a1" ;
     :input-source = "A0 data:sonde1:/apps/ingestdata/bbs-cf/in/prt.cur" ;

data:
}
```

Figure 1: Sample ARM netCDF header.

that the version of the software that reads in the data is provided to document the history of the data stream. The next section of the file, which has been deleted in this example, would contain the individual data fields.

Summary

In designing and implementing the CART Data Environment we have developed a hierarchical model of metadata that parallels the structure of the principal data. Our implementation strategy has been to directly link the two structures whenever possible so that data and metadata are stored together.

We have also learned some important lessons. First, the volume of metadata can be significantly larger than the volume of primary data. For time-series data, where we allow for metadata to be associated with each individual data point, the ratio of metadata to primary data is in the range of 3 to 7. For imagery data, where we do not attempt to associate metadata with individual pixels, the ratio is much smaller than 1.

Second, tools for looking at metadata are as important as tools for looking at primary data. Tools that provide an integrated view of primary data and metadata are most desirable but do not yet exist.

Finally, it is important to deal with metadata explicitly during design. It is tempting to model metadata as the set of attributes of the principal data. While this is a valid perspective, it tends to hide the role of metadata in searching and creating subsets of the principal data. We believe that modeling the metadata explicitly with appropriate linkages to the principal data is a more useful approach.

References

[1] U.S. Department of Energy (DOE). 1990. *Atmospheric Radiation Measurement Program Plan*. DOE/ER-0441, Washington, D.C.

[2] Unidata Program Center. March 1991. *NetCDF User's Guide: An Interface for Data Access*. Boulder, CO.

[3] University of Illinois at Urbana-Champaign. July 1990. *NCSA HDF Calling Interfaces and Utilities*, Champaign, IL.

Metadata Compiled and Distributed by the Carbon Dioxide Information Analysis Center for Global Climate Change and Greenhouse Gas-Related Data Bases

Thomas A. Boden
Carbon Dioxide Information Analysis Center*
Environmental Sciences Division
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6335

1. The Carbon Dioxide Information Analysis Center

The Carbon Dioxide Information Analysis Center (CDIAC) compiles and provides information to help international researchers, policymakers, and educators evaluate complex environmental issues, including potential climate change, that are associated with elevated levels of atmospheric carbon dioxide (CO_2) and other trace gases.

CDIAC is located within the Environmental Sciences Division of Oak Ridge National Laboratory (ORNL) in Oak Ridge, Tennessee, and is line-funded by the U. S. Department of Energy's (DOE) Global Change Research Program (GCRP). CDIAC, which has been in operation since 1982, is an information analysis center (IAC) and conducts the following activities:

- identifies sources of primary data at national and international levels;
- obtains, archives, evaluates, and distributes data and computer models;
- fully documents select data sets and computer models and offers them as numeric data packages (NDPs) and computer model packages (CMPs);
- distributes data and computer models on a variety of magnetic and electronic medias, including 9-track magnetic tapes, IBM-formatted floppy diskettes, and CD-ROM, and over Internet, Omnet, and Bitnet electronic networks;
- develops derived, often multidisciplinary, data products useful for carbon cycle and climate change research;
- distributes reports pertinent to greenhouse effect and climate change issues;
- produces the newsletter *CDIAC Communications;* and
- acts, in general, as the information focus for theGCRP's research projects.

Since its inception, CDIAC has responded to thousands of requests for information, and since 1985 it has distributed more than 70,000 reports, NDPs, and CMPs to 97 countries worldwide.

2. The Information Analysis Center

As the scope of scientific inquiry expands to include global issues, the data that must be analyzed increase in magnitude, spatial coverage, organizational complexity, and temporal scale (see Fig. 1).

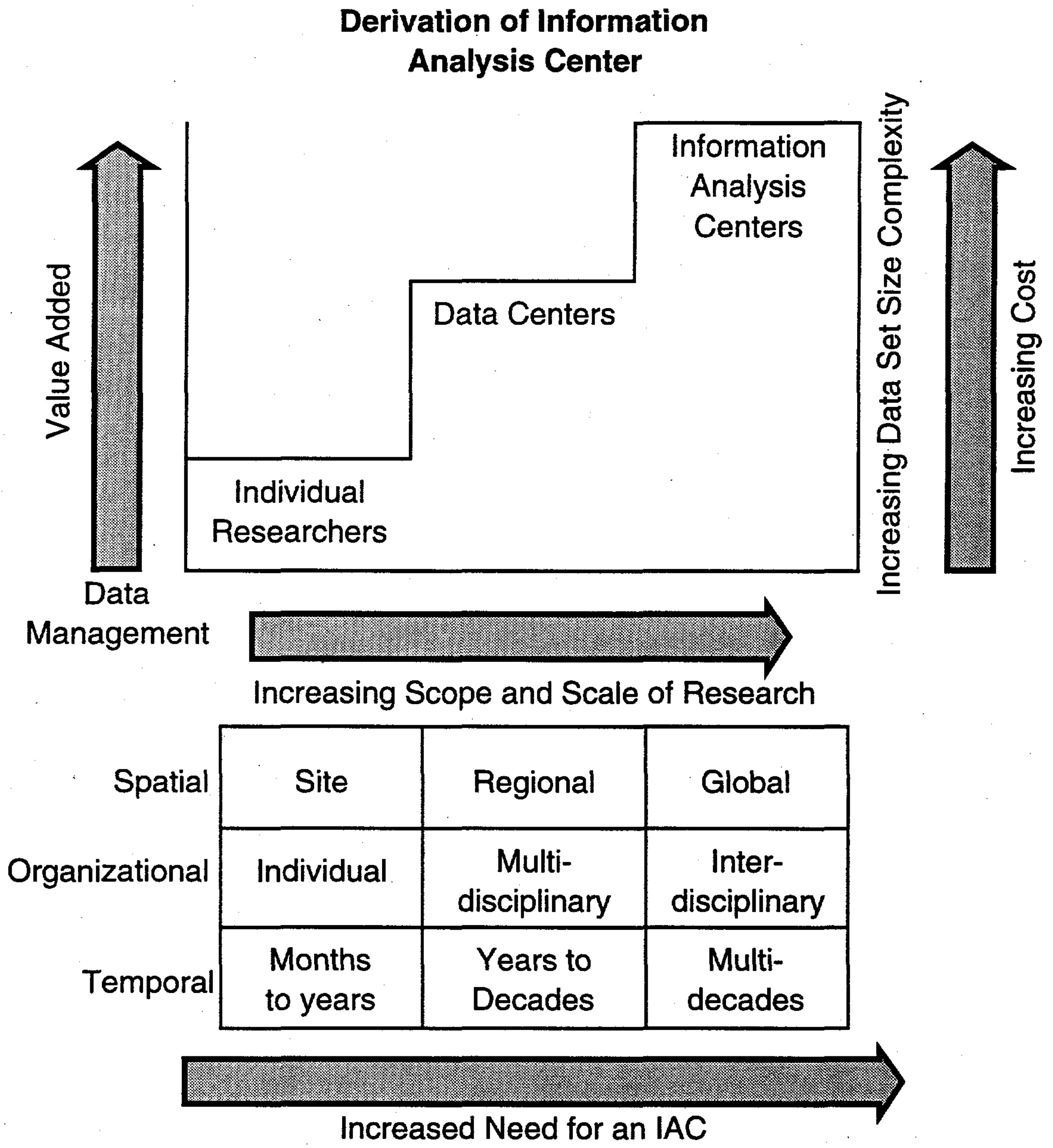
There often is a gap between what the individual researcher or disciplinary data center produces as a dataproduct and what the new global change researcher needs to support his or her analysis. Multidisciplinary data sets suitable for analysis by interdisciplinary teams are becoming more essential to global change research. As a result, individual and data-center approaches to data management become inadequate by themselves. At this point, the concept of a proactive information analysis center becomes justified, in which information refers to the entire suite of support services it renders and analysis refers to its proactive, analytical role in identifying and supplying value-added, derived, multidisciplinary information products.

* CDIAC is sponsored by the U.S. Department of Energy, Office of Health and Environmental Research, Environmental Sciences Division. Oak Ridge National Laboratory is managed by Martin Marietta Energy Systems, Inc., for the U.S. Department of Energy under Contract DE-AC05-84OR21400. Environmental Sciences Division Publication No. 4022

The study of global issues requires a data management system that can provide not only the vehicle for exchange of extant, acquired information, but also the mechanism for adding value to existing information. The IAC synthesizes new, derived, multidisciplinary data sets when needed, provides for the highest standards of quality assurance and documentation, and supports the full complement of information services necessary to address a global change issue.

The IAC does not replace the need for traditional discipline-oriented data centers; traditional centers have and will continue to play a fundamental role in supporting global change research. Instead, the IAC concept adds to the traditional role of centers by expanding the concept of the data center.

The IAC is issue-driven, applying innovative approaches to research data management and


Derivation of Information Analysis Center
Value Added
Data Management
Individual Researchers
Data Centers
Information Analysis Centers
Increasing Data Set Size Complexity
Increasing Cost
Increasing Scope and Scale of Research
Increased Need for an IAC

Spatial	Site	Regional	Global
Organizational	Individual	Multi-disciplinary	Inter-disciplinary
Temporal	Months to years	Years to Decades	Multi-decades

Fig. 1 Data management requirements of different information environments and the derivation of an Information Analysis Center.

augmenting available information resources. It produces a higher level of data and information products to support a broadly defined user community that includes not just researchers, but policy makers and educators as well.

The IAC does not have the responsibility to archive all the data in a particular discipline or, indeed, all the data related to a particular issue, but it works with the research and policy communities to identify those information products most needed by their communities. It then endeavors to acquire them or, in many cases, works to assemble derived products, creating new, value-added information packages. This freedom from the disciplinary data center's obligation to accept all the data generated in a specific field of research is a significant key to the potential success of an IAC. An IAC without both a focused mission and freedom to selectively acquire data in support of its users would fail.

The IAC is able to focus its attention on a smaller list of data products and services that uniquely and specifically address the issue at hand. Justifiable under the purview of an IAC are such activities as determining the need for and compiling multidisciplinary data sets; applying extensive quality assurance to extant data sets to derive new, benchmark environmental data bases; compiling custom-annotated bibliographies and bibliographic searches; publishing resource catalogs, directories of researchers and policy makers, and newsletters; and sponsoring workshops. The IAC must be responsive to its user community and must perform whatever information anddata-related tasks are necessary to support its issue-oriented field of research.

3. CDIAC's Metadata

CDIAC is an IAC. CDIAC's basic philosophy towards data management and metadata is to offer select, critical data bases with tremendous levels of documentation instead of offering many data bases with little or no accompanying metadata. We have the luxury and freedom from our DOE sponsors to be selective in the data bases we archive, document, and distribute. This freedom allows us to devote more of our resources to quality assurance, data analysis, and documentation efforts than to basic archival commitments. It also allows us to prepare comprehensive documentation far and above what traditional discipline-oriented data centers can offer. The metadata we prepare is offered in the form of NDPs and CMPs. These packages attempt to provide every facet of information a potential user might need to access, use, and analyze a primary scientific data base. The NDPs and CMPs ensure that these data bases may be easily used and understood for decades. Presently, CDIAC offers approximately 50 NDPs and CMPs with primary data files ranging from 3400 bytes to 1.2 gigabytes in size. NDPs and CMPs offer both written and digital metadata. These documents discuss not only the structure of a data base but also its contents, limitations, history, and past and potential applications. The written metadata documentation provides the following information about the primary data:

- names and affiliations of the principal investigators or individuals who originally compiled the data base;
- historical background information (i.e., Why was the data base compiled? Who funded the effort?);
- source and scope of the data base, including the methodology used to create the data base and the temporal and spatial coverage of the data base;
- limitations and restrictions of the data base;
- potential uses of the data base;
- detailed descriptions of the structure of the data base (i.e., data set names, contents, formats, units, codes, etc.);
- graphical data presentations;
- partial listings of data files;
- lists of key references; and
- reprints of published papers that discuss the compilation of the data base or an analysis of the data base.

The digital metadata may include files that contain:

- calibration and reference gas standards,
- station inventories,
- station histories,
- flag codes,
- equipment performance results,
- quality assessments (Z-scores, results from homogeneity checks, etc.), and
- footnotes or comments

The process by which these NDPs and CMPs are compiled is shown in Fig. 2. Several key components of this process are worth discussing further.

Quality assurance and adding value

Rarely does CDIAC receive information that is immediately ready for distribution; either there are problems with the primary data or because insufficient metadata exists to fully understand and analyze the primary data base. We assume that any data, either primary data or metadata, that arrives at CDIAC has problems. To guarantee data of the highest possible quality, CDIAC conducts extensive quality assurance (QA) reviews. Reviews involve examining the primary and ancillary data for completeness, reasonableness, and accuracy. Although they have common objectives, these reviews are tailored to each data set, often requiring extensive programming efforts. Although time-consuming, the QA process is an important component in the value-added concept of ensuring accurate, usable data for researchers.

Technical review and collaboration

In order to compile and obtain sufficient written and digital metadata for a scientific data base, we have found that is extremely helpful to involve the individuals responsible for compiling the primary data base in our review of that data base and the preparation of the accompanying metadata. We have also found that users can be of great assistance in identifying needed metadata.

At CDIAC, we encourage the principal scientists to assist us in establishing our QA checks for their primary data and identifying what metadata is needed. We also insist that they review whatever metadata is prepared prior to its release. CDIAC does not distribute a data base until the contributing scientist has reviewed the metadata prepared by us and granted written permission for CDIAC to disseminate it. We seek the assistance of the principal scientists in our QA checks simply because no one better understands the contents of the data base. Few, if any, data centers have staff with training in all the disciplines required to properly address the variety of data bases encountered in global change research. CDIAC is no exception, although our staff have technical backgrounds across many scientific disciplines. Instead, CDIAC complements their in-house talent with the expertise of the contributing scientist and other scientists located at ORNL in developing appropriate data checks and determining what digital metadata will be needed by users.

It is often difficult, even with the help of other scientists, to identify or anticipate what metadata will be needed by a data base user for their particular analysis or research. We try to anticipate the metadata needs of potential users by sending preliminary versions of the primary data and accompanying metadata to "beta test sites" or potential users who have a keen interest in the data base and the abilities to rigorously exercise the data base. This is done to ensure that the level and degree of metadata prepared thus far are sufficient and to create a mechanism by which additional metadata suggestions can be received and still implemented before wide distribution.

Feedback

CDIAC maintains records of all individuals that request data from us. These records allow us to identify our user community and improve products to meet their needs, notify data base recipients of updates or revisions, and compile lists of recipients that we share with our contributing scientists. Periodically, we survey our users so that we may determine, among other things, whether they found the level of our documentation to be sufficient, what other types ofancillary metadata would benefit them, what computing capabilities they have, and on what new types of media they would like to see NDPs and CMPs made available.

4. Metadata Problems Encountered by CDIAC

The three biggest problems encountered by CDIAC regarding metadata are the expense of preparing it, the need to identify and anticipate what metadata are needed by our users, and the need to preserve the usefulness of the primary data base by not offering too much metadata. Preparing metadata is very expensive, whether it entails preparing a manuscript, preparing digital station histories, or generating summary products. Metadata prepared for one NDP or

CMP drain resources and time that could be devoted to other data products. CDIAC has a diverse user community, and it is impossible to identify or anticipate the needs of every potential user. Metadata that are essential for some users may not be needed or wanted by others. Often the written and digital metadata become quite voluminous. This can be very cumbersome, and even annoying, for the user who wants to extract only a few records from the primary data base or just "wants the results". To date, we have attempted to compile and prepare the most comprehensive metadata possible with a general focus on the metadata needs of other researchers.

5. *Conclusions*

Metadata are essential for climate change and greenhouse gas-related research. CDIAC devotes considerable time and resources preparing written and digital metadata and trying to identify what quantity and level of metadata are needed by CDIAC's diverse, multidisciplinary audience. CDIAC's basic philosophy is to offer a few benchmark data sets that are fully complemented

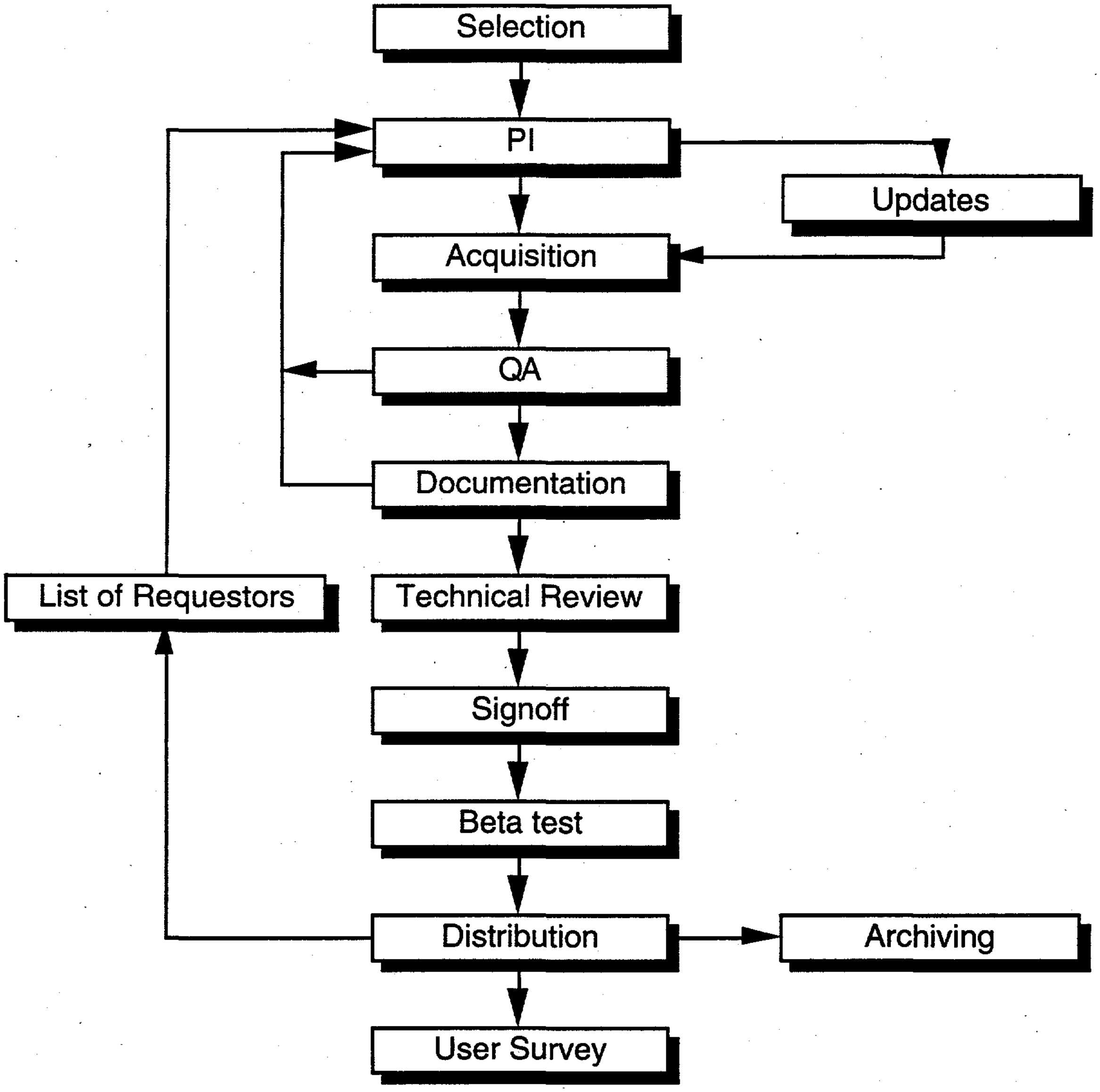


Fig. 2 The process followed by the Carbon Dioxide Information Analysis Center in compiling textual metadata.

with metadata rather than offer many insufficiently documented data bases. The process we follow to compile our metadata addresses many metadata issues and results in superior metadata. CDIAC's philosophies and methods are ideal for preparing comprehensive metadata for a handful of moderate-sized data bases each year. We recognize that not all data centers and agencies have the freedoms that we have in preparing metadata and that the process used by CDIAC to compile metadata is not appropriate for all data bases and projects. However, certain key components of CDIAC's process of compiling metadata, namely the assumption that data need further scrutiny and documentation before release, the interaction and involvement with scientists and users during preparation of metadata, and the post compilation feedback, are appropriate for any project that attempts to capture, compile and distribute scientific metadata.

Directory Interchange Format: A Metadata Tool For The Noaa Earth System Data Directory

Gerald S. Barton
October 26, 1992

Earth System Data and Information Management Program
National Oceanic and Atmospheric Administration
1825 Connecticut Ave. NW, Washington, DC 20235

The NOAA Earth System Data Directory is an on-line computer guide to environmental data held by the National Oceanic and Atmospheric Administration. Any user can access the directory to search for data maintained in NOAA offices. Searches can be made by several options such as discipline, location, parameter, key word, and data center. The user may review descriptions of data set found by the search. A key field of the description is a pointer to the office which holds the data. NOAA is a participant in the Interagency Working Group on Data Management for Global Change, and the NOAADIR is part of the International Directory Network. Telecommunication linkage to the Global Change Master Directory and use of the Data Interchange Format allow interchange of data descriptions with other directories in the system. The NOAADIR is the key to NOAA's Earth System Data and Information Management Program.

1. INTRODUCTION

The NOAA Earth System Data Directory (NOAADIR) is an on-line computer guide to environmental data held by the National Oceanic and Atmospheric Administration. The NOAADIR was established in late 1989 to serve two major functions:

1. it provides NOAA with a common system for documenting data held in NOAA offices
2. it provides the general research and scientific community with the means to locate NOAA data sets useful for their studies.

The directory is part of a national and international network of Data Directories based on the NASA Master Directory. The International Directory Network (IDN) extends the information worldwide. Most of the Directories on the network use identical software developed for the NASA Master Directory system.

The common metadata definitions are defined in the Directory Interchange Format, or DIF, which was developed by NASA/NOAA/USGS team as a standard way of documenting high level information about space and environmental data sets (NASA 1989). All of the directories in the Master Directory system use the DIF structure or can exchange data using DIF.

2. NOAA EARTH SYSTEM DATA DIRECTORY

The NOAADIR is operational on a VAX-11/780 computer located in the National Oceanographic Data Center in Washington, DC. This central location is available to all users as a free service to NOAA customers.

NOAADIR uses software developed for the NASA Master Directory. NASA has made the software available to institutions who are participating members of the International Directory Network. The software uses Fortran and C programming languages for system functions and SQL commands for communications with the Data Base Management System (DBMS) which contains the data descriptions. NOAADIR uses the ORACLE DBMS, while the NASA Master Directory uses the Sharebase DBMS Machine for the data base. The use of the SQL commands allow the use of different DBMS software at different computer sites.

NOAADIR can be accessed from anywhere in the United States via direct phone lines including a toll free 800 number, connections to the NASA Space Physics Applications Network, and connections to the Telenet Telecommunications Network.

The user connects with NOAADIR via a controlled VAX account called NOAADIR. This account allows the user to access only the search facilities of the NOAA directory. The user cannot access any other software on the VAX, thus assuring the security of the system.

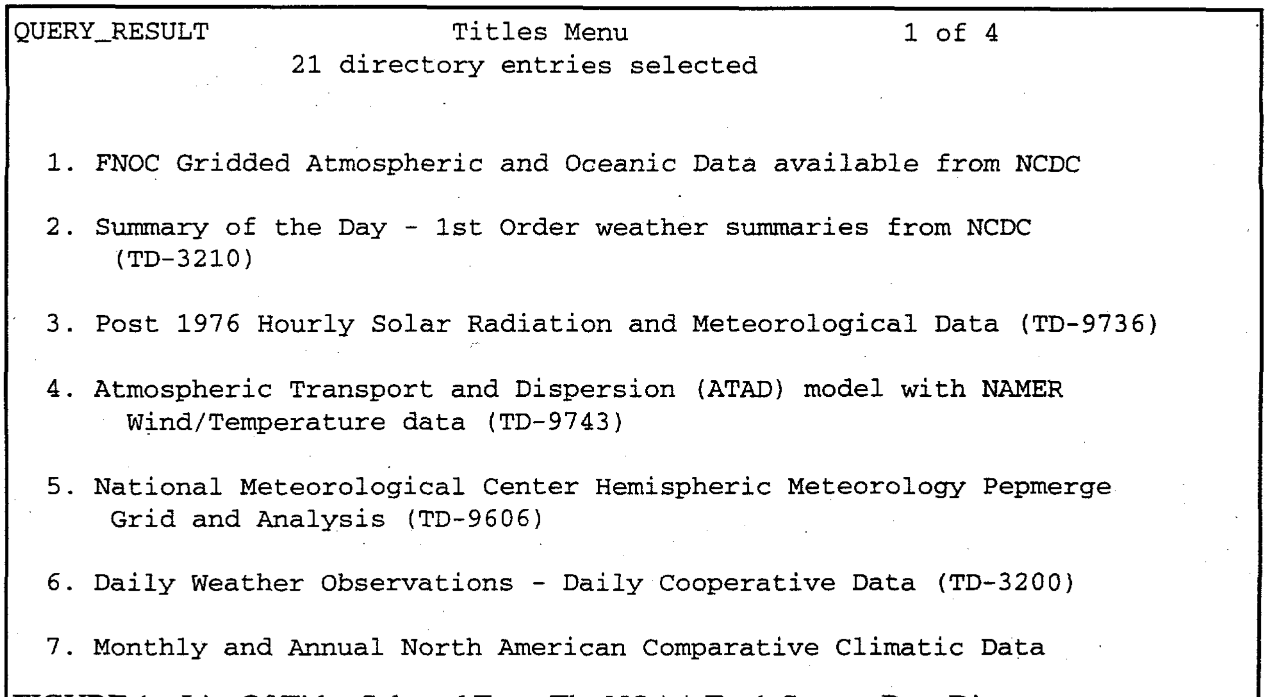
3. SAMPLE NOAA DATA DESCRIPTION

A description of a NOAA data set in the Directory Interchange Format provides a snapshot of the data set including DATA SET ID, TITLE, SUMMARY description, START and STOP DATES, SENSOR NAME, SOURCE NAME, INVESTIGATOR, TECHNICAL CONTACT, DATA CENTER, PARAMETERS, KEYWORDS, LOCATION, geographic COVERAGE, and REFERENCE.

The search software is presented to the user in a series of screen oriented menus. The user is guided through the search steps via the menus, and HELP is available at any point. A search can be made on a variety of fields, for example, DATA CENTER, LOCATION, PARAMETER, or KEYWORDS. The most reliable search fields are those which have fixed terms, such as LOCATION, DATA CENTER, and PARAMETERS. The user can view all of the valid words for a field by entering a ? symbol which results in a numbered list of permissible terms. The entry of the number of the desired term adds the term to the search parameter list.

The result of a search is all of the data sets in the data base which meet the search criteria. The user is given the list in a series of screens each containing about seven data set TITLES. A sample listing of titles from the NOAADIR is given in Figure 1.

The user selects a desired data set by number to view the complete data description. Data descriptions are listed in screen format, and one can move back and forth between the pages. The information could be captured, if desired, to the user's Personal Computer disk for further use.



```
QUERY_RESULT                  Titles Menu                 1 of 4
                  21 directory entries selected


  1. FNOC Gridded Atmospheric and Oceanic Data available from NCDC

  2. Summary of the Day - 1st Order weather summaries from NCDC
      (TD-3210)

  3. Post 1976 Hourly Solar Radiation and Meteorological Data (TD-9736)

  4. Atmospheric Transport and Dispersion (ATAD) model with NAMER
       Wind/Temperature data (TD-9743)

  5. National Meteorological Center Hemispheric Meteorology Pepmerge
      Grid and Analysis (TD-9606)

  6. Daily Weather Observations - Daily Cooperative Data (TD-3200)

  7. Monthly and Annual North American Comparative Climatic Data
```

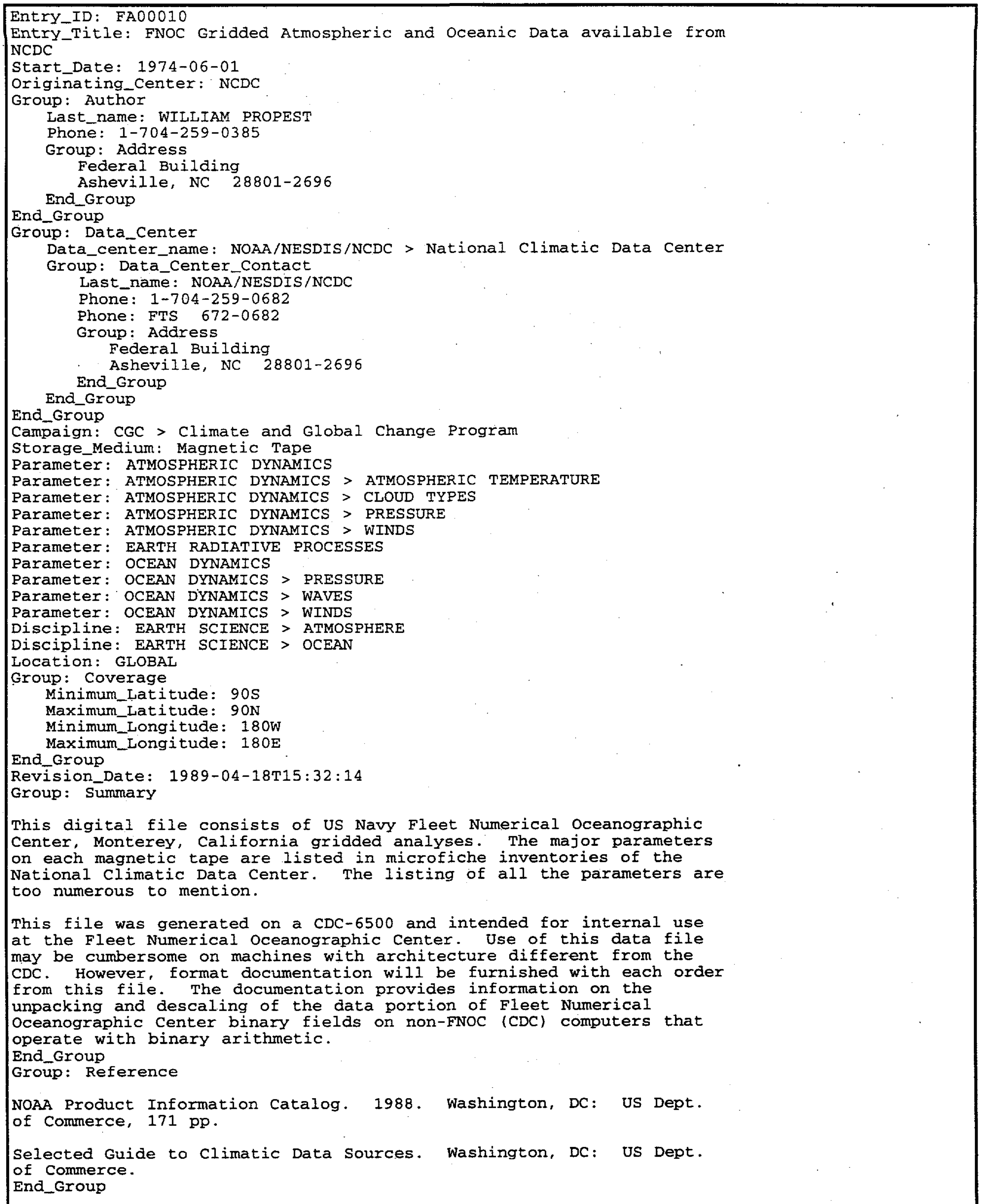
FIGURE 1. List Of Titles Selected From The NOAA Earth System Data Directory

A sample listing of a NOAA data description in the Directory Interchange Format is given in Figure 2.

4. NOAA DIRECTORY AS A DATA MANAGEMENT TOOL

The NOAA Earth System Data Directory gives NOAA control over data sets located in many different parts of the organization. NOAA is a large organization with over 12,000 employees located throughout the United States. There are five major components of NOAA which collect, process, and analyze data sets and produce products from the data.

The NOAADIR is the keystone in the NOAA Data Management effort. The directory will allow NOAA managers to know where their data are located, who is collecting and processing the data, and where the data are available for distribution to users. This will give NOAA managers control over their data which they do not have today. Outside users can use the directory to locate data sets which may be useful for their research. Future enhancements to the directory will allow the user to transfer from the directory to a NOAA data system. Once transferred to the data system, such as the National Climatic Data Center, the user will be able to access inventory systems, order data, and perhaps access selected data sets. The capabilities to "Link" to other systems are already available in the software system, but network linkages and software procedures in the data centers must be established. These capabilities will provide a powerful data system to all users of NOAA data.



```
Entry_ID: FA00010
Entry_Title: FNOC Gridded Atmospheric and Oceanic Data available from
NCDC
Start_Date: 1974-06-01
Originating_Center: NCDC
Group: Author
   Last_name: WILLIAM PROPEST
   Phone: 1-704-259-0385
   Group: Address
      Federal Building
      Asheville, NC  28801-2696
   End_Group
End_Group
Group: Data_Center
   Data_center_name: NOAA/NESDIS/NCDC > National Climatic Data Center
   Group: Data_Center_Contact
      Last_name: NOAA/NESDIS/NCDC
      Phone: 1-704-259-0682
      Phone: FTS  672-0682
      Group: Address
         Federal Building
         Asheville, NC  28801-2696
      End_Group
   End_Group
End_Group
Campaign: CGC > Climate and Global Change Program
Storage_Medium: Magnetic Tape
Parameter: ATMOSPHERIC DYNAMICS
Parameter: ATMOSPHERIC DYNAMICS > ATMOSPHERIC TEMPERATURE
Parameter: ATMOSPHERIC DYNAMICS > CLOUD TYPES
Parameter: ATMOSPHERIC DYNAMICS > PRESSURE
Parameter: ATMOSPHERIC DYNAMICS > WINDS
Parameter: EARTH RADIATIVE PROCESSES
Parameter: OCEAN DYNAMICS
Parameter: OCEAN DYNAMICS > PRESSURE
Parameter: OCEAN DYNAMICS > WAVES
Parameter: OCEAN DYNAMICS > WINDS
Discipline: EARTH SCIENCE > ATMOSPHERE
Discipline: EARTH SCIENCE > OCEAN
Location: GLOBAL
Group: Coverage
   Minimum_Latitude: 90S
   Maximum_Latitude: 90N
   Minimum_Longitude: 180W
   Maximum_Longitude: 180E
End_Group
Revision_Date: 1989-04-18T15:32:14
Group: Summary

This digital file consists of US Navy Fleet Numerical Oceanographic
Center, Monterey, California gridded analyses.  The major parameters
on each magnetic tape are listed in microfiche inventories of the
National Climatic Data Center.  The listing of all the parameters are
too numerous to mention.

This file was generated on a CDC-6500 and intended for internal use
at the Fleet Numerical Oceanographic Center.  Use of this data file
may be cumbersome on machines with architecture different from the
CDC.  However, format documentation will be furnished with each order
from this file.  The documentation provides information on the
unpacking and descaling of the data portion of Fleet Numerical
Oceanographic Center binary fields on non-FNOC (CDC) computers that
operate with binary arithmetic.
End_Group
Group: Reference

NOAA Product Information Catalog.  1988.  Washington, DC:  US Dept.
of Commerce, 171 pp.

Selected Guide to Climatic Data Sources.  Washington, DC:  US Dept.
of Commerce.
End_Group
```

FIGURE 2. Sample NOAA Data Description Listed In Directory Interchange Format

NOAA is a participant in the Interagency Working Group on Data Management for Global Change, which has established the Global Change Master Directory (GCMD). NOAADIR is part of the International Directory Network sponsored be the Committee on Earth Observation Satellites. The IDN is a system of data directories which includes the GCMD, the European Space Agency Directory, and the Japanese Master Directory. Telecommunication linkages and use of the DIF allow interchange of data descriptions with other directories in the system.

5. CONCLUSION

The directories in the IDN system are based on a metadata standard called the Directory Interchange Format. The use of the DIF provides a common structure and vocabulary definitions which allow data from many disciplines to be described, maintained, and interchanged between directory systems. The system is operational nationally and internationally, and is the basic tool for data and information management in the Global Change community.

6. REFERENCE

NASA National Space Science Data Center, December 1991: Directory Interchange Format Manual Version 4.0, NASA Goddard Space Flight Center, NSSDC/WDC-A R&S 91-32.

The Global Land Information System: The Use of Metadata on Three Levels

D. K. Scholz and T. B. Smith
Hughes STX Corporation[1]

The U.S. Geological Survey (USGS) Global Land Information System (GLIS) was developed to provide an interactive on-line inquiry system for global-change researchers to access and order satellite and thematic data sets. GLIS supports each data set by providing descriptive information known in the data management community as metadata.

1. Three Levels of Metadata within GLIS

GLIS contains metadata for the holdings of the Department of the Interior (DOI) agencies as well as data held by other Federal agencies, research organizations, and academic institutions. These metadata are organized in GLIS to meet the following three levels of information needs:

1) summary or directory information to aid the researcher in identifying the types of data sets that might meet research needs;

2) detailed user guides that provide specific information about each data set; and

3) necessary descriptive geographic, date, and similar fields common to all inventory data bases to facilitate searching across otherwise different inventories.

Directory-level metadata

Directory-level metadata within GLIS consists of high level descriptive information about data sets in data base field format. Similar fielded metadata are created for each data set in GLIS to ensure a common descriptive structure for all data sets. The structure used for this level of metadata is the directory interchange format (DIF), originally developed as part of the National Aeronautics and Space Administration Global Change Master Directory.

The DIF structure includes fields such as geographic spatial coverage, start and stop date for data collection, originating data center, distributing data center, descriptive keywords, references, and a brief descriptive paragraph.

The DIF field structure accomplishes two things. First, it facilitates development of a data base query capability that allows researchers who are unfamiliar with available data sets to search by coverage area, acquisition date, keyword, etc. Secondly, use of the DIF structure at many data centers ensures that a common framework and a minimum level of metadata are available to search data sets that are maintained elsewhere. This capability is becoming more critical as the number of data centers with online inventories continues to grow.

Guide-level metadata

Once a researcher has identified potentially useful data sets, further refinement of the candidate data sets can be made by more careful review of guide-level metadata. In GLIS, the guide is created and used as if it were a hardcopy user manual, but with all the conveniences of electronic media. That is, technical or less widely understood terms are hypertext linked by an online glossary or are directed to other areas within the guide. Terms or linkages are shown in highlighted text so that the GLIS user can quickly move from these terms to their explanations or related descriptions, and then return to continue reading the original document. Hypertext links not only tie text together, but also link illustrations and data samples with their proper locations in the text.

To ensure a sense of continuity and to facilitate comparisons among data set guides, all GLIS

[1] Work performed under U.S. Geological Surveycontract 1434-92-C-40004

guides have a similar table of contents, as follows:

Background
Extent of coverage
Data characteristics
  Spatial resolution
  Temporal resolution
  Data organization
Data availability
  Procedures for obtaining data
  Products and services
Applications and related data sets
References
  Journal articles and study reports
Appendix

Inventory-level metadata
Once a researcher has identified the type of data set that meets project requirements, it is necessary to do detailed searching of data set contents, or records, for specific data. Within GLIS, these searchable inventories are satellite image holdings containing more than three million digital scenes collected by spacecraft-borne remote sensing devices.

Inventory records, too, must follow a common convention for descriptive metadata. GLIS inventory-level metadata are structured such that every record in every data set has at least six common fields: a unique entity identification, date of acquisition, geographic coordinates, availability of on-line image browse, distributing data center, and the date that the metadata record was last updated. These six fields ensure that a search of data records from several data sets can be done using identical search criteria.

Each data set, however, has unique characteristics that are important to use in searching for specific records. For example, some data sets might have fields that contain a measure of quality, percent cloud cover, or other unique parameters recorded at the time of acquisition. These metadata fields provide information that the researcher may need to make a final selection of data set records. Within GLIS, a typical searchable inventory might have 20 or more such data-set-specific metadata fields, which the researcher can use to make a final selection.

2. Metadata Issues and Concerns

The primary concern that GLIS and all other information systems must deal with is the accuracy and utility of the metadata they contain. Metadata that are too general or vague are of little use to the researcher who needs specific information to determine if a data set would be useful. Conversely, metadata formats that are complex and unrealistically detailed frustrate the information management specialist whose job it is to populate the information system. Metadata formats designed soley by a researcher who has in-depth knowledge of the data set may be too detailed, and, consequently, too difficult for the novice who needs the data.

Information systems, including the metadata they contain and their human-interface tools, need to be designed by a team of experts from various backgrounds. Most importantly, the metadata that forms the heart of such a system must adequately describe data holdings without making the system too difficult to maintain or to use. The single most difficult task in maintaining GLIS has been in building partnerships with the researchers who have developed the data sets. These researchers must also be willing to help document the data collection. They are the ones who have a unique perspective on the characteristics and utility of the data they have developed. With the assistance of experienced information management specialists, these data can be described and formatted for the benefit of others who might have research needs that these data can fulfill.

Within GLIS, development of all three levels of metadata (directory, guide, and inventory) is done in partnership with an expert who either knows the data intimately through past use or is the developer of the data set. All GLIS DIF's, guides, and inventory schemas are formally reviewed and approved by technical experts and information scientists before being incorporated into the GLIS system. Any other approach to developing metadata would not only lack scientific soundness, but might present unnecessary frustrations to researchers trying to use the inquiry system. GLIS and other similar information systems will measure their success not only by how much their systems help the number of users who utilize it as a tool to locate data, but also by the number of researchers who recognize the value of online query and who are willing to contribute data sets and the metadata that describe them.

Scientific Data Management The Role of Metadata

Ian Newman
The GENIE Project[1] ,
Department of Computer Studies,
Loughborough University,
Loughborough, Leics. LE11 3TU UK

Abstract:

This paper presents an amalgam of: experiences gained through talking to UK Data Centres & Environmental researchers in the first phase of the GENIE project; the comments and observations that were made by the participants at SDM92. It reviews the methods of data management in the environmental sciences and the joint roles of metadata as a means of locating relevant data and as a means of turning 'data' into 'information'. Finally, it presents a comparison of the use of metadata in different fields (environmental science, pure science, social science and business).

1 Introduction (The UK scene & the GENIE Project)

In 1991, a working party established by a number of government and research agencies in the UK reported on the provision of data to support researchers into Global Environmental Change (GEC). They concluded that a very large amount of potentially useful data already existed but that this resource was badly under-used. To overcome the problem they recommended that a UK GEC Data Network Facility be established. This was intended to provide GEC researchers with a means of identifying data that may be of relevance to their research and of obtaining access to the data once they were identified. One of the research agencies (the Economic and Social Research Committee) made funds available to establish such a Facility and, after an open tender procedure to award the contract, work actually commenced in April 1992.

The project to fulfil the contract is known as GENIE (Global Environmental Network for Information Exchange). An experimental service for users at two or three sites is planned to be operational by April 1993. This service will be extended to cover users at about forty sites a year later. The two initial tasks for staff associated with the project have been to collect information from existing (UK) GEC Data Centres on the way their data are currently managed and to finalise the structure of a suitable metadata management system.

The remainder of the paper examines the management of data and the role of metadata in the environmental sciences primarily from the perspective of UK current practice but enriched by feedback from the comments of the US practitioners at the SDM92 workshop. A brief comparison is offered with the data management procedures used by other holders and users of data.

2 Remote Sensed Data

Much of the data in the environmental sciences is provided as products derived from remote sensed images. The raw data are, typically, captured and recorded automatically. One or more sensors on a satellite are used to collect reflected signals which are communicated to a ground station that carries out some elementary processing of the data then records and stores them. The only control that is exercised up to this point is whether to operate the sensors

[1] The support of the Economic and Social Research Council (UK) is gratefully acknowledged. Work on metadata is being carried out as part of the programme of the ESRC UK Global Environmental Change Data Network Facility

and/or to record a particular image. The next stage is to process the raw data to provide 'more useful' information for users. The processing algorithms which are used are designed to deliver 'information' which will be meaningful to some particular group of users and which can be used for a specific purpose (e.g. assessing vegetation coverage in tropical forest areas or monitoring 'set aside' land by the CEC agricultural department). It is these derived products which are normally of interest to environmental scientists although some of them are interested in the algorithms that can be used to derive particular 'information' from the raw data (e.g. the validity and sensitivity of the algorithms and the 'quality' of the derived data - its usefulness for the purpose for which it has been derived).

The raw image that is stored at the ground station is merely a collection of numbers. On its own it has no value at all. Its value as information lies wholly in the metadata that is associated, either explicitly, or implicitly, with the image. This gives the satellite identification and satellite orbit information, the time and date of collection of the data and the sensor characteristics for each of the recorded channels of data (if the ground station only records information from one satellite then only the time and date needs to be stored and even this can be done implicitly using the name or physical location of the image i.e. its 'position' in a sequence of stored images). The time and date, together with the satellite orbit information and satellite attitude/sensor direction information, can be used to derive a spatial location for the image (a set of base coordinates and extents). The sensor characteristics (calibration - signal strengths that correspond to each data value and their expected variation with time - and sensitivity) can be used with the date and time information to determine the likely bounds for the actual signal that the sensor was detecting. The satellite orbit information also gives the height of the satellite which can be used with the position information to determine the height above the ground and thus to estimate the strength of the signal at ground level for 'passive' systems. For 'active' systems where a signal is transmitted from the satellite and the reflected signal is collected, the time taken, when compared with the height, indicates whether the reflection actually came from the ground or from the atmosphere (clouds).

Algorithms that produce derived products use some or all of the metadata to produce the refined image. Sensitivity information (giving the bounds for the expected variance in the results) can be used to derive the 'quality' of the derived image for the desired purpose. However, a product which has 'low' quality for one purpose may indicate that the raw data could be used effectively for a different purpose. As an example, if a scientist were interested in ground coverage but there was a high density of cloud then the derivation should indicate that the result was of low quality for this purpose. A meteorologist interested in cloud formation and structure, on the other hand, could find a significant amount of useful information in the same raw image and may be able to use a different derivation algorithm to deliver a high quality product from the same raw data for this different purpose.

The derived products thus also need extensive metadata if they are to be interpreted accurately. The raw images from which they are derived are clearly an essential starting point (i.e. the satellite, time, date, spatial coverage, sensor data used, and sensor calibration information used). In addition, the algorithm that was used to carry out the derivation, with all its parameters, would need to be specified (or, as a surrogate, the date on which the derivation was produced and the people producing it). For academic purposes (i.e. for correctly attributing the source of information) the name of the people who generated the derivation algorithm, and any academic references, would also be useful.

3 Other Data

Although much of the data that is used in the environmental sciences in both the UK and the US derives from remote sensed images not by any means all of it does. For many years prior to the arrival of airborne/satellite surveys, 'surface', 'sub-surface' and atmospheric information has been collected and recorded. As an example, the British Geological Survey

(BGS) has a collection of rock and fossil samples going back to the mid-nineteenth century and all organisations carrying out boring operations in the UK still have, by law, to provide core samples to the BGS. Similarly, the British Oceanographic Data Centre (BODC) follows the example of its predecessor organisations by collecting data (current speeds, salinity at different depths, sea temperature) about the oceans obtained by ships and by tethered sensors. Furthermore, there is a growing need to provide confirmatory information to support and help evaluate remote sensed observations (ground truthing), and very substantial ground based sensor systems exist for monitoring both the earth and the atmosphere.

In these cases too the metadata is at least as important as the data. Neither a rock sample nor a measurement of salinity are of much use by themselves. For most purposes it is necessary to know where (in three dimensions) and when the information was collected (although the BGS have, on occasions, had a truck load of samples delivered with no accompanying metadata). In the case of the salinity measurement it is also necessary to know that it is salinity that is being measured (not temperature or current velocity which are also just numbers once they get in a computer) and the units which are being used. If the possibility of human error (either deliberate or accidental) is to be allowed for then the name of the collector could be an important piece of metadata, together with the name of the person recording the information and the date it was recorded.

Most of the above comments were made either from the perspective of a primary user (typically a Principle Investigator or PI) or, more usually, from the perspective of someone in a Data Centre with a 'good feel' for the subject, the experimental set up and the requirements/interests of the PI. The participants at SDM92 made it clear that, if other people were to use the data (or if PIs were to use data that they did not collect personally), metadata was also needed on the background to the data collection exercise (e.g. why was the data collected; what 'special events' occured during the experiment that the PI would have put in a notebook).

4 Metadata and Data - the Role of Metadata

Repeatedly in both the UK discussions and at SDM92, it was noted that the distinction between data and metadata is not clear cut. For example, there is a difference between 'one off' measurements and a 'series'. In the former the spatial location and time of collection information is normally part of the data because the data is not part of a systematic collection (e.g. fossil data, data collected from ships). In the latter, it is a part of the metadata. Thus, there is no need to record the spatial location with every measurement taken from an anchored current speed recorder nor is there a need to record the time with every reading if readings are recorded automatically every hour, say. In these cases a record of the location, the start time and the frequency are part of the metadata together with conversion tables for the sensor and information about the reliability of the sensor (e.g. reading drift; how missed or spurious additional readings are to be detected/recognised).

Metadata, as discussed above, would be used to interpret data (i.e. to turn data into information) for a particular group of users for a particular purpose. At SDM92 this was split into several parts:

- metadata for the PI (technical information on the instruments, information that would have gone in the experimental notebook);
- metadata for 'secondary' research users (a superset of the first category, what was the purpose of the data collection [why was the data collected], what assumptions were made in designing the data collection method [and in performing any subsequent processing], what data have been stored; how can they be accessed, when were they collected, where were they collected, what time period(s) and geographical area(s) do they relate/refer to, how much do they cost to access, what are the errors/limitations in the data (accuracy), what processing methods can legitimately be applied, who collected the data, who has used the data and with what results, how frequently were

observations made, what sampling techniques were used, what spatial aggregations were used, what were the units of measurement, to what precision were measurements taken);

- metadata for data managers (where a dataset is located - physically and/or logically, how it is stored, how it can be recovered, how much storage space does it occupies what software is needed to manage &/or display it);
- metadata for policy makers (a summary of the key issues with pictorial illustrations, probably drawn from several experimental sources).

A second need that was highlighted by the UK initiative, but was also recognised at SDM92 was for metadata to help users locate the information that they require. In this case the users are secondary researchers, who may be seeking different insights from the data themselves, or who could be investigating the methods used to collect or analyse the data, or the experimental design or ...). The only certainty is that they are unlikely to have the same interests, needs or views as the people who stored the data and provided the metadata in the first place. A further likelihood is that they will use different terminology from the originators.

Metadata which helps a user to locate relevant datasets can be thought of like publicity material. If it did not exist then potential users would not be able to access the data themselves. However, if there is no clear distinction between data and metadata there is even less distinction between the metadata required to locate a dataset and the metadata required to use it (e.g. the spatial area in which measurements were taken could/would be used for both purposes).

5 The Availability of Metadata in the UK

Many users already access data from sources known within their discipline. Several of the UK environmental Data Centres are providing, or planning to provide, their users with metadata services to help them locate useful information. In the most successful cases, they are already providing users with linked metadata and data products (e.g. BODC provide a floppy disc containing a program and data which can be used on PCs to display information about in-shore waters taken from a number of datasets).

Furthermore, some UK users already recognise the more general need to locate useful data from other disciplines. As noted in the introduction, once it is operational, the GENIE system is intended to provide this type of access to metadata and data for UK GEC researchers. However, in the short term, a number of UK users have found, and started to access, the Master Directory in the US while others are accessing its twin (the PID at ESA in Italy).

Although there is no clear cut policy in the UK, most data and almost all metadata is, in practice, free at the point of use although it can be extremely difficult to gain access to the required data (which was the starting point for the GENIE project). However, there is, currently, some pressures for data centres to recoup the costs of storing and managing data either directly from the users or by showing their funding agencies that the data are being used. There is also a requirement for commercial data suppliers to be given a more level playing field in which to operate although this is, to some extent, balanced by a need to minimise the costs of research by keeping data free to researchers. In either case it seems that metadata will continue to be free as a means of publicising available products and of allowing the underlying data to be identified and requested (if no-one knows data exist then they will not be used). On the other hand, there may be pressure for the metadata that is needed to supplement the data and to assist in their interpretation to be 'paid for' in the same fashion as the data they relate to.

6 Metadata in Other Disciplines

The use of metadata in the environmental sciences can be compared with its use in other disciplines by examining six aspects which affect data management:

- the variety and scale of data being managed: environmental data is both large and varied, though some business systems probably have as much data and social science data may be more varied;
- the medium on which data is held (environmental data is still held on a variety of media, this is also true in other sciences but may be less true in business);
- the structure of the data being described: most environmental data, in common with most business data and most science data, tends to be highly structured and meaning is associated with position in the structure; Social Science data may be less highly structured;
- the availability and accessibility of the data: most environmental data is being made available to a wide range of people, the emphasis is on making it more accessible; scientific data is generally less accessible; business data is generally made deliberately inaccessible;
- Privacy and security: environmental scientists are being encouraged to make their data readily available to everyone, to a lesser extent this also applies to both 'pure' and social scientists; business data is generally kept private and a much higher level of security is imposed;
- Quality (accuracy and usefulness) & cost (this is linked to both accessibility and privacy): good quality data which is accurate and of the required precision is at a premium in all subject areas (though data that would be of good quality for all the purposes for which it might be used possibly cannot exist). Cost varies greatly, though all sectors inevitably wish to keep their outlay on metadata and data to a minimum. There is a policy in the environmental sciences in the US that the metadata should be free at the point of use if it was collected or stored using federal funding.

Annotative Metadata in Scientific Applications

by
Lois M.L. Delcambre and David Maier
Computer Science and Engineering Department
Oregon Graduate Institute of Science and Technology
19600 NW von Neumann Drive
Beaverton OR 97006-1999
lmd@cse.ogi.edu   maier@cse.ogi.edu
(503) 690-1689 (503) 690-1154

1. Introduction

Scientific applications exhibit a number of features that are clearly distinct from traditional or even object-oriented database systems. This paper concentrates on the various types of metadata in scientific applications. A number of other challenging issues and topics associated with scientific data management (e.g., data volume, data archiving, visualization, data dredging) are outside the scope of this paper.

The definition and management of metadata has always been an implicit part of database systems, even from the early work on the hierarchical and network models. Such metadata is comprised of the schema and related items, often stored in a data dictionary. Characteristics of this type of metadata are that it:

(1) defines the name and format of the fields at the instance level,

(2) exists before any of the instance level data (e.g., tuples) can appear, and

(3) provides metadata at the collection of data level (e.g., relations or classes) as opposed to the instance level.

Most researchers addressing scientific data management acknowledge that there is a need for additional metadata that allows the application scientists to describe or annotate their observations, findings, readings, conclusions, etc.

We distinguish two types of metadata: *denotative metadata* corresponding to the traditional metadata contained in a schema and *annotative metadata* to capture the application-specific description of data [D92]. Denotative metadata is metadata that describes the structure and semantics of base data. Annotative metadata is metadata the describes the nature, source, location, quality, etc. of base data. Shoshani [S92] makes a similar distinction but uses the terms *structural* and *descriptive* metadata, respectively. As an example, a sensor reading may provide temperature as the base data but the annotative metadata may describe the position of the sensor, the time of the reading, the settings of the instrument, relevant circumstantial information (e.g., that a solar eclipse was in progress), etc. As the raw data is filtered, analyzed, and summarized, additional annotation may be captured, e.g., to describe the quality of the reading based on statistical or other type of analysis.

The focus of this paper is annotative metadata. The next section provides an initial characterization of annotative metadata along several dimensions. The following section presents an instance level annotation facility originally developed for the design of satellites that is currently being applied to the management of scientific data. The paper concludes with a brief discussion of the challenges associated with the support of metadata in scientific applications.

2. Types of Annotative Metadata in Environmental Science

Environmental sciences almost always involves the observation of the Earth. The type, frequency, and granularity of the observations vary widely but the basic data includes observations, gathered over time, normally oriented to the appropriate position on or above the globe. The annotative metadata for environmental science thus usually provides additional descriptive information concerning this time-sequenced, spatially-oriented observation data.

Annotative metadata generally provides further description for what we call base data. However, the annotation-of relationship may differ for various users. It is quite clear that both primary data and annotative metadata are important in scientific applications and that both can be the target of user queries.

One way to characterize annotative metadata is based on the *level* of the primary data that the annotation is associated with. At the lowest level, annotation may be directly attached to the individual scientific observation. As an example, the time, location, sensor type, and quality assessment of an observation might be captured and associated directly with the observation. We refer to this as *instance-level* annotative metadata. As data is collected into sets, series, or other higher level objects, it may be desirable to associate annotative metadata at any or all of these aggregated levels using *collection-level* annotative metadata. Collections may occurs at an arbitrary number of levels (e.g., sets of sets) so this term actually refers to a number of levels.

Another way to characterize annotative metadata is according to its intended *use*. One use of annotative metadata is to assist with the selection and location of scientific data. There are two different aspects of location. First, the location (i.e., time and space) associated with the observed, scientific data is often used to select environmental data. Second, the massive and globally distributed nature of scientific data requires that the physical location of files and archives be ascertained. Both aspects of location can be described using annotative metadata.

Another use of metadata is to describe the quality assessment of the data. Each raw observation is often evaluated and the metadata can describe the precision, accuracy, completeness (e.g., whether the value is missing), confidence (e.g., whether the value is an outlier), and so forth.

A third use for metadata is to capture the results of analyzing or interpreting the data. The environmental scientists ultimately synthesize the observational data to identify physical phenomena and to reach scientific conclusions. Such results must be captured within the scientific data and must be directly related to the proper context, i.e., must be attached to the proper observational data.

Metadata can support all of these uses: location, quality assessment, and analysis and interpretation. Furthermore, these types of metadata based on intended use may occur at any or all levels, the instance level as well as various collection levels.

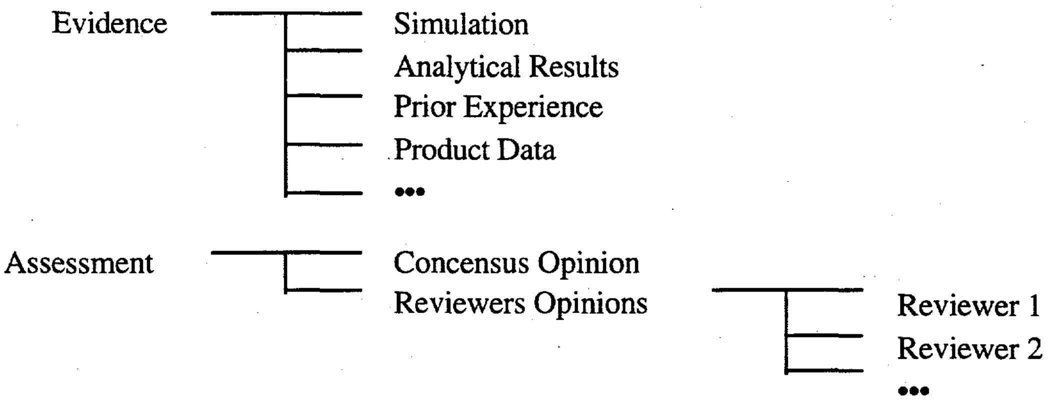
The final method for characterizing metadata presented in this paper is based on the the metadata used to implicitly represent objects or entities of the application. As an example, the sensor type used to capture observational data may be recorded as metadata about the observation. A conceptually more direct approach would be to model sensors explicitly, including such attributes as sensor type, model and serial number. Traditional databases present the same choice when a schema designer chooses to represent a given aspect of an application implicitly as an attribute (e.g., of some other object or entity) or explicitly as an entity in its own right.

3. A Flexible Annotation Facility

AutoCRAT, the Automated Constraint Refinement and Assessment Tool, is a tool for the articulation and management of design constraints [DS92, SDG92, SDD91]. AutoCRAT implements a generic design framework based on the Theory of Plausible Design [AD87]. The tool was developed to help reduce the cost of satellite systems by providing support for the design process early in the product or system life cycle. AutoCRAT supports the statement and top-down refinement and bottom-up validation of design constraints.

Early in the development of AutoCRAT, it became clear that the designer would like to capture a number of annotative entries, in addition to the actual *constraint statement,* including such things as: the *source* of the constraint, the *justification* for including the constraint, and the *author* of the constraint. The assessment of designs is particularly important in the space systems domain where, early in the life cycle, they rely almost entirely on the assessment of various review teams (e.g., for the preliminary design review). Each reviewer's opinion as well as the consensus opinion about each constraint can be recorded in AutoCRAT as a form of annotation.

The annotation fields in AutoCRAT provide the basic attributes for constraints as well as for all types of annotation. Consider the example shown in the figure below with two, top-level annotation fields for constraints: Evidence and Assessment. Each of these annotation fields can be associated with each and every constraint. Each annotation can be further refined. For example, the top-level annotation field named Evidence is further refined according to the types of evidence: Simulation, Analytical Results, Prior Experience (which in space jargon means that it has successfully "flown before"), and Product Data (for commercial products). The semantics of sub-annotation require that the constraints associated with the Product Data annotation are associated with the Evidence annotation and, conversely, the constraints associated with the Product Data annotation are a subset of the constraints associated with the Evidence annotation. The annotation structure provides hierarchical attributes for constraints where each node in the hierarchy can support a textual annotation entry. The sub-annotation fields can be viewed as annotations that provide further refinement of an annotation. In practice, AutoCRAT users define many more annotation fields in addition to those shown below.



Unique features incorporated into the AutoCRAT annotation facility are:

1. Annotation fields can be added at any time. Thus the "schema" that defines the annotation fields can be modified dynamically, while AutoCRAT is running and while the product is under design.
2. Each annotation entry is stored in an underlying relational DBMS as a variable length text field of up to 32k characters.
3. Each constraint can have zero or more annotation entries (up to one for each annotation field defined). The structure of the annotation provides a taxonomic hierarchy for the application.
4. The annotation structure is shown graphically within AutoCRAT and serves as a point and click query facility. Clicking on an annotation name returns all constraints that currently have an annotation entry for that annotation field. Annotation fields can also be combined using AND, OR, and NOT to provide a user-friendly query language.
5. The semantics of annotation and sub-annotation are supported by the AutoCRAT software. Thus the annotation facility is dynamic, flexible but also strictly managed according to the semantics of annotation.

The limitations of the annotation facility supported in AutoCRAT for scientific annotation are listed here.

1. All annotation entries are arbitrary-length text fields. There is no possibility for domain or field type definitions as in a traditional schema.
2. AutoCRAT is a monolithic system; the only object of interest is the constraint. Thus all other objects in the application are represented only implicitly, through the annotation. Scientific and most other applications require support for multiple object types.
3. AutoCRAT is also a flat system; annotation is supported only at the instance level.

In AutoCRAT, the constraint serves as the primary object of interest and all annotations (of constraints) are considered metadata for the constraints. However, in scientific applications, constraints may serve as a form of metadata.

4. Discussion

The utility of annotative metadata has been clearly demonstrated in AutoCRAT and is intuitively required in scientific applications, as well. In order to merge the AutoCRAT annotation facility with an Object-Oriented DBMS it may be desirable to use the annotation facility to support all of the simple-valued attributes associated with objects. This suggests that the top-level annotation fields (in AutoCRAT terms) are, in fact, attributes of an object type. Sub-annotation fields are then annotative entries for the parent annotation field. Note that the aggregation or property links that directly interconnect abstract objects in an object-oriented schema are considered to be distinct from the simple-valued attributes mentioned above.

Browsing through and querying scientific data is extremely important and can benefit from uniform access to the primary data, annotative metadata and denotative metadata (including the denotative definition of the annotation structure). The notion of a self-describing database originally proposed for a relational context [MR86, MR85] would provide a uniform interface to denotative metadata as well as annotative and regular data. This implies that scientific users could ask queries like: "what data fields are captured for the solar flare experiments conducted in the Mojave desert?", "which sensor readings include annotative entries for *deviations in normal settings*?", etc.

Some of the challenges associated with scientific metadata are: providing the capability for annotation associated with any entity, at any level, and for any purpose according to the needs of the application; providing uniform access to the annotative and denotative metadata to promote the use of the data; propagating and summarizing the metadata as the data is aggregated during analysis; implementing a system rich with annotation in such a massive and distributed context; and supporting discrepancies in terminology and mapping terminology from one application or database to another. This final challenge can be addressed through the development of standardized terminology and through the development of technology to map from one set of terms or annotation structure to another.

References

[AD87] Aguero, U. and Dasgupta, S., "A plausibility-driven approach to computer architecture design", *Communications of the ACM*, 30(11):922-932, Nov. 1987.

[D92] DeVaney, D.M., Personal Communication, August 1992.

[DS92] Delcambre, L. and Schwartz, D., "AutoCRAT: Automated Support for Design in a Concurrent Engineering Environment", *Proc. of the*

European Joint conference on Engineering Systems Design and Analysis, June 1992.

[MR85] Mark, L. and Roussopoulos, "The New Database Architecture Framework - A Progress Report", *Information Systems: Theoretical and Formal Aspects*, Sernadas, A., Bubenko, J., Olive, A. (eds.), North-Holland, New York, 1985, pp. 3-18.

[MR86] Mark, L. and Roussopoulos, N., "Metadata Management", *IEEE Computer*, Vol. 19, No. 12, Dec. 1986, pp. 26-36.

[S92] Shoshani, A., "Metadata Management for Scientific Applications", Presentation at the *CESDIS Workshop on Scientific Data Management*, College Park, Maryland, Sept. 1992.

[SDD91] Schwartz, D., Delcambre, L., and Dasgupta, S., "AutoCRAT: An automated design tool for constraint refinement", *Proc. of the Simulation MultiConference, Artificial Intelligence and Simulation Conference*, New Orleans, Louisiana, April 1991.

[SDG92] Schwartz, D., Delcambre, L., and Gillam, G., "AutoCRAT Templates for Design Knowledge Capture", *Proc. of the International Space Year Conference on Earth and Space Science Information Systems,* Los Angeles, California, Feb. 1992.

Metadata Standards and Concepts for Interdisciplinary Scientific Data Systems

Donald E. Strebel
Versar, Inc.
9200 Rumsey Rd.
Columbia, Md 21045

strebel@ltp.gsfc.nasa.gov

and

Blanche W. Meeson
Code 902.2
NASA/GSFC
Greenbelt, MD 20771

meeson@pldsg3.gsfc.nasa.gov

INTRODUCTION

We have developed standards and a conceptual framework for scientific data system metadata based on our experiences supporting focussed field experiments, long-term data archiving, and data publication. In each area, we have handled a broad range of data types including those from satellites, aircraft, ground based field instruments, laboratory instruments, and outputs from simulation models.

We draw a distinction between the types of metadata and the functions of metadata. In an information system, metadata simultaneously serve three functions: data management, data access, and data analysis. Each of these functions has a different set of users with different requirements. We also define several types of metadata, including: auxillary documents and analog information, data set summaries, data set detailed descriptions, descriptions of individual data granules, and descriptions of individual elements that comprise the data granules. Although these types are sometimes viewed as forming a hierarchy, that is not the best way to conceptualize the relations between them. From a science user's point of view, we particularly like drawing an analogy between these metadata types and the components of a scientific paper.

As described above, we thus view metadata requirements as falling into a matrix of 5 types by 3 functions. Each type of metadata is not necessarily used for each function, but some types are used for all functions. We do not believe that this classification is unnecessarily complex: the matrix is the key to analyzing metadata requirements and developing standards and implementations which resolve the needs of all users.

Standards and guidelines are required for each type of metadata, and they must be consistent between types. In addition, there must be methods for evolution of standards and resolution of conflicts that may arise due to the multiple functions of each metadata type. To the extent possible, these methods must be based on consensus between the different user groups (e.g system developers, information management staff, and scientist users). However, there is also need for a tie-breaking rule. One we prefer is that the science user functions (data access and data analysis) take precedence over system functions, provided that this does not compromise the information system. In general, information management staff has the familiarity with the system and the technical capability to adapt to and work around user needs. The system will not be successful, however, unless the science user needs are met.

DEFINITIONS

To describe our approach in more detail, we first need to draw attention to a few definitions. Data systems and metadata are frequently described using several terms which our experience has shown have a wide variety of meanings and hence lead to confusion if not clearly defined initially.

By DATA SET we mean an aggregation of related data that have been collected either all by the same instrument, such as the AVHRR instrument on the NOAA series of satellites, OR has been collected as part of an experiment following a single coordinated plan, such as data from a handful of sites collected at monthly intervals over 2 years. With this definition, a single AVHRR image would not comprise a data set, but rather a single instance within the entire AVHRR data set, which is the collection of all such instances.

Another relatively new term which we will use is GRANULE. This term refers to a single instance of data such as the AVHRR image just mentioned or to a scientifically meaningful grouping of point data, for example, a day's worth of air temperature data at a single site.

Associated with a data object such as a data set or a granule will be a set of metadata items called DESCRIPTORS. For example, a data set containing observation date, time, and air temperature would be associated, at the most detailed level, with descriptors giving the characteristics of these three elements.

A rigorous application of these defintions is not always the most logical from the point of view of providing documentation to a user. An example might be a suite of instruments mounted on a common platform measuring related information, e.g. micrometeorolgical data (wet bulb temperature, dry bulb temperature, air pressure, wind velocity, and net radiation). Loosely speaking, a scientist would normally be interested in the "micrometeorological data set" and expect granules to contain all of the related measurements for a given day and site. It is often more practical to stretch the definitions to encompass such aggregations than to force an inappropriate organization of the metadata.

CONCEPTUAL MODEL

Our framework of 5 metadata types and 3 metadata functions has developed over several years.

It has evolved out of our specific experience with metadata and metadata standards used by several data systems, including two that we have developed and operated (the Pilot Land Data System [PLDS] and the FIFE Information System [FIS]). To be specific, our concept of metadata for data management has its roots in NASA's Climate Data System, but has evolved from that through our experience with the NASA Master Directory, PLDS, and the FIS. The concept of metadata to support access to and interactive presentation of data to scientific users is founded in our experience with the PLDS. And finally, the metadata to support data analysis has its beginnings in the work of the Climate and the Planetary Data Systems and has matured through our work with the FIFE Information System.

The 5 types of metadata required for these three functions (data management, data access, and data analysis) are most easily described to the scientific user community by comparing them to the components of a scientific paper. Using this approach, data set summaries are analogous to the abstract of a scientific paper. The summary captures the overall focus and content of the data set, thus it addresses the same level of detail and basic content as an abstract.

Documentation Type/Function Matrix

Type of Information	Use Function		
	Data Management	**Data Access**	**Data Analysis**
Fundamental (Dictionary)	Organize the data	Select the data items of interest, determine output formats, etc.	Infrequent use as long as information also in detailed docs.
Detailed (Catalog/Guide)	Record ancillary information about the data. Support user requests	Obtain a technically meaningful descripiton of the data; determine whether data set is useful.	Primary guide to using the data
Summarized (Directory)	Index and track data sets. Record brief data set description.	Identify data sets which may be of interest.	Infrequent use.
Auxillary (User Guides)	Store off-line information about data	Infrequent use.	Reference for detailed questions about the data
Access (Inventory)	Index and track the data items in a data set. Prepare special distribution reuqests.	Determine whether particular subsets of data items are available	Infrequent use.

The detailed data set descriptions can then be viewed as the text or body of the paper. This is where data collection and analysis methods are described, as well as the scientific use of the data. The captions and text descriptions of the figures and tables within a scientific paper are a bit like the descriptions of individual data granules (the figures and tables are, in a sense, the data granules). The descriptions of individual elements that comprise the granules are equivalent to a glossary of the specialized scientific terms used in a paper (sometimes included in a paper, or often by reference to standard definitional materials in the field).

Finally, auxillary documents and analog information are most like the references cited in a paper, since these materials contain specific pieces of data or extensive descriptions that are not immediately essential to the use of the data, but are necessary for a full understanding of the context of the data collection and analysis.

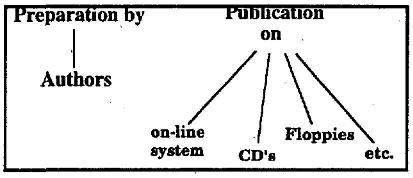
STANDARDS, GUIDELINES

We have developed a wide variety of standards and guidelines to implement these concepts in practical situations. The need for these standards and guidelines arose out of a need to support data publication and to establish consistency across different sites within a on-line distributed information system. Initially, we started with the conventions used by the Planetary Data System and then modified them from our own experience building and operating scientific data systems. Standards were needed for all of the metadata that were presented to the scientific user and for much of the metadata that were used to drive the presentation of information to the users.

Motivation for Standards and Guidelines

- **Needed to Support:**

  † **Data Publication**
  (Management)



  † **Data Distribution** **Access & Analysis**

- **Needed to Establish: Consistency, Quality, Ease of use, Expectations, Comprehension**
- **Needed to Streamline and simplify**

During the development of these stanards we used the these guiding principles to help reduce the effort involved and to resolve conflicts.

Principles Guiding Development of Standards

- **Methodology**
  - † **Use existing Stds/Guidelines**
  - † **Grow existing Stds where necessary**
  - † **New Stds/Guidelines only where others don't exist**
  - † **New Stds/Guidelines driven by scientific need**
  - † **Assume continuous change**
  - † **Stds/Guidelines developed by scientific community with Data System assistance**
  - † **Consensus between developers, information managers, and scientists**
- **Resolution of conflicts**
  - † **Science user functions take precedence (data analysis, data access)**

A comprehensive set of descriptors is required for each and every data set, including a standard core set that is applied to all data sets in an information system. Guidelines and rules are needed for the use of descriptors, so that data sets with common elements use the same descriptors for the common elements. Moreover, every descriptor needed a standard list of metadata which was required to define it, thereby ensuring that its meaning was clearly distinct from any other descriptor. No single descriptor should have more than one meaning and descriptors for a data set should be common across a distributed information system. To help implement these rules, standards and guidelines for names and abbreviations used in descriptors were also created.

EXPERIENCE AND EXAMPLES

As mentioned, our approach to documentation was developed through application to handling data from several experiments and archiving efforts. In particular, the Pilot Land Data System was a testbed for developing data system services for the NASA Land Science community. This initially involved building an online inventory system, then expanded to support of active field experiment data bases and archiving and publication of final data sets from field experiments and other research projects.

The specific objective of PLDS was to provide a distributed data and information management service to the land science research community by archiving, retrieving, and transferring data. As PLDS was designed and built, many documenation concepts evolved and were tested, driven by specific application problems. Guidance was provided by the science community through an active Science Working Group interacting with PLDS management and the development staff. After the baseline PLDS was established, further development, particularly of detailed data set descriptions, was required to archive and publish the data from the First ISLSCP (International Satellite Land Surface Climatology Project) Field Experiment, usually called FIFE. The final configuration of the system was a suite of on-line and off-line services that foster and enable capabilities for the community.

When developing and then operating early versions of the on-line data ordering and distribution service, it became obvious that clearly defined and agreed upon standards were essential to the usefulness of this service to the scientific community. It was critical to assure the uniformity and consistency of the information from one science support site to another, as well as from data set to data set. The on-line presentation of the data also had to be easily/readily modifiable at little to no financial cost. This led us to several of our standards and guildelines, including one which defined the metadata that described every data element. The contents of such a data element descriptor are:

Name	Descriptive name for the data element
Sequence	The order of appearance
Type	Data type for this element
Size	The maximum size of this element
Definition/Description	Scientific definition of the value
UOM	The units of measure of the value
Mandatory indicator	Is this element mandatory to maintain data base consistency ?
Format	Specific format of the values
Range or list of	Min and Max values or valid entries descrete entry list
Key field indicator	Is this element part of the primary key for the table ?
Keywords	Cross-referencing keywords
Report parameters	Column name used for output, default output column width, etc.

Note that all three metadata functions are addressed: management (sequence, type, size, mandatory indicator, key field indicator), access (name, format, range and valid entries, keywords, report parameters), and analysis (name, definition, UOM, format).

As the on-line system was constructed, and more importantly during the development and operation of the CD-ROM data publication service, our focus shifted to the detailed metadata describing the data collector, data collection and data analysis procedures, data usage, etc. This information is essential for the long-term utility of the data to the scientific community. Moreover, for this metadata to be useful, the contents must be standardized so that the scientific community can rely upon the information provided with a data set. Other requirements are that there must be procedures to assure the accuracy of the information and to insure that it is distributed with the scientific data.

The detailed data set description contents that we advocate is similar to that used by the NASA Climate Data System. Several significant modifications and enhancements have been made, however, to accomodate non-satellite data obtained from field experiments such as FIFE. We have also reordered the topics to correspond to the general outline of a scientific paper, in keeping with our conceptual analogy and our desire to create documents that are intutively accessible to scientist users.

The major sections of our detailed data set descriptions are:

1. Title
2. Investigator(s)
3. Introduction
4. Theory of Measurements
5. Equipment
6. Procedure
7. Observations
8. Data Description
9. Data Manipulations
10. Errors
11. Notes
12. References
13. Data Access
14. Output Products and Availability

Each section has several specific subsections, and in some cases third level paragraphs as well. Contents definitions and examples for each item have been created in the context of documenting the FIFE data collection. The outline is flexible enough to handle a wide variety of data, while being specific enough to allow us to devise effective access tools and to standardize writing and editing procedures. A completed description for a specific data set has a length of roughly 20 single spaced pages of text and is sufficient to provide a stand alone introduction to the data sufficient for an interested but but non-expert scientist to begin working with the data.

The task of publishing a scientific data collection on CD-ROM for permanent (i.e. decades) storage and use has made us accutely aware of the human element of the scientific data documentation process. Obtaining full data set descriptions, in a consistent and usable format, turned out to be a more daunting task than organizing and formatting the data itself. While our *scientific paper* approach seems readily accepted (and in some cases, applauded) by the scientific community, many investigators are reticent to apply their time and resources to the documentation effort for their data sets. It appears to them to be a rather tedious chore. To counter this fairly understandable reaction, the data system staff must participate actively in drawing out the information, assuring its completeness, and preparing it for publication. These activities require scientific expertise on the information management staff, as well as a process of peer review of the documentation. It is currently not easy to get commitments for such peer review, because data set documentation is not recognized as a formal publication and there are few career benefits associated with reviewing it.

One lesson here is that adequate scientific documentation of a principal investigator's data set is an intensive human task: automatic tools can simplify and expedite the task, but they cannot substitute for the basic human effort of gathering the information and editing it into an intelligible form. Metadata systems must therefore be designed with a proper balance and organization of people, hardware, and software.

FUTURE

At this point we have learned how to store and deliver scientifc data, accompanied by appropriate metadata, to users removed from the original investigators in both time and space. Although we have developed systems that allow the user a great deal of flexibility in defining and extracting data of interest, our concept and treatment of metadata is still largely static. That is, the metadata descriptors are associated in a fixed way with the data sets and granules as defined by the original investigators and the information management staff.

To some extent, this is a sensible approach, since the definition of a data set and the provision of the scientific metadata describing it are primarily a scientific responsiblity of the investigatror who collects the data. However, there are often many scientifically valid ways to define, select, and/or organize meaningful subsets of the same collection of data. We believe that by adequately rigorous definition of metadata types and functions, relational concepts can be applied to its organization, and a dynamic metadata system can be devised. That is, as a user interactively defines a data aggregation of interest, different than that constructed by the original investigator(s), the accompanying metadata descriptors would be reorganized to form a coherent description of the selected data.

Where Do We Go from Here?

Goal: To empower scientists not involved with the data collection to make use of the data according to relations they define

Requires	**Current**
Fully dynamic queries (i.e., any relation)	**Limited dynamic queries** Constrained by predefined relations
Dynamic data documentation/aggregation	**Largely static** Limited dynamic ability on value definitions

Research Needed:

Formal Theory of Data/Documentation

Associations, structures and aggregation

Database Structures with Dynamic Relation Capabilities

(Current relational data base management systems are based on static, predefined relations)

The technological capability to construct a dynamic metadata system of this sort is probably currently available. What is lacking is a suitably rigorous defintion of metadata, a formal understanding of metadata relations, and acceptance of uniform metadata standards by the information management and scientific communities. We submit that successfully addressing these issues will make possible highly flexible scientific information systems that will automatically and robustly supply all of the appropriate supporting information required for valid scientific interpretations of the data, regardless of the complexity of a scientist's query.

Conceptual Schemas: Multi-Faceted Tools For Desktop Scientific Experiment Management

Yannis E. Ioannidis[1]
Miron Livny
Computer Sciences Department
University of Wisconsin
Madison, WI 53706
{yannis,miron } @cs.wisc.edu

ABSTRACT

In this paper, we identify some of the fundamental issues that must be addressed in designing a desktop Experiment Management System (EMS). We develop an abstraction of the set of activities performed by scientists throughout the course of an experimental study, and based on that abstraction we propose an EMS architecture that can support all such activities. The proposed EMS architecture is centered around the extensive use of conceptual schemas, which express the structure of information in experimental studies. Schemas are called to play new roles that are not usually found in traditional database systems. We provide a detailed exposition of these new roles and describe certain characteristics that the data model of the EMS must have in order for schemas expressed in it to successfully play these roles. Finally, we present the specifics of our own effort to develop an EMS, focusing on the main features of the of the data model of the system, which we have developed based on the needs of experiment management.

1 . INTRODUCTION

In the past few years, several scientific communities have initiated very ambitious and broad-ranged projects whose goals are to significantly advance the frontiers of knowledge in their disciplines by solving very hard problems that until recently were considered unapproachable. Such efforts are expected to last for many years and will play the role of umbrella projects under which several scientific questions will be investigated. The NASA Eos project and the NIH Human Genome project are two examples of national and international scientific endeavors that belong to this category. The goal of Eos is to collect data about the earth and its atmosphere that will be used by earth scientists for global change research, while the goal of the Human Genome project is to sequence the human DNA and from that understand the nature of genetic diseases. In this paper, we use the term global project to refer to a large scale scientific effort like the ones above. A major component of such projects is the collection of measurements on complex phenomena. Such activities will generate huge amounts of data (sometimes measured in petabytes one petabyte is equal to 10^{15} bytes), which will then be studied by thousands of researchers. Managing this surge of scientific data poses many

[1] Partially supported by the National Science Foundation under Grant IRI-9157368 (PYI Award) and by grants from DEC, HP, and AT&T.

challenges, with which current database technology is unable to deal. Several technical problems need to be solved before Scientific Database Systems can become a reality. An excellent account of these problems together with an overall picture of the major scientific projects that are currently under way is given in the summary of the NSF Workshop on Scientific Database Management [Fren9O].

The widespread availability of the unprecedented collections of data gathered as part of the above projects will generate much scientific activity at the level of individual scientists or small teams of scientists. Smaller projects will be initiated to study a variety of phenomena related to the global projects, using small fractions of the available data. In this paper, we use the term local [2] study to refer to such smaller scale research efforts. Given the scale of such studies, it is desirable that the experiments and the data generated from them be managed directly by the scientists themselves, who will not be experts in database systems. There are no adequate management tools, however, that are natural and intuitive to the nonexpert and offer the desired functionality. Thus, similarly to the large-scale projects, these smaller studies will also suffer from the lack of appropriate technical support.

The above is perceived as a major problem for experiments studies in most scientific disciplines even today. Based on our own experience with experimental computer science [Livn87] and from joint work that we have undertaken with scientists from a wide range of experimental disciplines (biotechnology, genetics, earth and space science, soil sciences, and high-energy physics), experiment and data management have become the bottleneck in such studies. In many cases, the lack of adequate management solutions significantly limits the scale and scope of the experiments. While some scientists store data in hundreds of flat files or, in the best case, under a simple relational database system, most of them still use paper notebooks, which are clearly inadequate tools for extensive experimentation.

There are some technical challenges that are unique to one of the two types of activities mentioned above, i.e., managing the collection and distribution of the primary data for a global project and managing a local experimental study. For example, dealing with large amounts of data is primarily an issue in global projects. On the other hand, supporting nonexpert scientists so that they manage the execution of experiments themselves is only an issue in local studies. Nevertheless, the main sources of many problems are common to both types of activities. Examples include the types of data, the size and complexity of the structure (schema) of the experiments, and the need to provide interfaces for nonexpert scientists to browse through and retrieve data. Solutions to these challenges should be applicable to systems that support either type of activity.

The general theme of this paper is managing local experimental studies. We introduce the term "desktop Experiment Management System" (EMS), to describe a system that supports such activities. Such a system, which includes a Database Management System (DBMS) as one of its components, will be the only tool that a scientist uses to manage his/her experimental studies. It will support the scientist in the design of the study, communicate with the appropriate environments from which the data for the study is collected, and store and manage that data. The operational environment of experimental

[2]" "

studies has the following unique characteristics that place certain demands on what the desired functionality of an EMS is:

(1) Each experimental study goes through several stages that are quite different from each other. To avoid overburdening the scientists, who should not have to be experts in database management, the EMS should provide a uniform interface that can be used in the diverse activities related to all these stages.

(2) In today's scientific laboratories, where experimental studies are conducted without much computerized technical support, communication among collaborating scientists is quite interactive. To facilitate the same mode of communication when computer technology is used, the EMS should provide an efficient and natural user interface that resembles, to the extent possible, the way scientists interact among themselves.

(3) Many experimental studies are in need of generating data in multiple diverse ways and using existing data from multiple sources. The EMS should be capable of communicating with all these heterogeneous information sources and integrating the data that they provide without requiring much detailed knowledge from the scientists.

Providing the above functionality presents many problems to today's technology. These problems are further exasperated by the complexity of the structure of the data and experiments manipulated by the EMS.

In this paper, we identify some of the fundamental issues that must be addressed in designing an EMS so that its goals may be achieved. An important aspect of this work is a proposed EMS architecture that is centered around the extensive use of conceptual schemas, which express the structure of information in experimental studies. Schemas are called to play new roles that are not usually found in traditional database systems. We provide a detailed exposition of these new roles and elaborate on the implications of such schema use. Specifically, we describe certain characteristics that the data model of the EMS must have in order for schemas expressed in it to successfully play these roles. An interesting side result of the above effort is the development of an abstraction of the set of activities performed by experimental scientists throughout the course of a study, on which the details of the proposed EMS architecture are based. Following the above general principles on how to support the management of experiments, we have undertaken an effort to develop a desktop EMS that achieves the desired goals. We present the specifics of our approach in the later part of this paper. In particular, we describe the salient features of the data model that we have developed for the EMS justifying their inclusion in the model by the needs of experiment management. We also discuss a case study where schemas expressed in that model played some of the new roles mentioned above in the context of some scientific experiments.

As a reference point that can be later used to illustrate the various issues raised in the paper, we describe a very simple experimental study. Simulation is being used to model the effect of weather on plant communities. Its input consists of weather parameters, which are humidity and wind speed and direction, and characteristics of a plant community, which are the locations of all plants and the number of leaves, height, and type (e.g., com, wheat) of each plant. Its output is the vegetation temperature, one temperature value for each plant. The simulation itself takes into account the relative placement of the plants and all the physical laws on how each type of plant reacts to the weather conditions based on its environment. An EMS used for this study will allow scientists to design the input and output structure of the experiments, invoke executions of the simulator, store the collected data, and submit queries on the experiment results.

Among all types of schemas, we only deal with conceptual/logical schemas in this paper. Hence, we often use the plain term "schema" instead of the full term "conceptual schema". Also, we imagine that most users of an EMS will be scientists, researchers, or technicians working in a laboratory. For the purposes of this paper, we make no distinctions among the above types of experimentalists, so we use all the above terms (including the term "user") indistinguishably to refer to the generic user of an EMS. Finally, databases containing the data associated with experimental studies are called "experiment databases".

This paper is organized as follows. Section 2 describes a common life-cycle that underlies most experimental studies. Section 3 outlines the functionality that an EMS should provide to its users and proposes an architecture that we have adopted for such a system that we are currently developing. Section 4 identifies some new roles that conceptual schemas are called to play in the context of an EMS. The characteristics that the data model should possess in order for its schemas to play these roles arc also identified in this section. Section 5 discusses the salient features of the Moose data model that we have developed for experiment management. Section 6 contains a brief description of a case study where some of the tools that we have developed for manipulating Moose schemas were used in an experimental study. Finally, Section 7 summarizes our approach for experiment management and discusses the future directions of our work.

2. LIFE-CYCLE OF EXPERIMENTAL STUDIES

To achieve its goals, an EMS will use conceptual schemas for various activities that are important throughout the course of an experimental study. From discussions with scientists from different disciplines, we have concluded that these activities are common to most experimental studies. We use the term life-cycle of an experimental study (or simply experiment life-cycle) to denote the entire set of these activities together with the way scientists iterate over them during such a study. In this section, we describe the different stages of that cycle, so that the details of the different roles of schemas throughout the cycle can be explained later. We should emphasize at this point that the experiment life-cycle that we describe only captures the activities involved in conducting the experiments and not those involved in setting up the appropriate experimentation environments. For example, in the case of simulation studies, it does not capture the programming task of developing the simulator, but it does capture the task of executing the simulator with a specific set of input parameters.

A pictorial abstraction of the experiment life-cycle is shown in Figure 1. It essentially consists of multiple loops traversed by the researcher multiple times in the course of a study. In the figure, the following stages can be identified:

Experiment Design: In this stage, the experimental frame of a study is laid out [Zeig76], that is, the structure of each experiment is defined. The experimental frame determines the variables that will be controlled in the experiments and defines what will be measured as output. For the example of the plants experiment of Section 1, this stage consists of identifying the input and output parameters of the simulator and their relationships based on their semantics. Properly designing the experiments is the most crucial aspect of an experimental study. A satisfactory design is rarely achieved in a single attempt. This process undergoes many iterations, usually interleaved with the execution of some experiments and analysis of the obtained data, before the design reaches its final form.

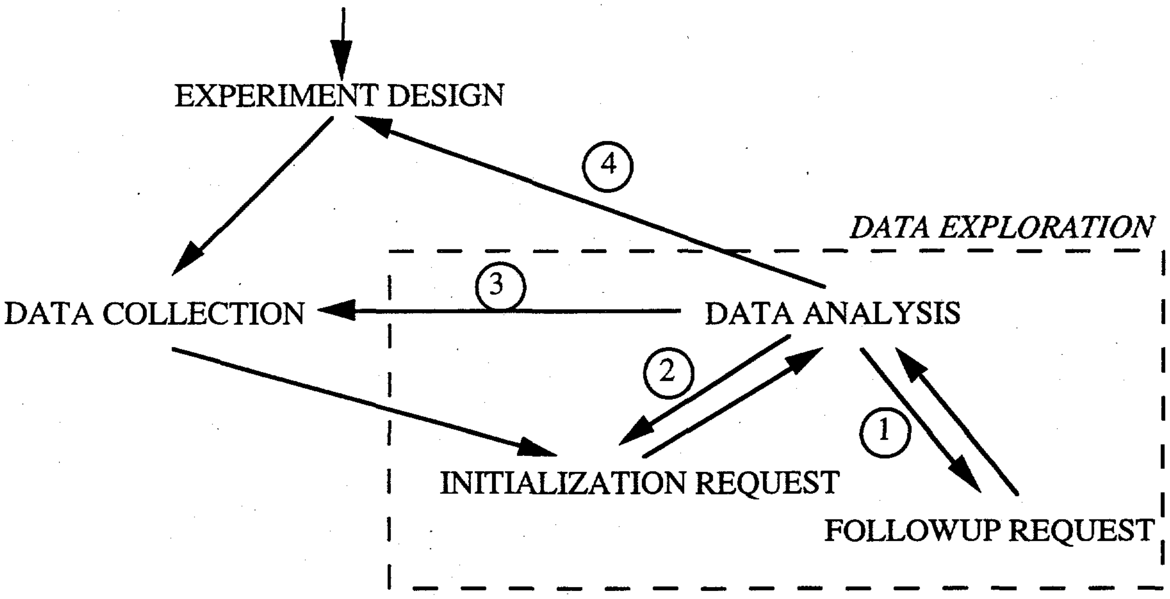


Figure 1: Life-cycle of an experimental study.

Data Collection: In this stage, experiments are actually conducted. The researcher specifies the experiment set up and the precise values of all the input parameters to the experiment, and the relevant output data is then collected. The data can either be distributed to some or all of the scientists involved in the study or it can simply be stored for later use. Simulating a specific plant community given certain values for its characteristics and the weather conditions is an example of an action in this stage for the plants experiment.

Data Exploration: In this stage, the researcher studies the collected data to draw conclusions about the subject of the experiment. As shown in Figure 1, there are three types of actions that the scientist may perform on the data, which are described separately.

(1) Initialization requests: Whenever scientists start to explore a new vain of thought in an experimental study, their first request on the collected data is very similar to a conventional query in traditional database systems. It references all properties of the phenomenon or system under study that are expected to remain unchanged throughout the exploration of the new idea. In principle, such a request needs to deal with the full experimental frame of the study and must include specifications of the values of many parameters, the relationships between several others, and some indication of what should be retrieved. A conceivable initialization request for the plants experiment may be for the final temperatures in corn communities with given plant characteristics and weather conditions when the distance between any two plants is less than I meter. In most cases, due to the amount of information that must be specified, posing such queries is a time consuming process.

(2) Data Analysis: After receiving the requested data, scientists analyze it based on domain specific knowledge that is relevant to the studied phenomenon. Occasionally, the analysis is not based on the data retrieved by the scientists' requests, but on the output of some further processing on it. Such processing is invoked by applying domain specific operators to the data, e.g., measuring the intensity of an image, extracting the statistical properties of a time series, or obtaining the difference between two functions.

(3) Follow-up Requests: Based on the results of the analysis of some obtained data, quite often scientists pose new requests that are very similar to the previous ones, having the answers of the latter as a reference point. This is due to the predominantly exploratory nature of experimental science, which forces scientists to navigate through a multi-dimensional space of parameters that captures the behavior of the observed phenomenon. As an example of a follow-up request, after the above initialization request in the plants experiment, scientists may ask for the same but for distances less than 2 meters. The difference between the two requests is only the value in the selection clause on distances. Follow-up requests represent the most common form of interaction in the course of a study. It is therefore extremely important that such requests be efficiently and conveniently expressed.

If Figure I is seen as a directed graph, then it is clear that all graph nodes except for data analysis have out degree 1, i.e., the successors of the corresponding stages are predetermined. On the other hand, after data analysis the study may move to any of the other stages. Each one of the corresponding arcs closes one of the loops of the life-cycle mentioned earlier. These loops can be totally ordered based on their frequency in the life-cycle, i.e., based on how often the scientist follows the corresponding arc after data analysis. That ordering is indicated in Figure I by the numbers labeling those arcs, where I indicates most frequent and 4 indicates less frequent. For example, it is more likely that a scientist will pose a follow-up request after analyzing some data than that he/she will redesign the experiment.

It is worth noting at this point that the separating line between the data collection and data exploration stages is rather hazy, in the sense that data exploration may involve hidden and not explicitly requested data collection. When a scientist is studying a phenomenon, whether a specific piece of information has already been collected or needs to be collected via an experiment is irrelevant. Thus, some requests in the data exploration stage may generate orders for data collection. For example, consider the initialization request on the plants experiment mentioned above. If corn communities have never been simulated with the given plant characteristics, then simulation may be automatically initiated as a result of this request to obtain some relevant data.

We should also mention that the above life-cycle represents experimental studies that are conducted either by individual scientists or by teams of scientists. In the latter case, most stages of the life-cycle involve communication among the collaborating scientists. This communication can be in the form of actual real-time interactions for decision making (mostly in experiment design and data analysis) or in the form of concurrent actions of multiple scientists, the results of which are later integrated together (mostly in data collection and initialization and follow-up requests).

Having presented the general structure of the life-cycle of an experimental study, we would like to use that cycle to clarify the distinction between the two types of activities mentioned in Section 1, i.e.,. managing and distributing the primary data for a global project and managing a local experimental study. The primary focus of the former activity is in the first two stages of the life-cycle, i.e., experiment design and data collection, with minimal or no interleaving between them. For example, after the initial design stage of measurements of the Eos project, satellites will be launched to start collecting the prescribed data without much interference. Requests for the collected data by interested scientists will be mostly for small subsets of data that are related to a small geographic region, period of time, or specific phenomenon. Although it is possible for a scientist to submit multiple such requests in a given session, especially in a browsing mode, the complex iterations that the full data exploration stage implies will not be required in general. After the data is identified and distributed, the scientist(s) involved will perform their study without any further interaction with the central repository. This study of the obtained data will be an activity of the second type mentioned above, involving the full life-cycle (Figure 1. Note that in a global project, experiment design and data collection are performed by a carefully chosen team of domain scientists and database administrators, whereas data exploration (in the form of simple browsing and retrieval) is performed by any scientist in the field. On the contrary, in a local study, the full life-cycle is performed by the same scientist(s). Clearly, the above distinctions between the two types of activities are not absolute. We believe, however, that in general there are some characteristic differences between them, which we hope have been exposed by the above comparison in the overall framework of the experiment life-cycle.

In the rest of the paper, we focus entirely on the second type of activity, i.e., on managing experimental studies conducted by individual researchers or small teams of them. Based on the above discussion and the structure of the experiment life-cycle, we examine some of the technical challenges faced by attempts to develop EMSs to support such activities. We then propose some solutions that we have adopted in our own efforts, which are centered around the versatility of conceptual schemas and their usefulness in a wide variety of tasks.

3. FUNCTIONALITY AND ARCHITECTURE OF EXPERIMENT MANAGEMENT SYSTEMS

Current database technology provides very primitive tools in the hands of scientists involved in experimental studies. All stages in the experiment life-cycle are viewed as distinct from each other with no or minimal communication among them. The transfer of data and the transition from one stage to the next are to a large extent "manual". Thus, many scientists end up using flat files to store the data of their experiments. For example, the following scenario is quite common. Collected data is stored in files with cryptic names like "out.1.100.13.7", which usually encode the values of the input parameters to the experiment. At data analysis time, application programs in conventional languages are written for every type of desired output. These programs have to look into the mass of files containing the relevant data, extract the useful information, and format it so that it is presented to the scientist in a meaningful way. Moreover, searching for the relevant data each time cannot be performed associatively (by the desired values of some parameters) but only by name, i.e., the given names of the files. Clearly, this process is very unnatural, tedious, error prone, and requires constant exposure of the scientist to the specific set of data, because the meaning of the various parameters is easily forgotten.

The role of an Experiment Management System (EMS) should be that of an agent between a scientist and a phenomenon under study. An EMS should provide the desired functionality for managing and analyzing data produced in experimental studies and overcome the inadequacies of today's technology. Such a system should be a single, integrated, tool that scientists can use throughout the life-cycle of an experimental study to effectively control and manage all aspects of the experimental process and the generated data, i.e., it should satisfy requirements (1) (3) of Section 1.

In order to achieve the above functionality, an EMS must be capable of both managing stored data and communicating with one or more experimentation environments (where experiments can be run) and other EMSs and DBMSs to obtain new data. Much like in a heterogeneous database system, given a scientist request, the EMS will first identify the experimentation environments and/or systems that are related to the request. It will then divide the request into pieces, translate each piece into the language of its target environment or system, and submit it for processing. In the end, the EMS will collect all the responses, generate one integrated result out of them, translate that into the appropriate user level representation, and return it to the scientist who posed the request.

An EMS should be able to communicate with other EMSs and DBMSs that manage data of interest already collected as part of other studies, so that duplication of effort is avoided. It should also be networked with several experimentation environments due to the very nature of experimental studies. In many cases, in order to investigate a phenomenon, or develop a new system, experiments under various control levels are performed. At one end of the spectrum are fully controlled experiments in which the system is simulated on a computer. In contrast to this, in the laboratory, where the environment can be controlled but the system has a life of its own, the observer has only partial control over the experiment. Finally, no control can be exercised when the real world is observed. An EMS that provides a cohesive interface to a range of experimental environments, which have been independently developed, possibly to solve problems of diverse scientific fields, has many advantages: a) transitions are smooth from one environment to the other, b) experimental data from different sources are analyzed in a single framework, and c) the EMS serves as a bridge between different experimental disciplines. An EMS with all the above capabilities will provide the richest possible support to scientists who will be able to flexibly use unlimited amounts of information to further their own research.

Another important feature that an EMS must have to achieve the desired functionality is that it must be capable of blurring the distinction between data collection and data exploration (data requests in particular) if the scientist so desires. This would conform to the natural vagueness of the separation between these two stages in the experiment life-cycle (Section 2). In particular, the scientist should be given the freedom to request data without any knowledge of whether it has already been measured and recorded or not. Depending on the situation, the EMS should decide whether to simply retrieve the data from its database, or initiate some action outside of the system.

Driven by the need for a tool to provide the kind of support described above in our own experimental studies, we have initiated an effort to develop an EMS at the University of Wisconsin. Figure 2 presents the architecture of that system, which we believe reflects the needs of a wide range of experimental studies. In addition to a component for the traditional query and storage services provided by a database system (Core DBMS), the EMS under development has an active component, which coordinates the interaction between the user and the experimentation environments (Experimentation Manager), and an analysis component for the stored data (Output Analyzer). The user interacts with the database system via intuitive language and graphical interfaces (User Interfaces). Finally, a variety of experimentation environments are coupled to the EMS via a component that translates data from its representation in the experimentation environments to its representation in the EMS and vice versa (Data Translator).

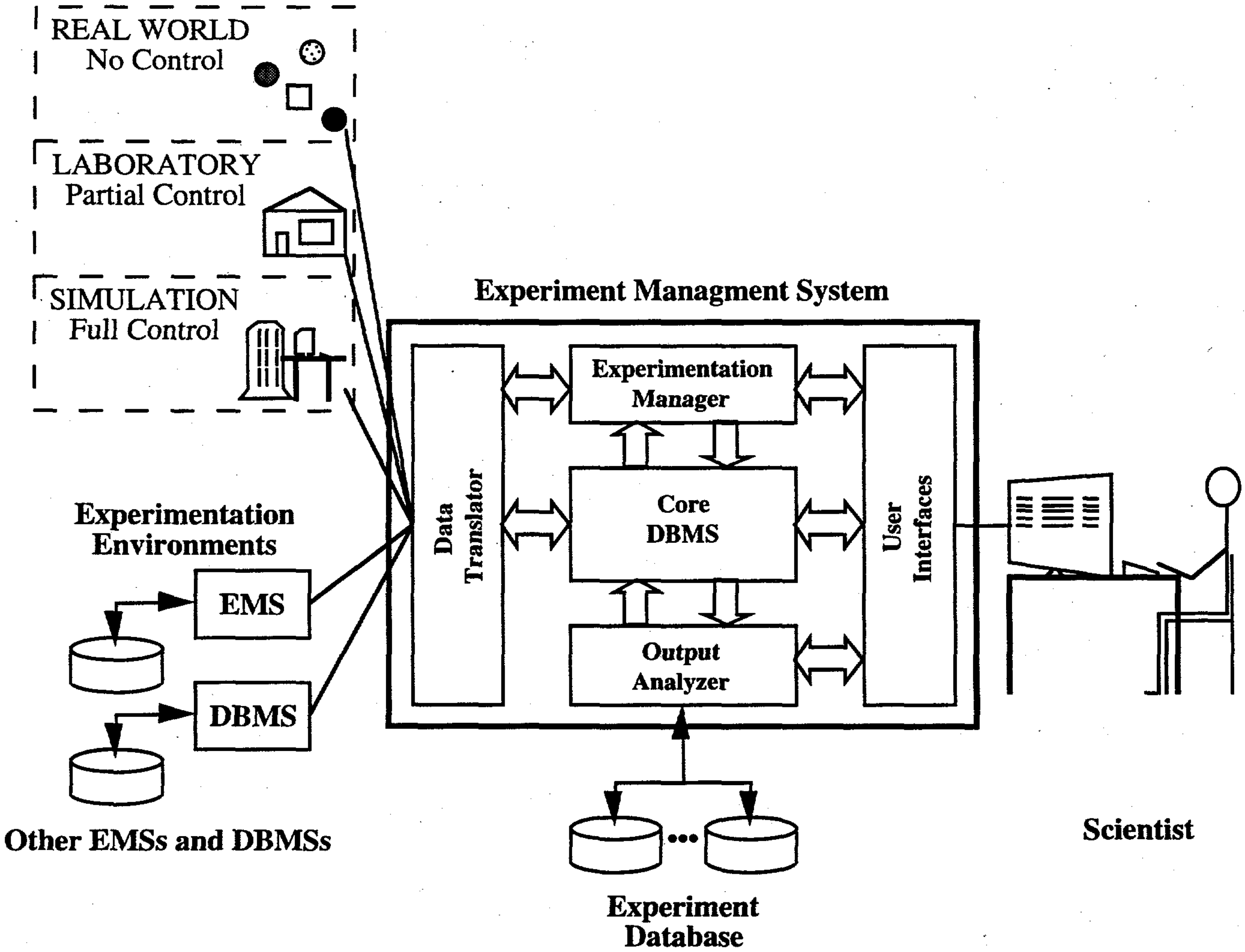


Figure 2: Architecture of an Experiment Management System.

We believe that a system developed based on the above architecture will have *achieved its* most important goals. It will provide an integrated environment to scientists that, unlike current practice, will feature a uniform interface that may be used for managing the entire life-cycle of an experimental study. Moreover, it will allow the design and execution of experiments and the access to scientific data to be done in ways that resemble as much as possible the way scientists interact among themselves using pencil and paper.

A fundamental premise of our effort has been that the above cannot be achieved unless the EMS provides a uniform and natural interface for all stages of the experiment life-cycle (item (2) in Section 1). The most important piece of information that is necessary in all these stages is the conceptual schema of the database related to an experimental study. Thus, it is only natural that our approach is "schema centric", where schemas are used in many roles (some of which are quite unique) throughout the experiment life-cycle, providing a common foundation for all types of interactions between the scientists and the EMS. This importance of schemas has also been the reason why our first priority has been to obtain a better understanding of schemas and their use, and to develop the schema support of the EMS, on top of which the rest of the system will be built. The results of this first phase of our work are described in the following sections.

In addition to our own effort to develop an EMS, some research laboratories have also been engaged in similar work trying to provide database support for scientific data. Examples include the "Laboratory Notebook" project in the Los Alamos National Laboratory [Nels9O] and the effort to develop data management tools for scientific applications in the Lawrence Berkeley Laboratory [Mark9l, Szet9l, Mark92]. Several aspects of our effort are found in at least some of these projects: schemas play several important roles, and intuitive (usually graphical) user interfaces are developed so that scientists may use them without much database expertise. On there other hand, several differences exist as well. The most important of them is that, to the best of our knowledge, our effort is the only one that attempts to provide a single tool for all stages of the experiment life-cycle. The other projects focus primarily on experiment design and initialization requests, which are similar to activities in traditional database management. Supporting data collection or the complex iterations of data exploration is not part of the functionality of the systems developed in these projects. Less significant are differences in the choice of data model (they are based on the relational or the extended entity relationship model, whereas we have developed our own object oriented data model), and in several other system aspects, on which we do not elaborate.

4. ROLES OF CONCEPTUAL SCHEMAS

In this section, we describe the roles that the schema plays in an EMS for each stage of the experiment life-cycle. In addition, we outline the features that a data model should have in order for schemas expressed in that model to play those roles.

4.1. Conceptual Schemas in the Experiment Life-cycle

By definition, schemas capture the structure and constraints of the data that is recorded in a database so that only valid data is accepted for storage. In most current database systems, schemas are used primarily for the above purpose. They are defined and altered by the database administrators, but cannot be manipulated or updated by end users. Such users may only consult the database system for information describing details of schemas, which is provided by help facilities. Thus, it is the user who initiates the flow of schema-related information out of the system. The DBMS itself is passive and only responds to user requests. The above state of affairs is considered adequate given the current roles played by schemas. In fact, the schemas of many databases are relatively small, so with frequent use, even memorization of the relevant parts of the schema by the user is common.

Although in traditional settings the above use of conceptual schemas is considered satisfactory, in an EMS it is not. For an EMS, the schema is a useful tool for many more activities than in traditional DBMSS In addition, by the nature of scientific studies, most user interactions with the system are in the form of ad hoc queries, whereas in traditional settings, running prepackaged application programs is much more common. In combination with the complexity and size of typical schemas of experimental studies, this makes the fact that the EMS has accurate knowledge of the schema much more valuable than it is in a conventional DBMS. Thus, the schema is called to play new roles in the context of an EMS. Accordingly, the EMS is forced to provide enhanced functionality compared to a traditional DBMS with respect to manipulating the schema and become active by taking the initiative in presenting the schema to the scientist.

Whereas in a conventional database the schema captures the structure of the data in the database, in experiment databases, the schema also captures the structure of the experiment itself. This is a side effect of the effort to describe the structure of the data: in order to organize the data in a meaningful way, the design of the experiment is essentially represented as well. For example, the schema of the plants experiment contains the various input and output parameters and their relationships to plant communities, which is precisely the information required to capture the design of the entire experiment. Based on this interpretation of its contents, the schema is called to play two new major roles in an EMS, in addition to its traditional roles:

(R1) In its first new role, the schema becomes the formal document describing the experiment. This is important for both individual and collaborative studies. Designing experiments, modifying earlier designs, describing experiments to others, integrating pieces of experiments into larger studies, and other activities that are usually based on arbitrary, and quite often free form, descriptions of experiments are now based on the conceptual schemas of the corresponding databases. In fact, in the first stage of the experiment life-cycle (Figure 1), the old notion of database design can now be seen in the new light of generalized experiment design.

(R2) In its second new role, the schema serves as the template for specifying data and experiments. Such specifications are useful both in interactions between scientists and in interactions between a scientist and the EMS. (Specifying data is not a new idea; although it has not been extensively used in commercial systems, it has been proposed and studied in the context of many research prototypes, e.g., [Agra9O, Bryc86, Fogg84, Gold85, Kunt89a, Pare92, Roge87, Wong82, Zloo77].) The ability of the schema to play this role is important in the data collection and data exploration stages (Figure 1). The system itself prompts the user with the schema, who then manipulates it appropriately for specifying query restrictions or for displaying query answers.

From the above description of the two roles, it becomes clear that the use of the schema spans all stages of the experiment life-cycle, which is fundamental to providing an integrated tool with a uniform interface to scientists. It also becomes clear that the conceptual schema undertakes these two roles not only within the EMS, but also in interactions between collaborating scientists as well. This can prove extremely important in the future, where multidisciplinary studies with large numbers of scientists participating will become more common [Hart92].

4.2. Necessary Data Model Characteristics

In order for schemas to play the above two roles successfully, they have to be expressed in a data model that has the following three characteristics. First, for Rl, the data model needs to be of high expressive power. Scientific experiments have quite complex structures, so the data relationships that must be captured in experiment databases are quite complex as well. The relational model is in general inadequate due to its simplicity. Its unique semantic primitive, the relation, is not powerful enough to express every aspect of an experiment design. The much richer object-oriented and semantic data models [Card9O,Zdon89] are the only serious candidates for such databases. Among other features, such data models offer primitives that can be used to represent complex objects (parts subpans), collection objects (sets), and class hierarchies with inheritance, which are very common in experiments.

Second, for both Rl and R2, the semantic primitives of the data model must be closely related to notions that scientists are currently using in their approach to experimental studies. A data model that is developed based on database expertise while ignoring the status quo in today's scientific laboratories is doomed to fail. Scientists must feel comfortable with the primitives of the data model so that they do not have to establish complicated mental mappings from their current way of thinking to that enforced by the model. As mentioned above, it is desirable for the data model to have high expressive power, but this should not be achieved at the expense of natural expression. The primitives of the data model should reflect the experience of scientists so that the complex data relationships found in experiment designs can be captured in a natural way.

Third, for both Rl and R2, schemas in the data model must have a succinct representation so that they are easily understood by scientists. Traditional text based data definition languages may not be the most appropriate tools for scientists to use for schema specification. SQL has quite a long and slow learning curve, and even for experienced users, writing very complex queries is not straightforward. It is doubtful that learning how to use a similar, but more complex, text based language for a very expressive data model is the best use of scientists' time. Intuitive graphical representations of schemas, supported by user friendly interfaces, will be of much more use to the scientific community.

From the above arguments, one may conclude that an EMS should have a graphical user interface that can deal with large and complex object oriented/semantic schemas in a natural way, allowing the user to manipulate the schemas for multiple purposes. Clearly, the demand for the first characteristic in a data model is not unique to scientific experiments. Many other applications have similar needs, which have driven the numerous research and commercial efforts to develop systems based on semantic and object-oriented data models. The demand I@or the second and third characteristics, however, is not so common. In most DBMSS, schemas are manipulated only by database administrators and complex queries are packaged in easy to invoke applications written by professional programmers, who are very experienced specialists in their respective fields. In an EMS, on the other hand, we want the scientists themselves to be able to interact with the system as both database administrators and sophisticated end users. Otherwise, much of the promised power of EMSs will be jeopardized. Thus, the need for data model primitives that are natural to scientists and for intuitive representation of schemas manipulated by easy to use tools is much more pressing in EMSs than, perhaps, other types of DBMSs and has received more focussed attention in our work.

In the following sections, we describe the data model that we have developed as part of our EMS effort together with some key features of the graphical interface that supports schemas in the model.

5. MOOSE: A DATA MODEL FOR SCIENTIFIC EXPERIMENTS

The EMS that we are developing is based on the Moose (Modeling Objects Of Scientific Experiments) object-oriented data model [loan89]. Although Moose is targeted for experimental data management, it is applicable in much more general settings as well. The salient features of Moose are described below.

5.1. Semantic Primitives of Moose

We first present the semantic primitives of Moose that define its expressive power. In the description, we put more emphasis on features that are not common among the already existing semantic and object oriented data models, justifying their inclusion in Moose by the needs of experiment management. Thus, we illustrate why we believe Moose satisfies both the first and the second desirable data model characteristic mentioned in Section 4.2.

Moose supports the notion of an object, which is quite intuitive to scientists because most often objects are used to represent physical entities that are relevant to experiments, e.g., the planet Jupiter or the part of the E. coli genome known as K-12 strain MG1655. Every object is assigned a unique object identifier and belongs to possibly multiple classes, inheriting properties from all of them. A class represents a set of objects having the same structure and the same properties. There are four system supported object classes, called base classes: integers, floats, character strings, and booleans.

The extent of a non-base class, i.e., the objects that are known members of the class, is explicitly stored in the database. This allows objects that are currently not part of any experiment design, i.e., that are not associated with other, higher level, objects, to still be stored and manipulated by the user. For example, in a simulation study of plant growth, one may want to introduce into the system a new variety of corn. Although, there are no experiments that have been run with this type of corn, nevertheless it is important that information about it is stored in the system, so that it is available later when the scientist decides to run experiments with it. In addition, "inventory queries" of the form "What types of corn do I have at my disposal?" are possible. The above needs are so common in experimental studies that having the scientist explicitly request the maintenance of the class extent for each class would require a significant effort, since it would have to be done for most classes in the schema.

In many experimental studies, object collections are often reused several times during the course of the study, e.g., a specific plant community. In addition, they are often associated with other pieces of information, which may or may not depend on the objects in the collection, e.g., the number of objects in the collection or some name given to the collection, respectively. To serve the above needs, collections of objects are individual objects themselves in Moose, carrying all the characteristics mentioned above. This uniform treatment of atomic and collection objects results in an economy of scale and makes sharing of collections and expressing properties of collections very natural. Otherwise, additional object classes would have to be defined, cluttering the schema and moving it further away from the usual intuition of scientists. There are four kinds of collection objects supported in Moose: sets, multisets (bags), indexed sets (a generalized form of arrays), and sequenced-sets (a generalized form of lists).

Each class in a Moose schema may be associated with many other classes, capturing a variety of relationships that may exist between the objects of the corresponding classes. Similarly to most semantic and object-oriented data models, Moose supports two major types of relationships: is-a relationships and part of relationships. The former capture semantic relationships whereas the latter capture structural relationships between objects of the participating classes. Specifically, is-a relationships relate classes to their subclasses (specializations) and vice versa, whereas part of relationships relate objects to their parts and vice versa. Every part of relationship is associated with a label, which serves the same purpose as an attribute name in relational DBMSS. For this reason, we occasionally use the term "attribute" to indicate part of relationships. The direction of a part of relationship is from a class of objects to the class of their parts. Every part of relationship, however, essentially captures a function and its inverse and can be explored in both directions. Therefore, it is associated with two labels. Quite often, one or both of these labels is equal to the name of the range class of the relationship traversed in the direction corresponding to the label, e.g., the utilization of a "cpu" is a "utilization". Whenever this is the case, we omit the label from the relationship declaration.

Two more types of relationships are supported by Moose to capture specialized associations of collection objects. A set to elements relationship connects a collection class to the class of elements in the collection, e.g., from the class of plant communities (sets of plants) to the class of plants. Exactly one such relationship must exist for each collection class. The need for this relationship is an immediate consequence of the need to support collections as first class objects. An indexing relationship connects an indexed set class to the collection class indexing it, which is the key set class of the relationship. Its semantics is that, each member of a key set is associated with exactly one member of the indexed set. Exactly one such relationship must exist for each indexed-set, except for those that are indexed by the natural numbers (i.e., those that are arrays in the traditional sense), for which such a relationship is implicit. Scientists need this relationship to express functional dependencies from the members of the key set class to those of the indexed set class. Such dependencies arise very often when the same parameter of the input or output of an experiment takes on a different value for each member of some collection related to the studied phenomenon or system, e.g., every distinct plant in a community has a different temperature. The parameter values (e.g., the temperatures) form the indexed set and the collection (i.e., the plants) forms the key set of this relationship. Through that, the parameter value associated with an object from the indexed set is directly available. In the absence of this type of relationship, for the same semantics to be captured, either auxiliary object classes would need to be defined, or the scientist would have to assign an integer number to each element in the key-set so that regular arrays could be used, which are supported by most systems. In the first case, again the size of the schema would increase with classes that play no substantial role in the scientist experiment design. In the latter case, the scientist would have to constantly use some unintuitive numbering to be able to indirectly associate elements to parameter values.

Finally, in Moose, part of relationships (and less often other relationships as well) can be declared as context dependent [Wien92]. A relationship of this type may be used to capture an association between a pair of object classes that depends on a third class as well. For example, in a study evaluating the performance of networks, a network site may be associated with a different job arrival rate in each experiment. This may be captured by a context dependent relationship between the class of sites and the class of arrival rates, with the class of experiments serving as the context. By definition, many relationships between objects in experimental studies are context-dependent on experiments. By having the ability to directly represent such relationships, scientists are able to design their experiments more naturally than otherwise.

Moose allows objects that are parts of a given object or instances of a given class to be defined either intentionally or extensionally. Specifically, the parts of an object do not have to be explicitly specified by the user. Moose supports the notion of a virtual attribute whose contents can be derived by the system through some computation associated with the attribute. Such computations are expressed in the form of rules that are based on the query language of Moose (whenever possible) or in some general computationally complete language among those supported by the system (whenever necessary).

Virtual attributes are especially useful to scientists for specifying aggregate computations over the members of collection objects. For example, every set of plants may be associated with a virtual attribute whose value is always calculated by counting the number of plants in the set. Given that the task of almost all experimentalists is the statistical study of some phenomenon or system, the implicit computation of aggregates as virtual attributes is an important tool. Moreover, the power of this feature goes beyond aggregate values and can be used for other purposes as well. For example, the entire output of an experiment can be considered as a virtual attribute that depends on the experiment input and The contents of which are computed by conducting an experiment.

Similarly, the membership of a class does not have to be explicitly specified by the user. Moose supports the notion of a virtual class whose membership can be derived by the system through rules associated with the is-a relationship between the class and some superclass of it. The importance of virtual classes can be realized by examining the experiment life-cycle (Figure 1). As a scientist explores the results of the conducted experiments, important characteristics of objects used in the experiments are identified. For example, a special behavior may be observed when the arrival rates in all sites of a network are the same. Upon such a discovery, it is common for scientists to give a special tag to such networks, e.g., call them "homogeneous networks", and put some further emphasis on investigating their behavior. In the context of an EMS, the equivalent steps are for scientists to define the virtual class of homogeneous networks as a subclass of net works and to associate the appropriate rule defining the members of the subclass with the corresponding is-a relationship. A nice side effect of this action is that any networks that happened to be homogeneous and were used before the scientist realized the importance of that subclass implicitly become its members, without any additional work.

Both types of implicit definitions remove significant work from scientists and enhance their ability to express complex relationships among classes. In addition, for all definitions expressed in the rule language of the system, inferences are made without explicit instructions from the users.

Finally, Moose supports many types of user defined structural constraints that may be used to control sharing among objects. A relationship may be one to one (referred to as single valued, non-shared), one to many (multivalued, non shared), many to one (single valued, shared), or many to many (shared, multivalued). Moose also has a constraint language, which may be used to express more complex structural constraints than the above. Such constraints express important aspects of the semantics captured by schemas. They are necessary in both general DBMSs and EMSS, which may use them to ensure the integrity of the stored data.

5.2. Graphical Representation of Moose Schemas

As mentioned in Section 4.2, the third important characteristic that a data model should possess in order to be useful in an EMS is that its schemas should have a succinct and intuitive representation, so that scientists who are nonexperts in database management can manipulate them without much effort. This has been one of the major concerns throughout the development of Moose.

The result of our work in this direction is that Moose schemas can be defined graphically and manipulated by appropriate actions directly on the iconic representations of their primitives. Specifically, every Moose schema has a straightforward directed graph representation. Every node in the graph represents a class of objects and is labeled by the class name. Base classes are represented as ellipses, to be easily distinguishable from the rest, while all other classes are represented as rectangles. In addition to the corresponding class name, nodes representing collection classes are also annotated with a special symbol identifying the type of the collection, e.g., for sets, for multisets, for indexed sets, and 0 for sequenced sets.

Arcs in the graph capture the various types of relationships supported by Moose. Part of relationships are denoted by solid arcs, is-a relationships are denoted by dotted arcs, set-to-elements relationships are denoted by double solid arcs, and indexing relationships are denoted by zig-zag arcs. Part of arcs are labeled with the name of the associated relationship, unless the label is the same as the name of the class at the head of the arc, in which case it is omitted from the graph. Context dependent arcs are annotated with the name of the context class as well. Finally, the structural constraints mentioned above that control sharing among objects can also be represented graphically. All four combinations of such constraints are shown in Figure 3 for part of arcs; the same constraints are captured similarly for the other appropriate arcs.

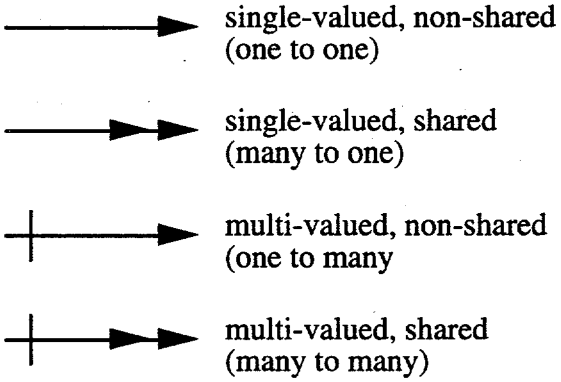


Figure 3: Graphical representation of structural constraints.

As an example of the graphical representation of Moose schemas, Figure 4 shows some possible schema for the plants experiment of Section 1. In addition to the original features, we also include the notion of homogeneous plant communities, which form a subclass of plant communities. The rule r associated with the corresponding is-a arc captures the precise definition of homogeneous plant communities, e.g., those where all plants are of the same type. Also, administrative information like the date of the experiment and the amount of time consumed by the simulator (its cost) are shown as attributes of the experiment. (We should emphasize that alterative schemas do exist for the plants experiment, some of which would possibly be more flexible than the one presented but also more complex. The above was chosen as a good trade off between simplicity and flexibility.) In Section 6, another complete Moose schema is shown graphically, in the form of a screen-dump of a prototype that we have developed for part of the user interface of the system.

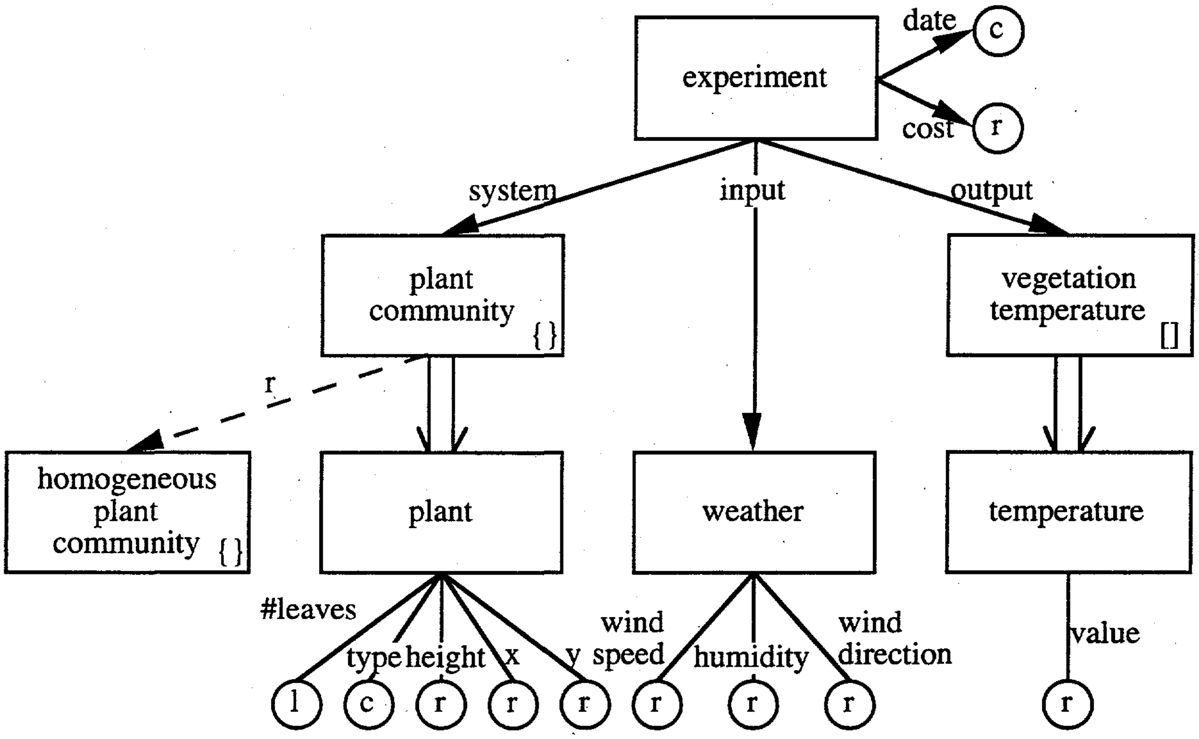


Figure 4: Possible Schemas for Plant Experiment

5.3. Moose Query Language

The query language of Moose is very similar to SQL and also has the flavor of other declarative object-oriented languages [Alas89, Banc89, Care88, Kife92, Kife89]. Since the query language is not the focus of this work, we are not describing it in any detail. We only want to point out that the fundamental construct in the language for traversing objects is the path expression. Path expressions connect objects in some class to other objects that are directly or indirectly related to them. A novel and powerful aspect of path expressions in the Moose language compared to other such languages is that they can include relationships of all types, and not only part of ones. Path expressions are closely associated with the graph representation of a Moose schema, since they essentially indicate paths in that graph. Based on this characteristic, a graphical query language that is closely related to the textual one is also under development. Given the focus of our effort, we expect that the graphical language will be the primary means of interaction of scientists with the system.

5.4. Summary Based on the above brief description of the features of Moose, we believe that it satisfies the three necessary characteristics described in Section 4.2. First, it supports a rich set of data types and semantic relationships between data that can be used in arbitrary combinations to capture the complex structures of experimental data. The various kinds of constraints can be used to ensure the integrity of the data, while the ability to define virtual attributes and classes (quite often through the use of rules) removes much burden from the user. Second, the semantic primitives of Moose have been chosen based on the needs of scientists. Especially the primitives that are not usually seen in other object oriented or semantic data models capture specific notions that are very intuitive to most scientists. The ability to use these same notions in the scientists' interactions with an EMS should make the system a much better and friendlier tool, better serving the needs of its users. Third, the graph corresponding to a Moose schema is a much more compact and intuitive representation of the schema than any other form, offering to scientists a better means to express their experiment designs. In addition, the development of a graphical user interface becomes possible, enhancing the usability of the system even further.

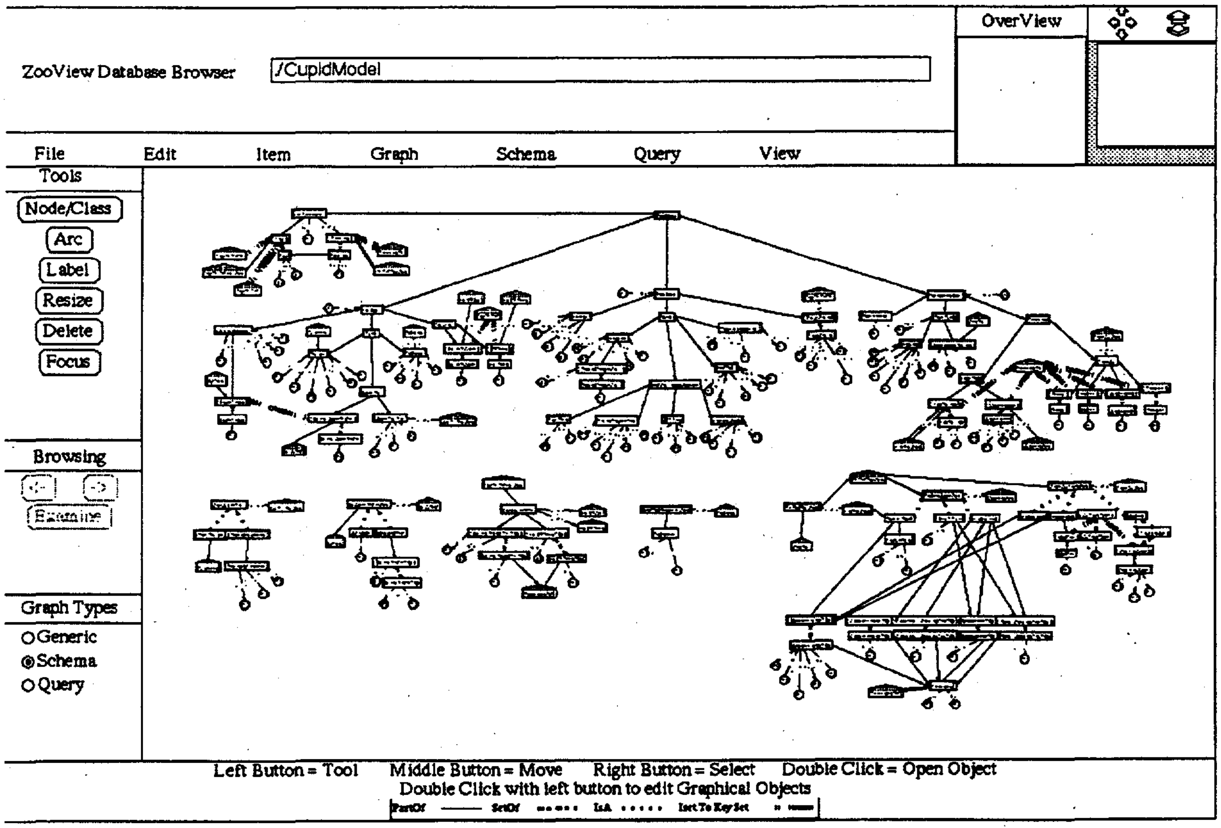
6. USING MOOSE FOR EXPERIMENT MANAGEMENT

Due to our firm belief in early prototyping, our study of experiment management has proceeded in parallel with the development of a prototype EMS, where the findings of our work are implemented so that they can be tested and validated. The goal of our effort is for the EMS to provide the appropriate technical support that will take advantage of the rich set of semantic primitives of Moose and its nice graph representation to become a versatile tool for scientists. In this section, we focus on a piece of the user interface of the system that has already been built, and describe its use in the context of an experimental study. In particular, we describe the graph editor of the system, which can be used to manipulate arbitrary types of graphs, and most importantly Moose schema graphs. The graph editor is a key part of the whole user interface of the EMS. This is due to the graph representation of Moose schemas, which transforms any schema manipulation to graph manipulation independent of the role played by the schema (Section 4).

The main difficulties in graph editing arise from the fact that Moose schemas for scientific experiments tend to be very large and can form an inscrutable maze of boxes and lines on the screen. Therefore, the key features that have been included in the graph editor deal with making large schemas more manageable. These are the following: (a) allowing parts of the schema to be made invisible; (b) collapsing subgraphs into single nodes; and (c) using "reference" nodes to eliminate very long arcs [Ioan92]. The above features are expected to prove very useful in supporting all aspects of schema use mentioned in Section 4. This expectation is justified by the results of our exposing the graph editor developed to "real" users, i.e., domain scientists, for experiment design (first stage in Figure 1). In all such collaborations, scientists from other disciplines have used the schema in its first new role, i.e., as a formal document describing experiments (Section 4) and the feedback obtained has been very encouraging.

Our work with John Norman from the Soil Sciences Department at the University of Wisconsin is one example of such collaborations. The main emphasis of his research is on simulating the growth of plants based on various environmental, soil, and ecological parameters. The primary tool in his studies is the Cupid model [Norm83], a Fortran program that simulates the necessary plant growth processes. The Cupid group has been using the graph editor that we have developed to document the structure of the input and output parameters of his model in the form of a Moose schema.

Cupid has been an excellent testbed for the capabilities of Moose and the graph editor because it has a complex structure and generates very large schema graphs. It simulates numerous processes that are parameterized, and therefore a large number of parameters need to be specified to characterize its input and output. Typically, about a hundred parameters are input to the model for any specific application, whereas the out put variables number in the several hundreds. Our collaboration with the Cupid group has shown that the graph editor we have developed can serve as a tool to organize the large amounts of data that Cupid manipulates. The schema for the input part of the Cupid model has been completed and contains more than a hundred object classes. It is shown in Figure 5 as a screen dump of the graph editor developed.



Left Button = Tool   Middle Button = Move   Right Button = Select   Double Click = Open Object
Double Click with left button to edit Graphical Objects

Figure 5: Moose schema for the Cupid input.

The feedback received from the Cupid group, which used Moose and the graph editor to design the schema of Figure 5 has been very encouraging [Ioan92]. The main benefits expressed follow quite closely the analysis of desirable features of the data model and the system described in earlier sections. Casting the Cupid data in an object oriented structure captured the complex array of data combinations that were part of the model in a natural way and made further modifications and enhancements of it easier. The graphical representation of the Moose schema was instrumental in allowing the members of the Cupid group to understand Moose relatively quickly and then use it for the experiment design. Finally, the resulting Moose schema played its RI role well: it served as a clear documentation of the input structure of the Cupid model and proved very useful in communicating its details to other scientists outside of the main group.

7. SUMMARY

One of the major problems faced by experimental sciences today is the lack of adequate tools for the management of experiments and data. We have undertaken the effort to develop a desktop Experiment Management System that will provide adequate suppose for scientists involved in experimental studies. In this paper, we have identified some of the fundamental issues that must be addressed in designing such an EMS. We have developed an abstraction of the set of activities performed by scientists throughout the course of an experimental study, and based on that abstraction we have proposed an EMS architecture that can support all such activities. The proposed EMS architecture is centered around the extensive use of conceptual schemas, which express the structure of information in experimental studies. Schemas are called to play new roles that are not usually found in traditional database systems. We have provided a detailed exposition of these new roles and have described certain characteristics that the data model of the EMS must have in order for schemas expressed in it to successfully play these roles. Finally, we have presented the specifics of our own effort to develop an EMS, focusing on the main features of the data model of the system. The feedback that we have received from domain scientists that have used the data model to design their experiments have been encouraging and have strengthened our belief that our approach will be able to serve the needs of experiment management.

Our effort to develop an effective EMS is far from complete. There are several issues that we are currently investigating and many others on which we plan to work in the future. Those currently under way include architectural issues on how the User Interface and the Core DBMS should communicate, efficient support for initialization and follow-up requests, graphical representation of objects, storage structures and query optimization in the Core DBMS, and formal issues on data and query translation. Additional issues left for the future include work on the Experimentation Manager, developing an internal interface layer so that a variety of output analysis tools can be connected to the system, e.g., visualization tools, and developing a data translator generator, which will be an easy to use toolkit for building data translators. In parallel to the above efforts, we will continue our collaborations with various domain scientists from different disciplines, so that the applicability of our findings to experiment management can be continuously tested and validated.

8. REFERENCES

[Agra9O] Agrawal, R., N. H. Gehani, and J. Srinivasan, "OdeView: The Graphical Interface to Ode", in Proc. of the 1990 ACM ,SIGMOD Conference on the Management of Data, Atlantic City, NJ, May 1990,, pp. 34 43.

[Alas89] Alashqur, A. M., S. Y. W. Su, and H. Lam, "OQL: A Query Language for Manipulating Object Oriented Databases", in Proc.]5th International VLDB Conference, Amsterdam, The Netherlands, August 1989, pp. 433 442.

[Banc89] Bancilhon, F., S. Cluet, and C. Delobel, "A Query Language for the 02 Object Oriented Database System", in Proc. 2nd International Workshop on Database Programming Languages, Salishan Lodge, CA, June 1989, pp. 122 138.

[Bryc86] Bryce, D. and R. Hull, "SNAP: A Graphics based Schema Manager", in Proc. 2nd International Conference on Data Engineering, Los Angeles, CA, February 1986.

[Card9O] Cardenas, A. F. and D. McLeod, Research Foundations in Object Oriented and Semantic Database Systems, Prentice Hall, Englewood Cliffs, NJ, 1990.

[Care88] Carey, M. J., D. J. DeWitt, and S. L. Vandenberg, "A Data Model and Query Language for EXODUS", in Proc. of the 1988 ACM-SIGMOD Conference on the Management of Data, Chicago, IL, June 1988, pp. 413 423.

[Fogg84] Fogg, D., "Lessons from a "Living In a Database" Graphical Query Interface", in Proc. of the 1984 ACM-SIGMOD Conference on the Management of Data, Boston, MA, June 1984, pp. 100 106.

[Fren9O] French, J. C., A. K. Jones, and J. L. Pfaltz, "Summary of the Final Report of the NSF Workshop on Scientific Database Management", ACM SIGMOD record 19,4 (December 1990), pp. 32 40.

[Gold85] Goldman, K. J., S. A. Goldman, P. C. Kanellakis, and S. B. Zdonik, "ISIS: Inter face for a Semantic Information System", in Proc. of the 1985 ACM SIGMOD Conference on the Management of Data, Austin, TX, May 1985, pp. 328 342.

[Hart92] Harttnanis, J. and H. Lin, Computing the Future, National Academy Press, Washington, DC, 1992.

[loan92] loannidis, Y., M. Livny, and E. Haber, "Graphical User Interfaces for the Management of Scientific Experiments and Data", ACM SIGMOD Record 20, 1 (March 1992), pp. 47 53.

[Ioan89] Ioannidis, Y. E. and M. Livny, "MOOSE: Modeling Objects in a Simulation Environment", in Information Processing 89, edited by G. X. Ritter, North Holland, Amsterdam, The Netherlands, August 1989, pp. 821 826.

[Kife89] Kifer, M. and G. Lausen, "F Logic: A Higher Order Language for Reasoning about Objects, Inheritance, and Scheme", in Proc. of the 1989 ACM SIGMOD Conference on the Management of Data, Portland, OR, June 1989, pp. 134 146.

[Kife92] Kifer, M., W. Kim, and Y. Sagiv, "Querying Object Oriented Databases", in Proc. of the 1992 ACM SIGMOD Conference on the Management of Data, San Diego, CA, June 1992, pp. 393 492.

[Kunt89a] Kunt7,, M. and R. Melchert, "Pasta 3's Graphical Query Language: Direct Manipulation, Cooperative Queries, Full Expressive Power", in Proc. 15th International VLDB Conference, Amsterdam, The Netherlands, August 1989, pp. 97 105.

[Livn87] Livny, M., "DeLab A Simulation Laboratory", in Proc. of the 1987 Winter Simulation Conference, Atlanta, GA, December 1987.

[Mark91] Markowitz,, V. M. and W. Fang, ",@Dl' A Database Schema Design and Translation Tool", Technical Report LBL 27843, LBL, Berkeley, CA, May 1991.

[Mark92] Markowitz, V. M. and A. Shoshani, "Query Specification and Translation Tools", Technical Report LBL 31155, LBL, Berkeley, CA, April 1992.

[Nels9O] Nelson, D., "The Laboratory Notebook Technical Manual", Technical Report LA UR 88 1256, Los Alamos National Laboratory, Los Alamos, NM, March 1990.

[Norm83] Norman, J. M. and G. S. Campbell, "Application of a Plant Environment Model to Problems in Irrigation", in Advances in Irrigation 11, edited by D. 1. Hillel, Academic Press, New York, NY, 1983, pp. 155 188.

[Pare92] Paredaens, J. et al., "An Overview of GOOD", ACM SIGMOD Record 20, 1 (March 1992), pp. 25 3 1.

[Roge87] Rogers, T. R. and R. G. G. Cattell, "Entity Relationship Database User Interfaces", in Proc. of the 6th International Conference on ER Approach, New York, NY, November 1987.

[Szet9l] Szeto, E. and V. M. Markowitz, "ERDRAW A Graphical Schema Specification Tool", Technical Report LBL PUB-3084, LBL, Berkeley, CA, May 1991.

[Wien92] Wiener, J., 0. Tsatalos, R. Miller, M. Livny, and Y. Ioannidis, "Direct Modeling of Context Dependent Associations in Semantic Data Models", submitted for publication, July 1992.

[Wong82] Wong, H. K. T. and 1. Kuo, "GUIDE: Graphical User Interface for Database Exploration", in Proc. 8th International VLDB Conference, Mexico City, Mexico, September 1982.

[Zdon89] Zdonik, S. B. and D. Maier, Readings in Object-Oriented Database Systems, Morgan Kaufmann, San Mateo, CA, 1989.

[Zeig76] Zeigler, B. P., Multifacetted Modelling and Discrete Event Simulation, John Wiley & Sons, New York, N.Y., 197

[Zloo77] Zloof, M. M., "Query by Example: A Database Language", IBM Systems Journal 16, 4 (1977), pp. 324 343.

Algebraic Optimization and Parallel Execution of Computations over Scientific Databases

Goetz Graefe, Portland State University
Richard H. Wolniewicz, University of Colorado at Boulder

1. The Volcano Scientific Database Project

Since many scientific applications manipulate massive amounts of data, database systems are being considered to replace the file systems currently in wide use for scientific applications. In order to counteract the performance penalty of additional software layers (i.e., the database management system), we are investigating the use of traditional database techniques to enhance the performance of computations over scientific databases. Our two focus areas are automatic optimization and parallelization of processing plans that include both numeric and database operations.

The integrated algebra is the crucial point in our research, because it permits breaking the barrier of optimization scopes as illustrated in Figure 1. Today's typical usage pattern of database systems in scientific applications is shown on the left; it includes a strong barrier between the database operations and the actual scientific computation, while our goal is shown on the right - after the barrier has been removed, the algebraic optimizer can operate on a much larger scope and therefore be more effective.

While database systems use algebras both on the logical level (e.g., relational algebra or the many proposals for object-oriented query algebras) and on the physical level (the set of execution algorithms), algebraic specification of scientific computations is also attractive and has been used in many interactive statistical software packages. Our contributions to the performance of computations over scientific databases are (i) to design a framework, both conceptually and in form of software tools, for algebraic optimizations of computations over scientific databases, (ii) to specify a *single* logical algebra that integrates both numerical scientific computations and traditional database operations for set- and pattern-matching, (iii) to complement the integrated logical algebra with a suitable and efficient physical algebra, including some equivalent algorithms to permit optimizer choices according to different situations (file sizes, etc.), (iv) to develop an appropriate cost model for the physical algebra, (v) to validate our optimization research by means of an algebraic query optimizer that maps an expression over the logical algebra (i.e., a computation over the scientific database) into an optimized processing plan, and (vi) to validate our cost model and parallel execution strategies by experimental measurements of computations using the Volcano query processing system [8].

The important advantages of using a single algebra are that (a) the scope of query optimization, which had been limited to the database system's retrieval and pattern matching operations, has been extended to

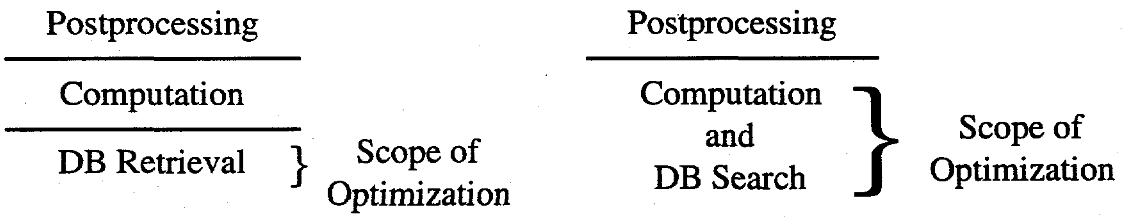


Figure 1: Breaking the Barrier in Optimization of Scopes

cover the entire computation, (b) the traditional two-level approach (retrieval vs. computation) has been overcome, permitting preliminary computations such as sampling to be performed before complex database operations such as matching of large database sets (joins, intersections, etc.), and (c) successful optimization techniques can be transferred easily from the database systems domain to scientific computations.

We currently have an operational optimizer developed with the Volcano Optimizer Generator [5, 7] that "understands" sets, time series, and spectra, i.e., relational algebra plus sampling, interpolation, extrapolation, digital filtering of time series, spectral filtering and convolutions, and Fourier transforms [11]. In addition to algebraic transformations and algorithm selection, the optimizer considers a variety of parallel execution strategies, and realizes them by inserting "exchange" operators into processing plans [3, 4]. We also have all operations listed above integrated into a single execution model and architecture, which is realized in an operational execution environment [5, 6]. We are currently linking the optimizer and the execution environment into a single experimental system.

2. An Example Application

The following is a sample application involving the integration of scientific operations and database techniques. This is a simplified model of a high altitude cloud model using both ground measurements of temperature, pressure, etc. and spacecraft data on high altitude water densities. These two data sets are the inputs into the calculation. Satellite data is arriving in real-time, and is driving the calculations. Ground observation data is available directly in a database file. Both data sets do not provide complete coverage of the atmosphere, so interpolation operations are applied to fill in gaps in the data. Interpolation can be accomplished by the existing Volcano interpolation operator. From these sets of data, observations near to each other are combined; for example, observations within 10km might be used together in the final calculations. Other uses for matching data from multiple sources in scientific data management include combining raw data with calibration data or matching current data with historical data (e.g., similar weather patterns). Matching of data values that fall within a range of each other is called a band join, and has been studied in [2]. With the matching data, a selection is performed to isolate the area of interest. Finally, the cloud cover calculations are applied, the data is graphed, etc. Such calculations can be performed by the existing Volcano filter operator using its "apply" support function.

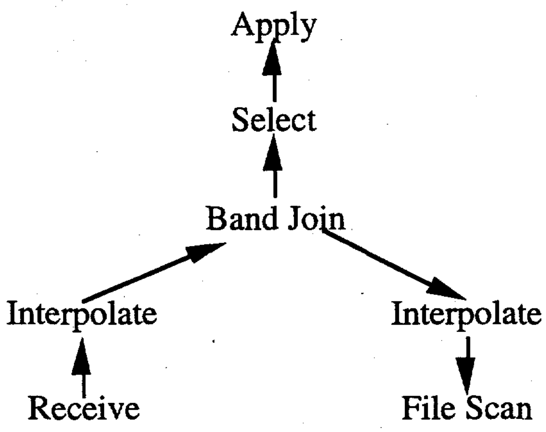


Figure 2. Operator-Structured Scientific Computation.

Figure 2 shows the full structure of the computation, with data flowing upward and the control flow indicated by arrows. Most of the computation is performed in data-driven dataflow, meaning that the control flow is upward, equal to the data flow. This is consistent with the real-time execution of the computation driven by the satellite receiver. However, the repetitive query to retrieve data from the database is executed in demand-driven dataflow, the model typically used in database query execution.

Having expressed the calculation as a series of data operations, we can apply database techniques to optimize and parallelize the tree. Some possible optimizations are shown in Figure 3. As the matching operation in the band join is likely to be expensive in comparison to other operators, performance can be increased by reducing the amount of data sent through the matching function. This is accomplished by pushing the selection operator (which reduces the data size) below the join, and by bringing the interpolation operators (which increase the data size) above the join. The latter step requires that the join become an outer-join, which is a join in which unmatched items are passed up the tree padded with null values.

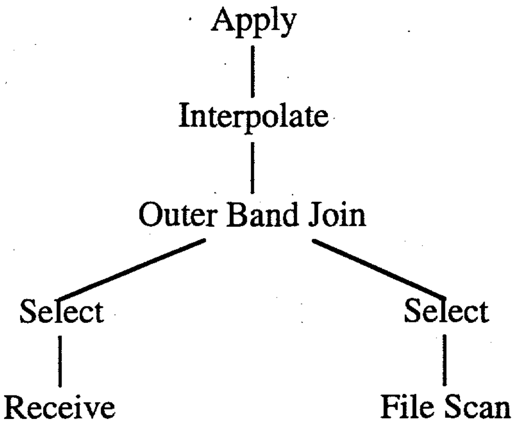


Figure 3: Optimized Computation

To exploit parallelization capabilities, exchange operators are inserted into the computation, for example below the join and above the interpolator as shown in Figure 4. These operators perform process management and data transfer between processes, including flow control. In Figure 5, the exchange operators are used to exploit parallelism: the input selection subtrees is executed by one process each, and two processes perform the join and the interpolation (each on half of the data), and one process supplies the final calculation and the display.

Our goal is to allow the optimization and parallelization of scientific operator trees to be performed automatically, allowing application scientists to take advantage of available resources efficiently without requiring detailed knowledge of algebraic optimization techniques, parallel algorithms, or the underlying machine architecture.

3. The Role of Meta-Data in Plan Optimization

Although meta-data are not the core of our research, we have identified a number of items that must be known for stored data and derived for intermediate results in order to facilitate effective optimization. In other words, the lack of these may impede optimization and therefore performance. Just as we distinguish between

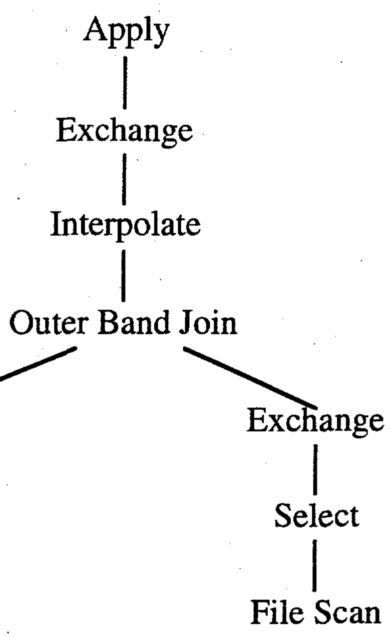


Figure 4. Computation with Exchange Operators.

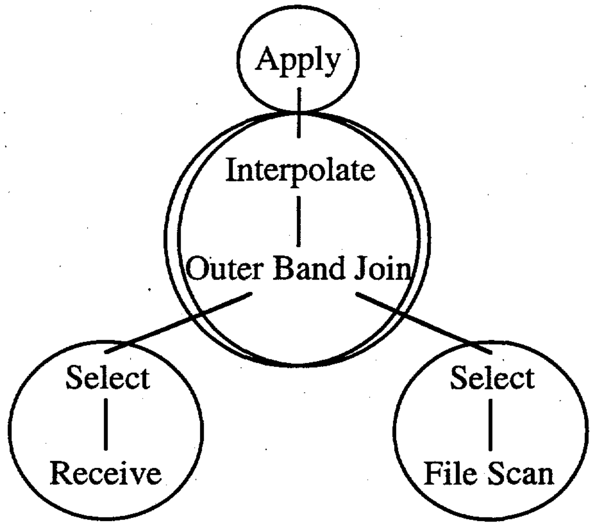


Figure 5. Processes in the Parallel Computation.

logical and physical operators, we distinguish between logical and physical properties. Logical properties can be derived within a logical algebra expression, while physical properties depend on the physical algebra and algorithm choice. For example, in the relational world, the schema (set of attribute and their names) and the relation cardinality are logical properties, whereas the sort order and the processing cost are physical properties.

Table 1 summarizes the properties used for optimization in our prototype. In order to permit effective query optimization, these items must be found in the catalogs or meta-data of a scientific database system. Similarly, during the optimization of complex computations, these items must be derived for all intermediate results. Thus, in our model of algebraic optimization, each logical and each physical operator has an associated property derivation function. For example, a relational join requires a function that derives the schema etc. of the join result from the schemas of its inputs and the join arguments, and the digital filtering algorithm requires a function that tags its output with the transfer function of the filter, e.g., in form of the filter constants or as Fourier series.

As our work progresses, we are likely to add more properties to the ones listed in Table 1. One of our short-term goals is to determine a set of properties that might be considered complete for the purposes of optimization. Logical properties are derived from the leaves (stored data sets) towards the root (final result) of a processing plan, i.e., from the properties of the inputs, the operators and the arguments. For physical properties, our model of optimization also includes the notion of "enforcers", i.e., physical algorithms that do not correspond to any logical operations. For example, the physical properties "sort order" and "uniform sampling intervals" can be enforced by a sort or by an interpolation operations. The optimizer automatically inserts appropriate enforcer operators at the most cost-efficient points in a processing plan.

The cost model is encapsulated in an abstract data type, permitting experimentation with different cost models. Example cost models include response time (cost = elapsed time) and resource usage (cost = CPU + I/O + Network). We are also experimenting with cost models that make cost decision undecidable at optimization time and delay plan choices until run time.

	Logical Properties	Physical Properties
Relations	Schema, cardinality, statistical summary data (e.g.,_histograms)	Sort order, compression, status, partitioning across_disks_or_a_network
Time Seriesl	Start and end times, measurement or sampling frequency, measurement channels (record fields), size, preprocessing history	Sequencing (sort order), uniformity of sampling intervals, partitioning across disks or a network
Spectra	Lowest and highest frequencies, measurement or sampling frequency, measurement channels (record fields), numeric precision, record size, preprocessing history	Sequencing (sort order), uniformity of frequency intervals, partitioning across disks or a network

Table 1. Logical and Physical Properties for Relations and Time Series.

4. Data Independence in Scientific Databases

Beyond automatic query optimization and parallel execution, a third concept that has contributed significantly to the success of modern database systems is data independence. In the three-level architecture shown in Figure 6, data are seen on three levels. On the physical level, files and records are stored on physical media, including storage aspects such as indexing, striping, and replication. On the conceptual level, data are much simpler - a dataset can be queried and manipulated independently of its physical representation. Thus, for example, the concepts of striping and replication have no place or meaning on the conceptual level. This separation of conceptual and physical levels is called physical data independence. The mapping of requests against the conceptual level into access path on the physical level is done automatically. Since there may be very many possible mappings of a conceptual request to physical access paths, this mapping is performed by the optimizer.

In addition, individual users may have specialized "views" into the database. These may be considered high-level access macros that may restrict the visible part of the database (for security and privacy reasons) or perform some on-the-fly computation. For example, a view might translate a date-of-birth into an age or add the current GPA to each student record in the database. The important point is that this value is not stored in the database but calculated from primary database data (e.g., the transcript) each time the view is accessed. The mapping from the external to the conceptual level is also done automatically, which is called logical data independence.

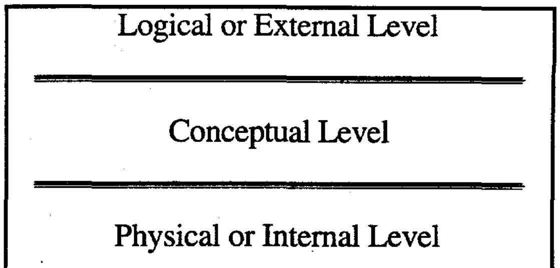


Figure 6. Three-Level Architecture of Data Models.

Since frequent on-the-fly calculation of views can be expensive, some database systems support materialized views, i.e., views are not calculated each time but stored and removed, updated, or marked out-of-date each time the underlying data change. The choice between on-the-fly calculation and materialization should be made with regard to all usages of a dataset, not from the limited perspective of a single user. Whether or not a view is materialized is part of the physical database design performed by the database administrator or an automated physical database design tool, not by an individual user.

Both logical and physical data independence could and should play an important role in scientific databases. When accessing a dataset for analysis, a scientist does not care how the dataset is represented on disk or in an archive, as long as the database system can materialize it in the required form. Moreover, when submitting a computation to the database system, a scientist would benefit from "macros" representing processing plans for the data, e.g., some basic preprocessing steps. And finally, materialized views could be managed automatically, not "by hand," with mechanisms derived from maintenance techniques developed for relational database systems, e.g. [1, 10].

In order to perform the mappings of requests from logical to conceptual and from conceptual to physical levels as well as to manage materialized views, the database management system needs mapping information. This information is data about data, i.e., meta-data. In relational database systems, this mapping information can be represented in non-procedural form. However, there is no reason why this information cannot be captured and retained in procedural form, i.e., as processing plans, as might be more appropriate in scientific databases. In future research, we plan on addressing appropriate view paradigms and mechanisms for scientific databases, which will also identify appropriate meta-data requirements.

5. Summary

In summary, we believe that an extensible query optimization tool such as the Volcano optimizer generator can be used for algebraic query optimization not only in relational and object-oriented environments but also in scientific database systems. Introducing automatic mapping of requests or processing plans not only promises efficient processing even for computer-naive users but also permits automatic parallelization of many processing steps such as data filtering as well as introduction of logical and physical data independence. Finally, we hope that the optimization and processing research for scientific databases can also be exploited for efficient query processing in object-oriented database systems that support bulk types other sets[9].

Acknowledgements

Our research on scientific databases is performed in collaboration with the Laboratory of Atmospheric and Space Physics at the University of Colorado at Boulder and is partially sponsored by the National Science Foundation with grants IRI-9116547 and ASC-9217394.

References

[1] J. A. Blakeley, P. A. Larson and F. W. Tompa, Efficiently Updating Materialized Views, Proc. ACM SIGMOD Conf., Washington, DC, May 1986, 61.

[2] D. J. DeWitt, J. E. Naughton and D. A. Schneider, An Evaluation of Non-Equijoin Algorithms, Proc. Int'l. Conf. on Very Large Data Bases, Barcelona, Spain, 1991, 443.

[3] G. Graefe, Encapsulation of Parallelism in the Volcano Query Processing System, Proc. ACM SIGMOD Conf., Atlantic City, NJ, May 1990, 102.

[4] G. Graefe and D. L. Davison, Encapsulation of Parallelism and Architecture-Independence in Extensible Database Query Processing, submitted for publication, November 1991. Also available as CU Boulder CS Tech. Rep. 559.

[5] G. Graefe, R. L. Cole, D. L. Davison, W. J. McKenna and R. H. Wolniewicz, Extensible Query Optimization and Parallel Execution in Volcano, in Query Processing for Advanced Database Applications, J.C. Freytag, G. Vossen and D. Maier (ed.), Morgan-Kaufman, San Mateo, CA, 1992. Also available as CU Boulder CS Tech. Rep

[6] G. Graefe, Query Evaluation Techniques for Large Databases, submitted for publication, January 1992. An earlier version is available as CU Boulder CS Tech. Rep. 92-579.

[7] G. Graefe and W. J. McKenna, The Volcano Optimizer Generator: Extensibility and Efficient Search, Proc. IEEE Conf. on Data Eng., Vienna, Austria, 1993. Also available as CU Boulder CS Tech. Rep. 563, December 1991.

[8] G. Graefe, Volcano, An Extensible and Parallel Dataflow Query Processing System, to appear in IEEE Trans. on Knowledge and Data Eng., 1993. A more detailed version is available as CU Boulder CS Tech. Rep.. 481, July 1990.

[9] D. Maier, S. Daniels, T. Keller, B. Vance, G. Graefe and W. McKenna, Challenges for Query Processing in Object-Oriented Databases, in Query Processing for Advanced Database Applications, J.C. Freytag, G. Vossen and D. Maier (ed.), Morgan-Kaufman, San Mateo, CA, 1992.

[10] A. Segev and J. Park, Updating Distributed Materialized Views, IEEE Trans. on Knowledge and Data Eng. 1, 2 (June 1989), 173.

[11] R. H. Wolniewicz and G. Graefe, Algebraic Optimization of Computations over Scientific Databases, in preparation, 1993.

Management of Inconsistency in Scientific Databases with Epsilon Serializability

Calton Pu
Department of Computer ScienceColumbia University
New York, NY 10027

Internet: calton@cse.ogi.edu

1. The Problem

A traditional database management system (DBMS) assumes that the database contains consistent data. This is particularly true when updates are involved. In classic transaction processing based on serializability [1], a transaction is defined as a program that transforms a consistent database state into another consistent state. There is little help from the DBMS if a transaction is starting from an inconsistent state.

Significant work has been done on information retrieval from the database containing imprecise, incomplete, or fuzzy data. They are primarily concerned with knowledge discovery instead of database state transformations.

Scientific data usually contain much incompleteness, imprecision, and inconsistency. This happens because of the nature of scientific data. Experimental data collected from scientific instruments naturally contain observational error in the instruments (e.g., determined by the precision of the hardware) and transmission error between the instrument and the database (e.g., when the instrument is on a satellite). Analytical data produced by theoretical models processing experimental data have uncertainty due to the accuracy and reliability of the models as well as the flaws in the experimental data, sometimes reduced by the theoretical models, but other times magnified.

An important question in scientific data management is how the DBMS can help scientists to handle inconsistency. The most immediate problem is how inconsistency is propagated when a database state is being transformed. Unlike traditional assumptions, where only updates need to be concerned about inconsistency states, if we start with an inconsistent database state, a query is also affected. The problem lies with the propagation of inconsistency in a query. Some operations preserve the amount of inconsistency, for example, errors from individual items accumulate linearly in a summation. Other operations magnify inconsistency quickly, for example, when numbers are used as the exponent in a power function.

2. Epsilon Serializability

We have introduced the notion of *epsilon-serializability* (ESR) as a generalization of serializability. The purpose of ESR [2,3,4,5,6] is to manage and control inconsistency on behalf of application programmers. ESR has three main advantages over previous ``weak consistency" models: (1) ESR is a general framework, applicable to a wide range of application semantics; (2) ESR is upward-compatible, since it reduces to serializability as $\varepsilon \rightarrow 0$; and (3) ESR has number of efficient algorithms that support it, derived from algorithms supporting serializability.

There are several benefits in allowing a controlled amount of inconsistency. For example, the DBMS may increase system throughput since data contention is lessened. For distributed TP systems, ESR allows asynchronous processing and therefore higher availability and autonomy.

The current ESR work is based on a regular geometry of database state spaces. Let *S* be a system state space. *S* is a *metric* space if it has the following properties:

- A distance function *dist*(u,v) is defined over every $\$ u,v \in S$ on real numbers.

- Triangle inequality.
  $dist(u,v) + dist(v,w) \geq dist(u,w)$
- Symmetry. $dist(u,v) = dist(v,u)$

A real world system state space usually contains strings and numerical values, too complex to be a metric space. For example, the bank system contains client name, address, account number, and account amount. However, the interesting updates happen only on the account amount attribute. If we consider the system state subspace by restricting our attention to the amount, we have a metric space. Scientific data that model the real world are invariably a metric space.

There are some examples of state spaces that are not symmetric. For example, the actual flying time from New York to California is longer than from California to New York because of jet stream. Also, there may be state spaces that do not respect triangle inequality. The investigation of non-metric distance spaces is an active area of research. Even though in a broad sense the question of whether a system state space is metric depends on the semantics of the system state since the physical world is a metric space, In any case, current algorithms that support ESR apply to any metric space, regardless of underlying system state semantics.

In general, ESR work is concerned with *how much* inconsistency is propagated in the database, when limited inconsistency is allowed in read and write operations. (Semantics-rich operations are another topic of active research.)

3. Kinds of Inconsistency

In the algorithms we have designed so far, we handle inconsistency in a simple way. We estimate the absolute upper limit of the inconsistency amount tolerable and maintain that limit on behalf of the applications. Although this is good enough for many applications such as banking, there are other kinds of inconsistency and uncertainty in scientific databases.

One important class of inconsistency in scientific data is denoted by error bars. When data contain some uncertainty, scientists usually adopt some statistical treatment of data and present their results through either error bars or confidence intervals. Processing queries and transactions that operate on data with such uncertainty, especially the handling of uncertainty propagation, is an important problem. Today's scientific programmers must handle the error and uncertainty themselves. The DBMS does not help much. This is an important part of the ESR research, with focus on support for scientific data. One approach that we are exploring is the design of an ``inconsistency algebra" for each type of uncertainty. If scientific programmers only use the operators defined in the algebra, then the DBMS will be able to handle inconsistency propagation according to the algebraic rules specified.

4. Conclusion

Meta-data is a term that usually denotes attribute description in relational databases and type description in object-oriented databases. We are investigating the possibility of extending DBMS support for the management of inconsistency by adding inconsistency description into meta data. The DBMS may help application programmer manager inconsistency by interpreting the meta data.

5. References

[1] P.A. Bernstein, V. Hadzilacos, and N. Goodman. *Concurrency Control and Recovery in Database Systems*. Addison-Wesley Publishing Company, first edition, 1987.

[2] C. Pu. Generalized transaction processing with epsilon-serializability. In *Proceedings of Fourth International Workshop on High Performance Transaction Systems*, Asilomar, California, September 1991.

[3] C. Pu and A. Leff. Replica control in distributed systems: An asynchronous approach. In *Proceedings of the 1991 ACM SIGMOD International Conference on Management of Data*, pages 377-386, Denver, May 1991.

[4] C. Pu and A. Leff. Autonomous transaction execution with epsilon-serializability. In *Proceedings of 1992 RIDE Workshop on Transaction and Query Processing*, Phoenix, February 1992. IEEE/Computer Society.

[5] K. Ramamrithan and C. Pu. A formal characterization of epsilon serializability. Technical Report CUCS-044-91, Department of Computer Science, Columbia University, 1991.

[6] K.L. Wu, P.S. Yu, and C. Pu. Divergence control for epsilon-serializability. *In Proceedings of Eighth International Conference on Data Engineering*, pages 506-515, Phoenix, February 1992. IEEE/Computer Society.

Toward A Taxonomy Of Metadata: The User's Perspective

Marcus Lester
Battelle Environmental Planning and Social Research Center
Seattle, Washington 98105-5428

1. Introduction

Data is expensive. Data acquisition and conversion can be the greatest expenses of a project, surpassing hardware, system design and data analysis. Metadata is data about data. it not the data of primary interest, but it describes various characteristics of a particular data set in greater or lesser detail. Metadata can significantly reduce a user's costs for locating, evaluating fitness for use, converting and analyzing a data set, if it accessible to the user and serves the user's needs. Standardized metadata is useful to the broadest range of users. Understanding user needs is an essential prerequisite to designing useful methods of incorporating metadata into large environmental sciences data sets.

There is a certain irony associated with metadata. Users always require metadata, else data are not useable. On the other hand, metadata is not a data production requirement. Thus, the party that is uniquely able to produce the metadata, and the one that usually bears the costs of production is the one with no apparent need for metadata. However, metadata represents value added to the data set that significantly reduces user costs for locating, evaluating, converting and analyzing the data of primary interest. Producers' recognition of this value depends upon the level of demand for existing data. If there is significant demand for existing or historical data, those who sell or archive data will view metadata less as an unrecoverable expense and more as wise investment.

2. Metadata Issues

2.1 Metadata as context

There are three principal sources for data: original survey, existing data sets and the operation of models or analytic procedures. The expenses for each are different, as are the problems encountered. Data from all sources are measurements that have been recorded. They become information when they are associated with some context. The richer the context, the greater the amount of information that is potentially available from the data. Some elements of context are the type of measurement, the number system, the instrument of measurement, the method of calibration, the location, the time, the medium used to record the data, the identity of the database and its structure. Without this context, the data conveys no information at all. Some unrecorded metadata remains only within the knowledge base of the principal investigator. Thus, data sets become unuseable when the principal investigator is no longer available to provide contextual information.

2.2 Shifting responsibilities for product liability

Data quality, encompassing reliability and fitness for the intended use, is a primary consideration in the selection of data. The Spatial Data Transfer Standard (SDTS 1992), recently adopted as FIPS 173, signals that the responsibility for ensuring data quality is shifting. "The purpose of the quality report is to provide detailed information for a *user to evaluate the fitness for a particular use* [emphasis added]. This style of standard can be characterized as 'truth in labeling,' rather than fixing arbitrary numerical thresholds of quality." (SDTS 1992, p. 1)

While this author is not giving legal advice, it appears that this is a new way of doing business. The responsibility for determining fitness now seems to lie with the user. The seller's responsibility is to include all relevant descriptive information, thus, guaranteeing the accuracy of the metadata but not the data. The user chooses the data set upon consideration of the metadata.

2.3 Metadata standards and interoperability

The value added by metadata is directly related to the extent that different users can access and use it. Conversely, metadata needs vary by user and data set. In addition, diverse efforts to categorize and encode metadata are in various stages of development and implementation. This situation presents a dilemma with respect to common use of metadata: whether it is more useful to promote the adoption of a single standard for metadata or, alternatively, to promote the interoperability of the micro-standards that currently exist.

For many users that have small and diverse data needs, the most effective resolution of this issue will be the adoption of a universal standard for metadata. However, standards are not necessarily the answer for all users. Standards are only useful as far as they are generally applicable. They cannot address all specific needs. Some users of large or specialized data sets will require metadata beyond any standard, especially for archiving, since the data may never be distributed outside their organizations. The argument, above, is not to forgo some standard, but to recognize that standards are not universally applicable.

Many large organizations that produce their data or that are are intimately connected with data production have generated micro-standards. In these cases, the needs of producer and user are somewhat confounded, and this is the perspective taken with the generation of these standards. It is seductively easy to take the approach of studying the problem and implementing solutions primarily from the provider's perspective. This will meet providers' needs and be commensurate with their current capabilities, but such solutions ultimately will be less acceptable and more expensive to general users.

Demarco (1979) has pointed out that a user needs analysis is always made. It is either an initial part of system design or the user does it after implementing the system. It is much less expensive to do the analysis early in the process than to wait for the user to find the deficiencies of an implemented system. This observation may be applied with equal validity to the generation of metadata standards.

As with many dilemmas, the middle ground is the most tenable. This view recognizes the usefulness of universal standards and recognizes the importance of existing micro-standards. A most useful standard will incorporate the characteristics of the metadata systems that are already in use. Thus, the most effective course may be: 1) to discover the similarities and details of these systems; 2) to promote interoperability of the most widely used systems; 3) to promote a set of guidelines that will allow subsequent metadata development efforts to maintain similarity with existing systems; and 4) to encourage the evolution of widely accepted guidelines into comprehensive standards.

3. A Conceptual Taxonomy Of Metadata

Goodchild (FGDC 1992 p. 3) suggests that metadata "(1) should be digital, structured, and support assessment of fitness for use; (2) are the responsibility of the provider and should accompany the data; (3) should be defined in the context of the user; (4) must have a standard terminology; (5) should focus on the prototypical user; and (6) are processable and viewable." Half these characteristics of metadata relate to use and users. Thus, the first step in discovering user needs for metadata is to define the fields of interest: users and metadata. A comprehensive study of users and their metadata needs is beyond the scope of this discussion. Rasmussen et al. (1990) describes a methodology drawn from studies in cognitive science.

The most useful taxonomy of metadata will be appropriate to the needs of the greatest range of users. Such a generally applicable taxonomy will be detailed, accounting for a variety of metadata needs. In addition, it will be structured to define clearly types of metadata. This paper suggests a hierarchical structure that reflects current uses of metadata and the contemporary efforts at standardization of spatial metadata. Since environmental sciences data often contains a spatial component, these efforts are worthy of consideration. They include:

SDTS (1992) provides for encoding of metadata relating to identification, catalog, security, spatial reference, registration, spatial domain, data dictionary (definition, domain, schema), transfer statistics, lineage, attribute accuracy, positional accuracy, logical consistency and completeness.

Roussopoulos (1982) recognizes schema and data dictionaries as metadata. These describe the structure and classification, respectively, of data within the database.

Vrana (1992) has suggested that qualitative metadata may be classified into four basic categories: identity; content; structure; and lineage.

Nebert (1992) sees metadata as a spectrum of characteristics that proceeds, respectively, from the general to the specific: identity; description; custodian; availability; spatial domain/extent; scale/resolution; source; intended use/purpose; date information; size in mb/# of features; processing steps; quality tests; tiling structure; data dictionary; data model/schema.

The Federal Geographic Data Committee (FGDC 1992) reports that a minimum set of metadata includes geographic location and footprint, corner points, units of measure, datum and spheroid, data type (vector or raster), spatial data model, components, view of reality, data dictionary, description of theme, data format, distribution media, how to access, cost, quality, digitizing process and collection method, source and data developers, author, intended use, date, scale, base map, minimum mapping unit and thresholds, restrictions on use.

It is clear from even this limited set of examples that some authors include characteristics that others ignore, and that some describe several independent characteristics, while others combine them into a single item. Without a general taxonomy, each description of spatial metadata is dependent upon its particular context or focus of application.

In the light of SDTS placing the burden of determining fitness for use upon the user, not the provider, Nebert's (1992) and FGDC's (1992) inclusion of "intended use/purpose" with spatial metadata is less appropriate than previously. The original intent of the producer is largely irrelevant, although this data may be considered in a prospective user's preliminary decision of applicability. Such metadata relates to past providers' attempts to limit implied warranties of fitness for use. This observation is not to torment Nebert or FGDC, but to recognize that production oriented views persist. The community of people concerned with large environmental sciences data sets has partly not recognized this fundamental change in the provision of spatial data. Further, the originally intended uses for data are not the limits of its usefulness. Limited or inappropriate items of metadata may unnecessarily restrict the breadth of use of a data set.

By their nature, all metadata is descriptive, but it is useful to differentiate two broad forms of description. Some metadata describes the general character of a data set with varying amounts of detail. Other metadata describes the reliability of the data. This taxonomy treats the two forms separately, because reliability metadata is usually used in a different context than the other metadata, and because it is quantitatively measurable, while the more generally descriptive metadata usually is not.

The following general taxonomy of metadata embodies a hierarchical classification scheme. It is not an exhaustive enumeration of the possible items of metadata. Such an enumeration is not currently possible. Different elements of metadata are relevant in different contexts, and any taxonomy of metadata must be flexible enough to accommodate a wide and varying range of user needs. In addition, this is a conceptual taxonomy. It is tenuously related to implementation, which will be heavily influenced by the type of data and its expected uses. For example, a description of instrument calibration may apply to the entire data set or only to a single data element.

3.1 **Qualitative Metadata**

Qualitative metadata is important for both archival and data sharing purposes. Its inclusion increases the utility and value of a data set to subsequent users. It is generally a report of the characteristics of the database, and with few exceptions, it is not testable for quantifiable accuracy.

3.1.1 **Identity**

This type of metadata identifies an individual data set, so that it may be retrieved at another location or time. A minimum identity is a unique identifier. This type may contain key words or other summary descriptive information to allow a preliminary evaluation of the data set's availability and fitness for a particular use.

3.1.2 **Content**

This is essentially the data dictionary describing all the data elements of the database, files and related databases. This metadata also describes data value ranges and other information about the type of data in the set, including:

- spatial extent, scale or spatial resolution of the data;
- thematic extent (features and value ranges included in the set) and thematic resolution (level of aggregation) of the data;
- temporal extent (total period) and temporal resolution (period between samples) of the data;

Some of this metadata also may be summarized in or referenced by the lineage metadata, below.

3.1.3 **Structure**

The database schema describes how the database represents and stores data. It is the logical organization or syntax for the database. Structural metadata is especially important for simplifying data conversion, often a significant expense.

3.1.4 **Security**

Security metadata describes the restrictions upon who may have access to the data.

3.1.5 **Lineage**

The SDTS (1992) includes a description of data lineage: the original sources together with all transformations of the data, leading to its present form. A list of important lineage metadata is:

- size of the data set;
- hardware and software required to access the data set;
- generating organization;
- sampling methods;
- time and duration (if applicable) of samples;
- responsible person or crew taking the sample;
- spatial coordinate system and method of measuring location of sample;
- storage and transportation of samples (method, location, duration, custodian);
- analysis and measurement of samples (organization, methods, date);
- instrument calibration methods and test results;
- method of recording measurements and units used;
- method and time of digitally encoding the measurements;
- all transformations of the digital data;
- transformations of spatial data from one coordinate system to another (especially any projection and its parameters);
- custodian or owner of data;
- persons to whom the data may be disclosed, and the conditions of disclosure.

Many items in the list, above, represent a level in the hierarchical taxonomy that subsumes additional lower levels. For example, there may be multiple transformations of the digital data, and each transformation may have a unique set of descriptors. Some transformations may apply to the entire data set, while others may apply to a subset. The descriptions may be wholly incorporated as metadata, or they may be incorporated as a reference to other published data (digital or otherwise). Thus, a hierarchical taxonomy can accommodate a wide range of implementation methods and particular instantiations.

The items of qualitative metadata vary in importance to different users, and different databases may exclude one or more of them. All are similar in that they are not generally testable by the user.

3.2 Reliability Metadata

All data is inaccurate to some extent. There is never an absolute correspondence between the measured values of a data set and ground truth: the values accepted as true or measured by the most accurate means. This is especially true of georeferenced data, which explicitly include an additional measure of location, and which may include one or more temporal measures.

The SDTS sets forth five forms of accuracy information to be included as part of data transfer: lineage, positional accuracy, attribute accuracy, logical consistency and completeness. This taxonomy classifies lineage as descriptive metadata. The current version of SDTS forms the basis for the following discussion of the characteristics of the other four forms of accuracy information. In addition, this taxonomy recognizes temporal accuracy as a separate component of metadata.

3.2.1 Positional accuracy

Positional accuracy includes the accuracy of the control survey, the survey method and all transformations to the final form that is the subject of the report. SDTS provides for four methods of measuring positional accuracy: deductive estimate (calibration tests and assumptions); internal evidence (closure of traverse or adjustment residuals); comparison to source ("check plots"); and comparison to an independent source of higher accuracy.

3.2.2 Attribute accuracy

The same methods are used to evaluate continuous measures as are used to evaluate positional accuracy. The accuracy of nominal measures may be tested by deductive estimate, independent samples (independent of the original survey) or an exhaustive polygon overlay, if spatial distribution is measured. The last test may be with reference to an independent repeated measurement or a source of higher accuracy.

3.2.3 Logical consistency

The logical consistency of the data set is the extent to which data values are within acceptable ranges for the data structure and the various types of data, that data are not missing from the data structure, that lines intersect only where intended (including polygon overlap), that all chains intersect at nodes, that a continuous cycle of chains bounds each polygon, and that each polygon completely encloses all inner rings (called islands or lakes).

3.2.4 Completeness

If both the spatial and thematic data are complete, then all space is accounted for as having some attribute and that all classes of attribute that should appear within the data set do so. Completeness may be reported with either an absolute (e.g., one missing class) or relative measure (e.g., per cent of coverage).

3.2.5 Temporal accuracy

Databases seldom include any measure of temporal accuracy. The implication is that, while any temporal data in the set may be inaccurate, this inaccuracy is reflected in its precision . That is, dates are assumed to be accurate to the day. Data given as hours and minutes, as an aerial photo, is assumed to be accurate to the minute. In fact, little temporal data has ever been available to users. Of necessity, one minimizes the significance of temporal data, when there is inadequate information with which to evaluate its reliability.

There are several aspects of temporal accuracy. The most obvious is the accuracy with which any particular time stamp is measured. This is not testable by the user, who must rely upon calibration data. Thus, this is qualitative metadata, properly included above. A second form of accuracy relates to the lag between the time stamp, often called world time, and the time that the data is entered into the database, usually called database time. Lester (1991) has described a model that allows a calculation of the general time lag between world time and database time. This evaluation of past performance allows a user to make some limited inference about the extent to which a current database is out of date.

4. CONCLUSION

Metadata is expensive, and this expense limits its inclusion in data sets. From the user's perspective, metadata helps to reduce the considerable cost of locating, evaluating, converting and analyzing data. The evolution of the market for data will show whether metadata generate savings to users that are greater than the increased cost of the included metadata. If this is so, then standards for metadata will help to lower producer costs by reducing research and development expenses associated with metadata. Further, by making data more generally understandable and useable, standards increase the market for data, and are instrumental in lowering marginal costs for metadata.

Metadata exists primarily for users. The most effective efforts to design standards for metadata will emphasize user needs, and they will incorporate the characteristics of existing designs that do so. The best design procedure will be to identify existing systems, promote interoperability, generate guidelines for future designs, and to promote the progression of guidelines to standards. A taxonomy of metadata that supports such a procedure is an enumeration of hierarchical types, from which particular users may select elements that serve their peculiar needs.

REFERENCES

DeMarco, T. 1979. Structured Analysis and System Specification. Yourdon Press: Englewood Cliffs, New Jersey.

FGDC (Federal Geographic Data Committee) 1992. *Report of the Information Exchange Forum On Spatial Metadata, June 16-18, 1992*. U.S. Geological Survey: Reston, Virginia.

Lester, M.K. 1991. A Conceptual Model of Multidimensional Time for Geographic Information Systems. Unpublished Masters Thesis. University of Washington: Seattle.

Nebert, D.D. 1992. Data Characteristics and Quality: The Importance of Spatial Metadata. GIS World 5(7): 64 - 67.

Rasmussen, J., A.M. Pejtersen and K. Schmidt 1990. Taxonomy for Cognitive Work Analysis. Risø National Laboratory. Roskilde, Denmark.

Roussopoulos, N. 1982. Proceedings of the Workshop on Self-describing Data Structures. Sponsored by NASA Goddard Space Flight Center and University of Maryland Computer Science Department. October, 1982.

SDTS (Spatial Data Transfer Standard Technical Review Board) 1992. Spatial Data Transfer Standard. U.S. Geological Survey: Reston, Virginia.

Using Semantic Values for Semantic Interoperability

Edward Sciore
Computer Science Department
Boston College
Chestnut Hill, MA 02167
sciore@bcvms.bc.edu

Michael Siegel*
Sloan School of Management
MIT
Cambridge, MA 02139
msiegel@sloan.mit.edu

Arnon Rosenthal
The MITRE Corporation
Burlington Rd
Bedford, MA 01730
arnie@mitre.org

1 *Introduction*

The increased use of large statistical and scientific databases has provided an incentive for the development of new techniques for representing and manipulating metadata. These techniques must provide not only for better access, maintenance and understanding of data, but must also ensure semantic interoperabitity among users and systems. Early attempts at metadata representation and manipulation [McC82, GK88, Law88] did not provide the capabilities for semantic comparison and semantic reconciliation, both key features in semantic interoperability, nor did they provide insight into the problems caused by changes to metadata. Our approach to these problems is to make context information—that is, the meaning, properties (e.g., its source, quality, precision), and organization of data—an active component of information systems.

We have found examples in many different application areas where explicit context specification would be useful. One such example is the trade price of a financial instrument, which might be reported as a number such as 101.25. This number has many possible interpretations—is it the latest price? the closing price? what currency? what is the scale factor? what is the precision? the accuracy? In addition, the system providing this data is likely to have a different interpretation than the system receiving it. Semantic interoperation becomes possible when the context information associated with the values in database can be made available to applications. In particular, an information system will have the ability not only to automatically determine whether data interchange is meaningful, but also to identify effective means for converting the data.

In this position paper we argue that *semantic values* should be the unit of exchange between information systems. A semantic value is a piece of data together with its associated context. We show how *conversion functions* can be used to give meaning to operations on semantic values, such as semantic comparison. We then apply our theory of semantic values to the relational model, considering the problem of what it means for semantic values to be stored in relations. We introduce an extension of SQL, called Context-SQL (C-SQL), in

* Supported in part by NSF grant IRI-90-2189 and the International Financial Services Research Center at MIT

which these extended relations can be accessed and updated. C-SQL contains features which allow users to specify and access context information explicitly. Users may also pose standard SQL queries, in which case the context conversions and manipulations occur completely transparently.

Although the use of an extended relational model is reasonable for new applications, it is unreasonable to expect existing databases to be converted to the new model. Consequently, we allow a database administrator to specify a *data environment.* Data environments encode knowledge about the semantics of the data, and make it possible for meta-attribute values to be calculated instead of stored. A data environment can have different scoping levels (e.g., a relation-level environment or a database-level environment), and may use different specification techniques *(e.g.,* lookup tables, rules, predicates). Similarly, we allow applications to specify *application environments.* Application environments create a "default context" for queries, allowing users to query the database as if the stored values actually had the specified context.

2 Semantic Values

A *simple value* is an instance of a *type.* The semantics of a simple value is determined solely by its type. That is, if *3* is of type *dollars* then it denotes 3 dollars, and cannot be compared with instances of type *yen* or *meters. A semantic value* is the association of a *context* with a simple value. We define a context to be a set; each element of the set is an assignment of a semantic value to a *property.* Note that this definition is recursive. That is, the value of a property can have a non-empty context. Simple values are defined to be equivalent to semantic values having an empty context.

We write semantic values by placing the context of a simple value next to it within parentheses. For example, the following semantic value might appear in a stock market application:

1. 25(Periodicity= 'quarterly' (FirstlssueDate 'Jan. 15'), Currency= 'USdollars')

Here, the value *1.25* has two properties: *Periodicity* and *Currency.* The value of the former property is also a semantic value having the property *FirstlssueDate.* The semantics of *1.25* can thus be interpreted as a quarterly dividend of 1.25 US dollars with a beginning cycle of January 15th.

The addition of a context to a simple value helps to more accurately specify its semantics. One consequence of this additional semantics is that two syntactically different semantic values can have the same meaning—simple examples are *4(LengthUnit='feet')* and *48(LengthUnit='inches').* Intuitively, these semantic values are equivalent because there exist *conversions* between them. Formally, a *conversion function* for property P is a function which converts a simple value from one value of *P* to another. For example, if *cvtLengthUnit* is a conversion function for *Length Unit,* then *cvtLengthunit(4,' feet', 'inches')* returns the value *48.* A conversion function may be implemented in any programming language, and may involve table lookup (for currency conversion), consulting online data sources (for timely currency conversion), or logical rules. The database system maintains a library of conversion functions. Some conversion functions will be defined by the system, whereas others will be defined by applications.

Let v be an arbitrary semantic value, and let C be a context containing values for some (or all) of the meta-attributes in the context of v. Then we define the function *cvtVal(v, C)* to return the semantic value that results from converting v to context C. For example,

cvtVal(40(Length Unit= 'feet',scaleFactor=1), (LengthUnit='inches',ScaleFactor=10))

returns the value *48(Length Unit= 'inches ',ScaleFactor=10).*

The function call *cvtVal(v, C)* returns a semantic value. The context of this return value is C, and its simple value is obtained by composing conversion functions. In particular, let C be the context $\{P_1 = p_1,...,P_n = p_n\}$, let p_i' be the value of P_i in the context of v, and let a be the simple value of v. Then the simple value of the semantic value returned by the function call is

$$cvtP_n\left(...cvtP_2(cvtP_1(a,p_1',p_1),p_2',p_2)...,p_n',p_n\right)$$

Note that any property values appearing in the context of v but not in C are ignored in the conversion.

We now turn to the issue of what it means to compare two semantic values. In general, the result of a comparison is a relative thing. For example, consider the semantic values *4(Currency = 'USdollars')* and *300(Currency = 'Pesetas'),* and suppose that *cvtCurrency(4, 'USdollars', 'Fesetas') = 300* but *cvtCurrency(300, 'Pesetas', 'USdollars') = 3.* Then the above two semantic values should be considered equal if we are interested in their worth in Pesetas but not if we are interested in their worth in US dollars. For another example, consider the semantic values *'Paris'(LocationGranularity = 'City')* and *'France '(Location Granularity = 'Country').* Here, the locations should be considered equivalent if we are interested in whether they denoted the same country, but should be inequivalent if we are interested in whether they denoted the same city. The intuition behind these examples leads to the following definition.

> Let v_1 and v_2 be two semantic values. These two values are *semantically equal* with respect to context C if *cvtVal*(v_1,C) returns the same semantic value as *cvtVal*(v_2 ,C).

That is, in order to tell if two semantic values are equal, we need a context in which to compare them. This context is culled the *target context* of the comparison. We also call this kind of equality *relative equality,* because the truth value of the comparison depends on the target context. Arithmetic operators such as addition and subtraction can be defined similarly for semantic values.

3 Semantic Values in Relations

In this section we extend the relational model so that tuples are composed of semantic values. We define a *relation schema* to be a set of *attributes,* each of which has a specified *type.* A *relation is* a set of *tuples; a* tuple contains a value for each attribute in the relation schema. If t is a tuple in a relation having attribute A, then its A-value is written *t.A*.

Unlike the standard relational model, we allow attribute values to be semantic values; that is, any value in a relation can have a non-empty context. We use extended dot notation to refer to the (semantic) value of an attribute's property. That is, if the semantic value *t.A* has a property *P*, then *t.A.P* refers to the semantic value of this property. A reference to a property *Q* of *P* would be written *t.A.P.Q*, and so on.

```
create table TRADES
        (CompanyName char(50),
        InstrumentType char( 10),
        Exchange char(20),
        TradePrice float4
                (PriceStatus char(20),
                Currency char(15)),

create table FINANCES
        (CompanyName char(50),
        Location char(40)
                (LocationGranularity char( 15)),
        NumberOfShares int,
        Revenues float4
                Scalefactor int,
                Currency char(15)),
        Dividend fioat4
                (Periodicity char(10)
                        (FirstIssueDate date),
                Currency char(15)))
```

Figure 1: Schema Definition for the TRADES and FINANCES Relations

In keeping with the spirit of the relational model, we require all relations to be *homogeneous*. That is, if *A* is an attribute in relation *r* then the *A*-values of all tuples in *r* have the same properties. This restriction implies that each semantic value for *A* can be implemented as a subtuple, by mapping each property to an attribute. Attributes corresponding to properties are called *meta-attributes;* the other attributes are called *base attributes.*

The declaration of a relation schema in C-SQL differs from standard SQL in that the association between an attribute and its meta-attributes must be specified. This association is achieved by placing the declaration of a meta-attribute after its associated attribute within nested parentheses. Figure 1 shows the specification of the relation schemas *TRADES* and *FINANCES.* The *TRADES* relation has the four base attributes *CompanyName, Instrument Type, Exchange,* and *TradePrice,* each tuple in this relation records the trading of a financial instrument (e.g., company's stock) on an exchange. The *FINANCES* relation has the five base attributes *CompanyName, Location, NumberOfShares, Revenues,* and *Dividend;* each tuple in this relation records some of the financial information about a company.

The meta-attribute values in a relation or database often have a regular, well-defined pattern. This regularity might be a consequence of behavior in the real world (e.g. "All US companies report earnings in US dollars"), business rules (e.g. "Dividends are always issued quarterly"), or characteristics of the database chosen by its DBA (e.g. "All *Revenue* values are stored with a scale factor of 1000").

We cull the place where knowledge about possible meta-attribute values is specified a *data environment.*

```
create data environment for TRADES by rules
        if InstrumentType = 'equity' and Exchange = 'madrid'
                then    TradePrice.PriceStatus = 'latestNominalPrice' and TradePrice.Currency = 'pesetas';
        if InstrumentType = 'equity' and Exchange = 'nyse'
                then    TradePrice.PriceStatus = 'latestTradePrice' and TradePrice.Currency = 'USdollars';
        if InstrumentType = 'future'
                then    TradePrice.PriceStatus = 'latestClosingPrice' and Currency = 'USdollars';
create data environment for DATABASE by predicate
        Currency = 'USDollars' and ScaleFactor = 1
```

Figure 2: Two Data Environment Specifications

Data environments can be specified in many ways.

- by rules [SM91];
- by predicates in a logic [CHS91, SC91];
- hy functional expressions;
- hy tables (including virtual tables);
- by tagged attribute properties (such as *source, quality, security)* [WM90].

Data environments can he defined at several levels [McC82]. In this paper we focus on the relation and database levels. Figure 2 presents C-SQL syntax for two environments: the relation *TRADES* and the entire database. The environment for *TRADES* is defined by rules, whereas the environment for the database is defined by a predicate. Note that met a-attributes need not be prefixed by attribute names. The meaning of the term *"Currency = 'USdollars'"* in the database-level environment of Figure 2 is that the value for the meta-attribute *Currency,* in every appropriate attribute of all relations, will be *USdollars.*

A data environment can specify that a meta-attribute is *strict.* For example in Figure 2, the meta-attributes *TradePrice.PriceStatus, TradePrice. Currency* are declared to be strict. A strict meta-attribute is one whose value is determined completely by the environment, and cannot be overridden.

Every database system makes some tacit assumptions about the data it contains. When a database is used in a wider setting than originally anticipated, its assumptions need to be made more explicit. This is one of the fundamental requirements for semantic interoperability, and it can be achieved by means of strict meta-attribute specifications. In particular, we note that strict meta-attributes need not be physically stored in a relation because their values for a given tuple can always be calculated. A data environment in which all meta-attributes are strict is called *a Database Metadata Dictionary* (DMD) [SM91]; a relation having such a data environment will be stored no differently from traditional relations. Consequently an existing traditional relation need not be changed in order to include meta-attributes - All that is needed is for a DMD to be associated with it.

4 Data Manipulation

In this section we consider the issue of how SQL requests are affected when relations contain semantic values, and examine how SQL can be extended in order to take greater advantage of the meta-attributes in the database.[1] We begin by considering a standard SQL query. For example, the following query retrieves the name and location of those companies having greater revenues than IBM:

```
select t1.CompanyName, t1.Location
from FINANCES t1 t2
where t2.CompanyName = 'IBM' and t1.Revenues > t2Revenues
```

There are two issues caused by the presence of semantic values in the database. First, the values appearing in the output tuples are *semantic* values. That is, not only is the location of the company retrieved, but its granularity as well. Second, all comparisons in the **where** clause are *semantic* comparisons. In the above query, each semantic value for *t1.Revenues* will be compared with each semantic value for *t2.Revenues* using semantic greater-than.

Constants in standard SQL queries have no specified context, and thus the context of a comparison involving a constant is empty[2]. Similarly, if an attribute has no meta-attributes, the context of any comparison involving it is also empty. Consequently, such semantic comparisons involve no conversion and are the same as standard (syntactic) comparison in SQL. For example, consider the following query:

```
select CompanyName
from TRADES
where TradePrice > 50
```

Because the context of the comparison is empty, the query retrieves the names of those companies trading higher than 50, regardless of the currency involved.

If the user wishes to retrieve companies whose trade price has a value greater than 50 US dollars, then a context must be associated with the constant. A natural way to specify this association is to use explicit semantic values; consequently, C-SQL extends SQL so that semantic values can be used as constants. In particular, the appropriate C-SQL query is the following:

```
select CompanyName
from TRADES
where TradePrice > 50(Currency = 'USdollars')
```

C-SQL also extends SQL in that meta-attributes can be accessed directly using an extended dot notation. For example, the following query retrieves the company name and trade price (including context) of all stock transactions having a TradePrice is expressed in Yen and has a value of greater than 100:

[1] Here it is only possible to examine queries in SQL. Similar extensions have been developed for update and view definition.

[2] at least, in the absence of application environments

```
select t.CompanyName, t.TradePrice
from TRADES t
where  t.InstrumentType = ‘equity’ and
       t.TradePrice > 100 and
       t.TradePrice.Currency = ‘yen’
```

Note that the comparison on t. *TradePrice. Currency* does not invoke any conversion; only tuples whose value for this attribute is ‘yen’ can appear in the answer.

The data accessed by an application may have many different contexts, especially if the data is coming from multiple sources. We have seen bow the use of semantic comparison helps to hide these differences from user requests, giving SQL (and C-SQL) much more expressive power than in standard relational systems. Here we examine how the application may specify it context requirements using application environments. As with data environments, application environments can be specified at multiple levels. In this paper we discuss three levels: the relation level, the database level, and the request level.

A relation-level application environment is similar to the corresponding data environment, with the following difference. Whereas a relation-level data environment describes the contexts of the values in a relation, a relation-level application environment describes what contexts the application desires to see. For example, an application might declare a relation-level environment for *FINANCES,* asserting that the value of *Revenues. Currency* is always ‘USdollars’. This application would then be able to view the relation as if its *Revenues-values* were actually stored that way. Intuitively, a relation-level environment provides a “semantic view” of the relation, in the sense that the user sees the same data but having a different representation. This intuition is expanded upon in [SM91], where relation-level application environments are called *Application Semantic Views.* Relation-level application environments can be specified by means of rules, predicates, tables, etc., exactly the same as the corresponding data environments. For example, Figure 3 defines a relation-level application environment for *TRADES.*

Database-level application environments consist of predicates assigning values to meta-attributes, similar to database-level data environments. These bindings specify desired meta-attribute values for data in a database; for example, the predicate *“Currency = ‘roubles’”* in Figure 3 asserts that all values in the database having met a-attribute *Currency* are to be viewed in roubles, unless a specification in the relation-level environment overrides it. Bindings in database-level application environments also provide default meta-attribute values for constants in a

```
create application environment for TRADES by rules
        if InstrumentType = ‘equity’ and Exchange = ‘madrid’
                then    TradePrice.PriceStatus = ‘latestNominalPrice’ and
                        TradePrice.Currency = ‘pesetas’;
        if Exchange = ‘nyse’
                then    TradePrice.PriceStatus = ‘latestTradePrice’ and
                        TradePrice.Currency = ‘USdollars’;
create application environment for DATABASE by predicates
        Currency = ‘roubles’
```

Figure 3: Two Application Environments

request, allowing semantic comparison to apply to even standard SQL queries.

As an example of the effect of application environments, suppose that an application accesses the database of Figure 1 using the environments of Figure 3, and consider the following query:

```
select CompanyName, TradePrice
from TRADES
where TradePrice > 60
```

The database-level environment implies that the constant *60* is interpreted to mean 60 roubles, and the relation-level environment implies that *TradePrice* may be in different currencies, depending on the values of *Instrument Type* and *Exchange.* The meaning of the where clause is to find those tuples whose trade price is greater in value than 60 roubles; however, the *TradePrice* values are not converted to rouhies in the output tuples.

We have already seen one form of explicit environment specification, namely the use of explicit semantic values in a request. For example, the use of the value *60(Currency='USdollars') in* a comparison asserts that the constant 60 has the environment *Currency= 'USdollars' for* that comparison. We now introduce into C-SQL a new language construct, called the **inEnvironment** clause, in which an application environment is specified for an entire request. The bindings in the **inEnvironment** clause override any bindings from implicit environments. For example, consider the following query:

```
select CompanyName, TradePrice
from TRADES
inEnvironment        Currency = 'USdollars' and
                     PriceStatus = 'latestTradePrice'
where TradePrice > 100
```

Here the constant *100* is interpreted as the latest trade price in US dollars. In addition, all *Trade-Price* values will be converted to the context {*Currency= 'USdollars', PriceStatus= 'latest TradePrice'*} before they are output.

5 Conclusions

We have presented a theory for semantic values as a way to improve semantic interoperability, and have applied this theory to the relational model. This approach provides a significant contribution to the use of metadata in large statistical and scientific databases, as it improves the comprehension of data and simplifies the exchange of data among multiple, possibly heterogeneous, information systems.

To be effective, our approach requires a new system agent called a *context mediator,* whose role is to perform semantic comparison and conversion. Instead of requiring each pair of communicating systems to determine an interface, the context mediator compares data context with required context and synthesizes the translations. With a context mediator, an information system will be able to increase its ability to exchange semantic values gradually. A C-SQL context mediator is currently being implemented as part of the C-SQL Project at MIT.

Our work suggests several lines for future research. First, context information needs to be attached to objects larger than single attributes; this may be easier in object-oriented models. Second, it is necessary to define algorithms that the context mediator can use to plan conversions; this planning can be nontrivial when conversions' behavior is more complex than simply changing a meta-attribute from one value to another. Third, C-SQL query optimization techniques are needed. Finally, it is necessary to better understand the tools needed to allow data administrators and application developers to cope with context interchange in a large-scale environment.

References

[CHS91] C. Collet, M. Huhns, and W. Shen. *Resource Integration Using an Existing Large Knowledge Base.* Technical Report ACT-OODS-127-91, MCC, 1991.

[CK88] A. Goldfine and P. Konig. A *Technical Overview of the Information Resource Dictionary System (Second Edition). NBSIR 88-3700,* National Bureau of Standards, 1988.

[Law88] M. H. Law. *Guide to Information Resource Dictionary System Applications: General Concepts and Strategic Systems Planning. 500-152,* National Bureau of Standards, 1988.

[McC82] J. McCarthy. Metadata management for large statistical database. In *Proceedings of the Eight International Conference on Very Large Database Systems,* pages 470-502, Mexico City, 1982.

[McC84] J. McCarthy. Scientific information = data + meta-data. In *Database Management: Proceedings of the Workshop November 1-2, U.S. Navy Postgraduate School, Monterey, California,* Department of Statistics Technical Report, Stanford University, 1984.

[SC91] M. Shen, W. and Huhns and C. Cottet. *Resource Integration without Application Modification.* Technical Report ACT-OODS-214-91, MCC, 1991.

[SM91] M. Siegel and S. Madnick. A metadata approach to resolving semantic conflicts. In *Proceeding of the 17th International Conference on Very Large Data Bases,* September 1991.

[WM90] R. Wang and S. Madnick. Data-source tagging. In *Proceeding from the Very large Database Conference,* 1990.

A Proposed Method of Linking Data and Metadata Using the Object Model

Michael R. Woodford
National Oceanographic Data Center, NOAA, Wahington D.C.

1. Introduction to NODC

The National Oceanographic Data Center (NODC) in Washington D.C. is one of three national data centers established within the Department ofCommerce's (DOC) National Oceanic and Atmospheric Administration (NOAA). NODC was founded as the national facilit for acquisition, processing, storage and dissemination of global oceanographic data. NODC maintains the nation's historical ocean data base and makes these data available to the public and private communities. The master data holdings include data from a variety of sources including Federal agencies; state and local government agencies; private industry and research institutions; and universities.

The World Data Center A (WDC-A) for Oceanography, which is operated by NODC, is the facility for data of foreign origin. A large portion of NODC's data holdings are the result of direct bilateral exchanges with other countries. WDC-A is part of the Word Data Center system which operates under guidelines issued by the International Council of Scientific Unions (ICSU). [1]

Currently, NODC's data holdings total over 30 gigabytes of physical, chemical, and biological ocean data. The NODC is involved in many programs and projects to improve the quantity of available ocean data as well as to make the data more accessible to te user community. Global programs include the World Ocean Circulation Experiment (WOCE); Tropical Ocean-Global Atmosphere (TOGA) program; Joint Global Ocean Flux Study (JGOFS); and the Global Temperature Salinity Pilot Project (GTSPP).[2]

2. The Data Dilemma

Data collected from the oceans in situ derive largely from individual cruises staffed by a diverse, international array of researchers armed with varying degrees of sampling equipment and analytical technology. As a result the data are obtained by emploing a variety of different methods, under diverse conditions, and using different formats. Experimental instrumentation and technology are also advancing at an accelerated rate, leading to changes in the standard methods of sampling and analytical procedre. As a result the degrees of accuracy and precision change also.

With such variance in the data sets, it is very difficult to make realistic and meaningful comparisons between data sets separated by time and technology. At the NODC, the sheer volume of data assures that similar data sets are, most likely, not easil comparable. We perceive that the role of metadata is to allow the barrier of inter-comparison to be broken down permitting meaningful interpretation and comparison of temporally disjointed data sets. For our purposes, we define metadata as informationthat describes and defines the sampling and analytical methodologies used, the physical conditions at the time of sampling, platform and platform type, etc. In short, anything that will help define the 'context' of the data collection and analysis.

At the present time the NODC collects a standard set of metadata with each data submission. The metadata are archived in hardcopy form and linked to the digitalized main data by the accession number assigned by NODC. Changes to this ineffective, traditonal method have been proposed and studied and, in some cases, prototyped with few effective results. The NODC is now making a concerted effort to design and build a system for on-line access to its ocean data. An integral part of this system will be liking the metadata to the relevant ocean data. To do this successfully, the system is being designed using the relatively new object oriented technology.

3. The Object Paradigm

Object orientation is a relatively new technology that has developed in the last few years with the need for re-usable, portable code. Briefly stated, an object is data with code that accesses and manipulates it. Each object has a defined set of statesand behaviors. The state of the object is determined by the values of its data. The behavior is expressed by methods or operations that operate on its attributes. The object model encapsulates an object with functionality that is specific only for thatobject's class. Descendants of that class inherit the functionality of the parent class thus providing a mechanism for sharing methods and data between hierarchically related class types. Since objects carry their behaviors with them, they exhibit polyorphic behavior. This characteristic results in the ability to create common protocols that will span a vast array of objects and object classes.

The result of the object model is that the code written to control objects is more general and implementation-dependent than procedural code. The programs are therefore more reusable, interchangeable, and understandable. For example, a program can be witten to handle a collection of data set objects. The program defines searches, sorts, QC, and output for the data records. Since the code does not specify how the records are stored, the same code can be used to manage a collection of metadata objects. This is possible since the functionality has been removed from the application program and is embedded around the objects.

4. What Next?

The development of a system that can give maximum user control, as well as clean and seamless integration of a variety of traditional and non-traditional data types, will require the integration of object-oriented programming technology with relational dta base technology. The incorporation of metadata as an integral part of the database requires a flexible system in which new data types and their operators can be defined by the user and understood by the DBMS. [3]

Using the object model it is possible to create a data set object class. A descendant metadata object class can then be defined for the parent data set. This will provide a direct link for each instance of the data set class to its related metadata. I will also provide a more direct way for the metadata to be accessed by any processing procedures used by the data set in QC, analysis, or plotting/display. The code needed to complete the application program will be minimized due to the inheritance of fnctionality provided from the parent class to the child class. Greater flexibility will exist in the actual structure of the data portion of the object, permitting the user interface to be as rigid or as flexible as needed.

Providing this object oriented system with an event-driven architecture using a graphical user interface (GUI) will provide a powerful, robust system that is easy to maintain and use. Object systems are also portable over an array of platforms making t easier to provide resources and services to users on different operating systems.

5. Conclusion

Thoroughly documented data sets are extremely valuable to the researcher. Metadata can provide that documentation and are critical to effective data analysis and interpretation. At present, a coherent definition of metadata is lacking, thus preventing he definition of the structure of the metadata object. The role of the metadata object will be specified by the functionality that it comes to possess as user needs define its ever growing role in understanding and interpreting 'hard' data. The object mdel provides a way of managing metadata and linking it to the data in a way that is both powerful and flexible.

6. References

[1] NODC, National Oceanographic Data Center Users Guide (2nd ed.), U.S. Dept of Commerce, 1991, section 2, page 1

[2] Withee, G.W. and R.J. Abram, NODC: Access to the Historical Ocean Data Record, Earth System Monitor, Sept. 1990, pp. 7-8

[3] Panel Reports from the NSF Workshop on Scientific Database Management, Mar. 1990, Tech. Report TR90-22, Univ. of Virginia, Dept. of Computer Science

Metadata Management In Scientific Applications

Arie Shoshani
Lawrence Berkeley Laboratory
Berkeley, California

1. What is metadata?

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. Without the collection and maintenance of metadata, the data may 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 badges designed to measure radiation levels. Is the information on the type of badges, 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.

Throughout this short paper we use examples from a real project we have been involved with at Lawrence Berkeley Laboratory for managing the data and metadata of a specific scientific application, involving low level radiation datasets.

2. Sources of metadata

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 history, 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 the "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 process 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 may require writing of programs to check the validity of the data. Thus, the metadata is this case are the integrity conditions including the programs that validated them.

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 corrections, and when they were made. Often, such data are 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, as well as documentation that explains the methodology, assumptions, and algorithms used. In addition, there is a need to keep track of versions of derived data.

4. Approaches for supporting metadata

One possible approach is to treat metadata 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. Another approach is to use a commercially available Data Dictionary System in conjunction with a Data Management System. There is also a vast literature that claim that Data Dictionary Systems are too limited for the support of metadata, and propose new techniques such as the use of Faceted Classification for organizing and indexing
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 and use 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 sup- porting 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. A more limited but practical approach is to organize the text content into categories of information and keywords, which can then be made part of the "searchable" metadata.

5. What can be done in the short term?

>From our experience with this example scientific database application we claim that it is profitable in the short term to:

(1) Treat metadata as just another database. The main advantage is that an underlying commercial DBMS can be used to manage the metadata, rather than using a specialized data dictionary system or a thesaurus systems.

(2) Describe the metadata using an object-oriented data model. Since metadata for scientific applications are usually quite complex, the use of an object model (e.g. some version of an Extended Entity-Relationship model) simplifies and clarifies the description of the metadata. Furthermore, one can use existing tools to map the object-level model to the model of an underlying DBMS (such as a relational DBMS), and the querying of the metadata directly at the object-level.

(3) Permit and interchangeable query and browsing mode over the metadata. Systems can and should be designed to permit the specification of conditions for searching the metadatabase (e.g. by subject terms, by dates, etc.), and browsing the instances of the data (e.g. select and instance of a dataset, and view bibliography related to it). Further, we claim that there should be a smooth transition between searching and browsing of the metadatabase (in order to locate datasets of interest) and the searching and browsing of the datasets themselves.

The presentation will include a description of a prototype system that was implemented in support of this specific scientific application, and illustrate the concepts used with a series of window interfaces.

Design and Implementation of a Computer Based Site Operations Log for the ARM Program

Joyce L. Tichler
Brookhaven National Laboratory, Upton, NY
Herbert J. Bernstein
Bernstein+Sons, Bellport, NY
Stephen F. Bobrowski, Ronald B. Melton,
Pacific Northwest Laboratory, Richland, WA
A. Peter Campbell
Argonne National Laboratory, Argonne, IL
Donna M. Edwards
Sandia National Laboratories, Livermore, CA
Paul Kanciruk, Paul Singley
Oak Ridge National Laboratory, Oak Ridge, TN

1. INTRODUCTION

The Atmospheric Radiation Measurement (ARM) Program is a Department of Energy (DOE) research effort to reduce the uncertainties found in general circulation and other models due to the effects of clouds and solar radiation (DOE 1990, Patrinos et al. 1990). ARM will provide an experimental testbed for the study of important atmospheric effects, particularly cloud and radiative processes, and testing of parameterizations of the processes for use in atmospheric models. The design of the testbed known as the Clouds and Radiation Testbed (CART), calls for five, long-term field data collection sites. The first site, located in the Southern Great Plains (SGP) in Lamont, OK began operation in the spring of 1992. The ARM site selection process is discussed in DOE 1991.

The CART Data Environment (CDE) is the element of the testbed which acquires the basic observations from the instruments and processes them to meet the ARM requirements. A formal design process was used to develop a description of the logical requirements for the CDE. The requirements and design of the CDE are discussed in Melton et al. (1991 and 1992).

This paper discusses the design and prototype implementation of a part of the CDE known as the site operations log, which records metadata defining the environment within which the data produced by the instruments is collected.

2. OVERVIEW OF THE CDE

The CDE has three major physical elements: the site data system, the experiment center and the ARM data archive. (Melton et al. 1992).

The site data system is situated at the SGP site; as other sites are added to CART, new portions of the site data system will be put in place to handle observations from the new sites. The site data system ingests data from the instruments, controls the operation of the instruments and provides the site operator with tools to aid him in operating the site and insuring the production of high quality data. The site operations log is a part of the site data system. Quality assessment and flagging of data streams is done in the site data system. Data from the instruments and the log are forwarded to the experiment center and the data archive.

The experiment center is situated at Pacific Northwest Laboratory in Richland, WA. Its primary task is providing the ARM science team with the measurements needed to perform their research. These measurements incorporate data from the site as well as from other sources of data such as satellite data. The experiment center carries out higher levels of data quality assessment on the data received from the site data system.

The ARM data archive is the long term repository of CART data. It will provide data retrospectively to the ARM science team and the general scientific community.

3. PURPOSE OF THE SITE OPERATIONS LOG

The site operations log (the "log") captures information about changes in status of objects associated with the operation of the site. These objects include the site environment, the instruments located at the site, the site data system, the data streams generated by the site data system and the site personnel. The log is an electronic analog to a logbook or lab notebook, recording information about the actual data ingested from the instruments, i.e. "metadata".

The log serves both as a tool in the operation of the site and a source of information for the analysis of the data collected at the site.

4. CONTENTS OF THE SITE OPERATIONS LOG

A facilitated design session was held to determine what information should be captured in the log. Those involved in the design session included representatives of the various types of users of the log as well as members of the CART data management team. The types of information which were determined to be needed were structured into classes of information about the site in general, the instruments located at the site, the data streams generated by the site data system, and the facilities that make up the site.

Information about the site as a whole was further subdivided into information about things such as meteorological conditions at the site, the site data system, personnel and staffing, surface conditions and environmental, safety and health warnings.

Information about the instruments was subdivided into categories such as alarms, changes in location, changes in mode or configuration and status forecasts.

5. FUNCTIONALITY OF THE PROTOTYPE IMPLEMENTATION OF THE LOG

The prototype implementation of the log was designed to satisfy the following requirements.

- Entries should all be of the form:
  date-time stamp
  source of entry
  subject of entry
  contents of entry.

- Entries may be generated automatically by computers which are part of the site data system or manually by personnel operating the site.

- Entries from the log are to be forwarded to the ARM archive and experiment center at regular intervals.

- An event-driven continuous stream of entries in the log should be available for *ad hoc* users.

6. METHOD OF IMPLEMENTATION

In order to have something up and running quickly, it was decided to take advantage of the UNIX™ mail utility to generate automatic entries. A c-callable version of sendmail is provided to generate electronic mail messages with the added capability of supplying a date-time stamp and originators in the call different from the environment in which the electronic message is being generated. This allows messages to reflect the true originator and time of origin of the information in the message. Automatically executed scripts do the necessary file manipulations and cleanups.

The software is based on a user account, assumed to be on a Sun SPARC™ workstation; that user receives electronic mail messages intended for the log. Each of the electronic mail messages is reformatted for transfer to an Empress™ data base and also forwarded via normal electronic mail forwarding mechanisms, without manual intervention.

The Empress™ data base provides a graphical user interface (GUI) for manual log entries. These entries are also converted to electronic mail messages.

As entries are made to the log, they are displayed on the operator's workstation. A capability also exists to send entries to a printer at predefined time intervals. Files containing entries for each day are kept in mail "mbox" form so that they can be examined using either the UNIX™ mail utility or the Sun™ DeskSet™ Environment for OpenWindows™ Mail Tool.

The flow of entries into the log is shown in Figure 1.

7. EXPERIENCE WITH THE PROTOTYPE LOG

7.1 IMPLEMENTATION

The original implementation was done in approximately three person months. Changes since then have occupied approximately another two person months. The most difficult part of the implementation has been interfacing to the other parts of the site data system which were designed and implemented by other members of the CART Data Management Team.

7.2 USE

Training users of the log has been straightforward since initial use has been based on use of the mail facilities with which most of them were already familiar.

8. FUTURE PLANS

As the operator becomes more experienced in operating the site, it is expected that there will be the need for greater structure than is currently provided in the manual entries.

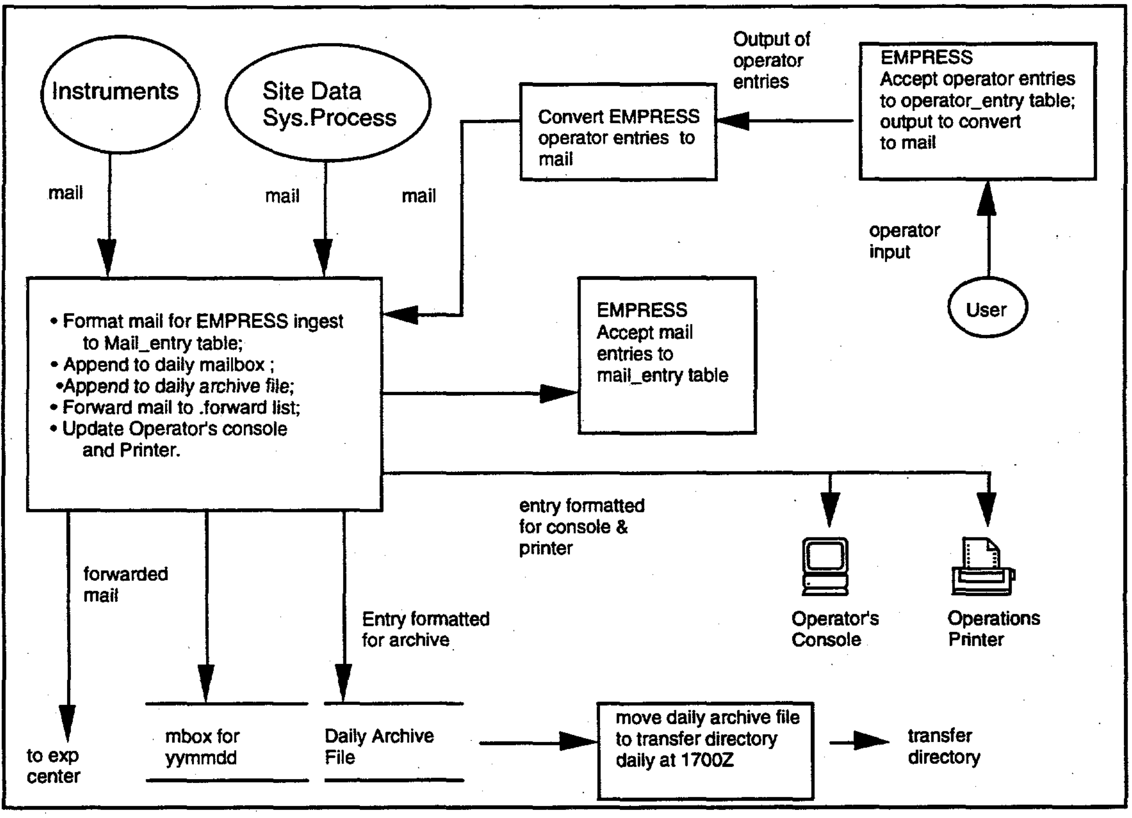

Instruments
Site Data Sys.Process
mail
mail
mail
Convert EMPRESS operator entries to mail
Output of operator entries
EMPRESS Accept operator entries to operator_entry table; output to convert to mail
operator input
User
• Format mail for EMPRESS ingest to Mail_entry table;
• Append to daily mailbox ;
•Append to daily archive file;
• Forward mail to .forward list;
• Update Operator's console and Printer.
EMPRESS Accept mail entries to mail_entry table
entry formatted for console & printer
Operator's Console
Operations Printer
forwarded mail
Entry formatted for archive
to exp center
mbox for yymmdd
Daily Archive File
move daily archive file to transfer directory daily at 1700Z
transfer directory

Figure 1. Flow of entries into the Site Operations Log

As the needs of the various users of the information of the log become clearer, it is planned to provided special purpose reports of selected entries in the log for them. Such requests can be divided by temporal requirements as follows:

Event driven: When information about specified objects are entered into the log, a report of the entry will be generated.

As requested: The user will request certain information from the log - e.g. the past performance of an instrument.

At regular intervals: A report will be generated at pre-specified intervals. An example would be to provide information to a daily operations plan.

ACKNOWLEDGMENTS

This work was supported by the Environmental Sciences Division of the U.S. Department of Energy, Contract Number DE-AC02-76CH00016.

REFERENCES

Melton, R.B., et al. 1991. "Design of the CART Data System for the U.S. Department of Energy's ARM Program." Proceedings of the Seventh International Conference on Interactive Information and Processing System for Meteorology, Oceanography, and Hydrology. January 14-18, 1991, New Orleans, Louisiana, American Meteorological Society, Boston, MA.

Melton, R.B., et al. 1992. "Clouds and Radiation Testbed Data Environment: Site Data System and Experiment Center." Proceedings of the Eighth International Conference on Interactive Information and Processing System for Meteorology, Oceanography, and Hydrology. January 5-10 1992, Atlanta, GA, American Meteorological Society, Boston, MA.

Patrinos, A. A., et al. 1990. "The Department of Energy Initiative on Atmospheric Radiation Measurements: A Study of Radiation Forcing and Feedbacks." Proceedings of the Seventh Conference on Atmospheric Radiation. July 23-27, 1990. San Francisco, CA, American Meteorological Society, Boston, MA.

U.S. Department of Energy (DOE). 1990. "Atmospheric Radiation Measurement Program Plan." DOE/ER-0441, Washington, DC.

U.S. Department of Energy (DOE). 1991. "Identification, Recommendation, and Justification of Potential Locales for ARM Sites." DOE/ER-0495T, Washington, DC.

Assembling textual metadata: a real-world experience.

W. R. Moninger
NOAA/Forecast Systems Laboratory

1. Summary

We discuss a system called Metalog--a repository of information and an electronic notebook for scientists. The system, and the *metadata it is designed to support*, is described in the context of other kinds of metadata and metadata support systems. We discuss our experiences using Metalog to document the Comprehensive Ocean-Atmosphere Data Set (COADS), which is used by more than 200 investigators world-wide. Finally, we present some of the lessons we have learned and issues that will need to be addressed by future metadata management systems.

2. Spectrum of metadata

Metadata consist of all the information necessary to properly use the data. Thus, Metadata may consist of any or all of the following, all at multiple levels of detail.

- Engineering information about sensors
- Sensor location information
- Temporal and spatial data coverage information
- Statistical data summaries
- Data processing information ("audit trails")
- Data formats
- Context information (how the data relate to other data)
- Information about data availability
- Information about errors and biases as a function of space and time
- Past doubts about the data that have been put to rest
- Open questions about the data
- Names and addresses of those who have analyzed the data

Although some of the metadata listed above are numerical information, much of the important information--particularly that information which results from detailed group analysis of the data--is most easily conveyed as textual or graphical. Moreover, metadata are dynamic; many important metadata, such as information about errors and biases in long-term trends, are not discovered until the data have been carefully studied in a variety of ways.

3. Some Pioneering Systems

Data analysis systems have always supported some metadata, such as the location and time information needed to display the data. In the last few years, the need for more kinds of metadata has begun to be more widely recognized, and many metadata support systems that take a variety of approaches to the problem have begun to be developed. Pioneering efforts in metadata management at NOAA in Boulder include the following.

- Trackline retrieval software at the National Geophysical Data Center (NGDC). Upon identification of useful data, these trackline extraction utilities search the databases producing plots of sensor tracks. The software has been applied to marine geophysical data, producing ship tracks, and to aeromagnetic data, producing aircraft tracks. Conceptually, this metadata inventory software can also extract satellite orbits from satellite data. The plots are helpful in selecting data sets for further analysis (Hittelman et al. 1991).

- A format description language known as Freeform has been developed at NGDC that allows easy reformatting of data sets. Freeform is supported by a software package that produces statistical summaries of data. These summaries have allowed NGDC to quickly identify potential errors in data sets having a variety of formats (Habermann and Miller 1992).

- A system known as MADER has been developed to support data from the Forecast Systems Laboratory's (FSL) Wind Profiler Demonstration Network. This real-time system provides on-line access to several types of metadata mentioned above (McGuirk, M. P., and S. Williams 1992).

- Metalog, discussed below in detail, was developed at FSL to facilitate the entry and management of dynamic textual metadata.

4. Metalog

Metalog (Moninger 1992; Moninger et al. 1992) started as an electronic log book for use in field experiments and data analysis. The system has a graphical user interface, and runs on Macintosh and PC-compatible (386 or more powerful) computers running Microsoft Windows 3.

Metalog stores free-form textual "comments," and individual comments can be of any length up to 32,000 characters. Comments can be retrieved by any word or phrase in the text; retrieval is not limited by pre-defined keywords. They can also be retrieved by any of several fields, such as type, project, and author.

Comments can be entered directly into Metalog, cut and pasted from other windows on the computer screen, and loaded directly from other text files. Comments can be cross-referenced and listed in multiple directories. Thus, comments can be used to annotate other comments. In this respect, Metalog is a hypertext system.

Because the system runs on personal computers, comments can be as private as the author desires. However, we have made provision for the sharing of comments among users. Comments can be printed or sent to a text file that can be read into other Metalogs or into any word processor document. Finally, comments can be sent to "metadata central," which at present is simply a file on each user's computer that appends all comments that are sent to it. The file can be periodically sent by mail or e-mail to a central location where the comments can be shared. Ultimately, we expect metadata central to be a direct link to one of several repositories of metadata.

Metalog is being used by several organizations in support of environmental research. For instance, the Paleoclimatology Program of NGDC is using Metalog to document a database on the little ice age, and to keep daily system records and instructions on how to use their graphical information system. NOAA's Wave Propagation Laboratory, and the National Center for Atmospheric Research are using an earlier version of Metalog to keep field notes during cloud and radiation studies.

5. Use of Metalog in Documenting COADS

COADS metadata were gathered and stored using Metalog. COADS Metalog currently contains 1.8 megabytes and consists of 628 comments and directories. Individual comments can and do appear in as many directories as are relevant, and are extensively cross-referenced. Contents include the following.

- Edited text from interviews with an expert on the assembly of COADS and the statistical processing applied to it.

- The text of two journal articles and a technical report about COADS.

- Abstracts of journal articles relevant to COADS found through a literature search and downloaded from an abstracting service.

- Citations relevant to COADS from a climate researcher's personal database of papers and talks.

- Sample information on routes and instrumentation for 10 ships, generated from a World Meteorological Organization data tape.

- Abstracts of talks given at the 1992 International COADS workshop.

- Ninety-five cross-referenced directories of comments.

- A tutorial on how to use Metalog to access the COADS metadata, and an extensive help system.

In addition, COADS Metalog accommodates users comments that can annotate the information already in the system or express other ideas. Also, users can develop their own directories and hierarchical structure. The number of additional comments and directories is limited only by imagination, enthusiasm, and disk space.

To date more than 50 environmental researchers worldwide have requested and received copies of Metalog. About half of the requesters are active users of COADS.

6. Lessons Learned

In applying Metalog to COADS, we learned several valuable lessons.

- Many comments about data do not need to reference a particular time and place. We quickly found that, at least for COADS, most comments refer to the data set as a whole, or to large spatial or temporal segments of the data. For example, there is no reason to give a precise spatial and temporal reference to a comment such as "Long-term trends in wind-speed may be artifacts because...". A structure (such as we originally anticipated) that required such references for every comment would be unnecessarily cumbersome.

- Authority of comments is important. Readers feel a need to know the source of items of metadata: whether it is from a journal article or from an off-the-cuff remark. Comments that are written, edited, and corrected by several people need to have a well-defined author history so that thoughts are correctly attributed. Comments that are incorrect, or correct but naive, can have a strong negative impact on the credibility of the entire package of metadata.

- The integrity of authors must be respected, or they could become hostile to the entire idea of electronic data documentation. Comments taken from journal articles (when allowed by copyright) should be verbatim from the printed version. In fact, we would not include a journal article in Metalog without approval of the author, simply because the difference between ASCII text and the typeset text might cause offense.

- In field experiments, use of electronic notebooks must be supported and encouraged by management. Our experience has been that, while approximately 8 field experiments, run by individual teams of experimenters, used Metalog fully, one multi-agency field experiment did not make wide use of Metalog. Hand-written notes, perhaps because they are more familiar, and may seem to be more private, may be preferred by some investigators.

7. Issues to Address

The complexity of environmental problems requires that increasing numbers of investigators with varied backgrounds will have to collaborate in studies and share data. Those who use the data must have adequate and up-to-date metadata. But metadata will only be timely and accurate if those with relevant information are willing to contribute it to a community base of metadata. Thus, the major problems that those who wish to establish metadata management systems face are largely sociological rather than technological. On the basis of our experience using metadata management systems over the past four years, we believe that these major problems need to be addressed.

Sociological:

- Cooperation. How does the community insure that individual scientists will be willing to share their information about data in a timely and complete manner? Concerns about precedence, politics, and prestige often mitigate against open sharing of information that may be important to proper data usage and interpretation.

- Credibility of information. How do users of metadata systems judge the credibility of the information contained? Information from refereed journal articles may--or may not--be more correct than information from technicians who have worked closely with the data and the sensors.

- Presentation of ongoing disagreements. Often, competent scientists will disagree about aspects of the data. For instance, there is substantial disagreement about which long-term trends in COADS are real and which are artifacts. New users of data sets should be forewarned about such open questions. To what extent will metadata managers be thrust into the role of editors and evaluators? What are proper policies to ensure that metadata include valid contrasting views?

Technological:

- Standards. Metadata will need to be shared by, and gathered from, many investigators. What are the appropriate standards for exchanging text, cross-references, and graphics?

- Version management. Metadata are dynamic; information about data that was once thought true may now be considered false. How should "old" and "new" metadata be merged to maintain consistent, credible information?

- Ease of use. Traditionally, metadata have been recorded on paper, which is highly portable, private, and easy to use. If scientists are to become willing to use computers in place of pen and paper, systems must provide advantages to individual scientists (such as speed of search and recall) that will outweigh the cost of having to learn to use yet another computer program.

Future systems for gathering and maintaining metadata will need to be designed and deployed with attention paid to these issues if they are to support wide, intelligent, and productive use of environmental data sets.

8. References

Habermann, T. and T. Miller 1992: Freeform Tutorial: A Flexible System of Format Specifications for Data Access. National Geophysical Data Center Internal Document (unpublished).

Hittelman, A. M., D. Metzger, and R. Buhmann 1991: "Improving Catalog Interoperability for Trackline Data Systems. *Earth System Monitor*, 1,4 (April) 5-6.

McGuirk, M. P. and S. Williams 1992: Wind Profiler Demonstration Network Metadata Access System. *Preprints, Eighth International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology*, Atlanta, Georgia. American meteorological Society, Boston. 46-49.

Moninger, W. R., S. D. Woodruff, R. L. Fozzard, and R. J. Slutz (1992): COADS Metalog: Dynamic COADS Documentation. *Proceedings of the International COADS Workshop, 13-15 January 1992, Boulder Colorado.* H. F. Diaz et al., eds. NOAA Environmental Research Laboratories (in preparation).

Moninger, W. R. (1992): Electronic Notes about Data: The Metadata Project. *Earth System Monitor* 2:1-3.

Collection and Dissemination of Data and Metadata from STORM-FEST

Arthur Wayne Brazille

U.S Weather Research Program Office
Office of the Chief Scientist
National Oceanic and Atmospheric Administration

1. Introduction

The Cooperative Distributed Interactive Atmospheric Catalog System (CODIAC) provides a new concept in data access for the mesoscale researcher. This interactive system offers the researcher the means to identify datasets of interest, the facilities to view metadata associated with the datasets, and the ability to automatically obtain the data of interest via either removable media or Internet file transfer.

This system has been developed to support the U.S Weather Research Program (USWRP). The goal of the system is to provide USWRP (and other) researchers with seamless access to a distributed meteorological database held at geographically dispersed data centers. This goal mandated that portability, scalability and, where possible, generality be significant design considerations.

The system has been developed by a coalition of NOAA organizations, specifically, the USWRP Office, the Forecast Systems Laboratory, and the National Climatic Data Center. See McGuirk (1991) for more details on this coalition and the development of the first prototype, MADER-91. The prototype system was implemented in December, 1991. Since that time, the system has been used by 333 people, and has delivered over 200 megabytes of data to users over the Internet.

This paper will address system features, describe how to gain access to the system, and give an overview of future system development plans, such as expansion to other data centers and the development of a new interface using a client/server paradigm.

2. System Features

The CODIAC System is designed to provide a seamless interface to data and metadata located at various data centers around the United States. The system utilizes the NASA Catalog Interoperability Model (Thieman, 1992). Level III interoperability is provided between the data centers involved, and Level II interoperability is provided to the NOAA and NASA Master Directories.

The system provides a variety of different functions and features. In the sections that follow, a brief description of the major components of the system is given.

2.1 Dataset Guide

The dataset guide provides descriptions of the various datasets supported by the system. The description includes such attributes as the dataset title, abstract, spatial and temporal resolution, archive center, level of quality control, type of observing system or network used to collect the dataset, and the name and other contact information for the curator of the dataset.

The dataset guide may be searched by a number of different methods, such as project, time, area, and observing system or network. Keyword search of the abstracts is also provided.

Project Information

The project information module provides a description of various field experiments (projects) whose data are managed by the system. The description gives a brief overview of the project, including a description of scientific objectives, and the spatial and temporal domain of the project.

Station Information

This module provides detailed descriptions of the observing platforms utilized to collect the data. Information such as station location, name, parameters observed, identification numbers, and times in operation are included in the description. These descriptions may be searched by a number of methods, similar to those described in Section .

Order Entry/Data Delivery

The Order Entry/Data Delivery system provides the user with the ability to obtain data directly from the data center through the CODIAC System. This module is composed of two parts: one which allows users to request delivery of data on magnetic media, and one which allows users connected to the Internet to download data that are on-line directly to their workstation or personal computer.

On-line data is provided at no charge through a cooperative agreement between the USWRP Office and the other data centers involved. Pricing and payment methods for delivery of data on off-line media vary from data center to data center. The system will compute the price for the desired data and offer the user payment methods based on the data center that the data will be obtained from.

Inventory Information

The inventory module provides detailed information describing, for each dataset, the specific times that data are in the archive. In most cases, this information is produced from the archived data files, and is very accurate.

Dataset Notes

This facility is an experimental implementation. It offers any scientist using the system the ability to attach her comments to a dataset. For example, if a researcher found a portion of a dataset to be bad, she could attach a note to the dataset describing the time period or stations that were bad. Other researchers that get the dataset later may then review the notes, and become immediately aware of the problem.

1. What is the internet (IP) address of the machine that you are using?

2. Does the machine that you are using support the X-Window windowing system?

3. What type of keyboard are does your machine have (for X-Window system), or what terminal type are you are using (non X-Window users)?

Keyboards that are supported under the X-Window system are:

PC/AT Keyboard
HP 9000/300 Keyboard
SUN3 Keyboard

DEC Keyboard
IBM System/6000 Keyboard
SUN4 Keyboard

Terminal types that are supported for non X-Window systems are:

VT100, VT220, etc.
SUN Console

PC running PC/TCP V. 2.05

Figure 1. Information needed to access the CODIAC System.

Reference Contacts

This is another experimental facility. When a researcher obtains data through the Order Entry/Data Delivery system, she is asked if she is willing to volunteer as a reference contact for the dataset. If she agrees, the system will ask her to characterize her familiarity with the dataset, and then her name and contact information (address, phone number, etc.) will be made available to all users of the system.

We hope that this facility will provide a way for users and potential users of a given dataset to network with scientists that are experienced in using the dataset.

ACCESSING CODIAC

The CODIAC system may be accessed by two methods. Each of the methods is described below, although access via Internet is preferred, as more of the system can be utilized. In the following sections, commands to be entered by the user are in **this font**.

Access Via Internet

To fully utilize the system across the Internet, You must have some knowledge of the configuration of the local system you will be using. In particular, you must know the information listed in Figure . If you do not know the answers to these questions, contact the support person for your system.

Once you have determined the answers to the questions in Figure , follow the sample dialogue given in Figure (X-Window users),

```
 telnet  storm.ucar.edu
Trying 128.117.88.53
Connected to 128.117.88.53
Escape character is '^]'.

SunOS UNIX (storm.mmm.ucar.edu)

login: storm
Password: research
Last login: Mon Aug 31 13:01:17 from cyclone.mmm.ucar
SunOS Release 4.1.1 (STORM) #7: Wed Aug 19 17:24:56 MDT 1992

          <Introductory text deleted to save space>

Are you running X-windows?[y/n/quit]->n

Emergency exit key = CTRL/C

Select your terminal type by number:

  1) vt100, vt220, etc.
  2) PC running an vt100, vt220, etc. emulation (PC/TCP v2.05)
  3) SUN Console
  4) exit

Enter the number of your choice->1

          <The CODIAC System Window should now be displayed on your terminal>
```

Figure 2. A CODIAC System login session for a non X-Window system user.

or Figure (non X-Window users). Remember, the commands that you type are in **this font**. Also note that editorial comments are contained between the <> symbols.

Access Via Modem

Access via modem is provided by the modem bank at The National Center for Atmospheric Research (NCAR). To obtain the toll-free number for access to NCAR, contact the Consultant On Duty (COD) at (303) 497-1278. After obtaining the number, follow the dialogue in Figure . Note that modem access supports character terminals only. X-Window connections are available only across the Internet.

Future Development

Future development of the CODIAC System falls into two areas: new software features; and connections to additional data centers. Each of these areas is described below.

New Features

Some of the new features that are to be added are: graphical display of data inventories; a database of meteorological events (e.g. tornados, hail, etc.); expanded project information, such as Intensive Observation Period (IOP) summaries and operations schedules; digitized daily weather maps (for times when field experiments were conducted); a software directory; and a client-server implementation of the software. Each of these features is described below.

The graphical display of data inventories is a critical component of the system. Most dataset inventories are presented as arrays with time and stations as axes. It is very difficult and time-consuming to determine what data is available from these arrays. The goal of the graphical display is to present the inventory as a plot of stations in a dataset with an indicator (for each station) of data availability for the station. The user will be able to interact with this graphic, changing time, parameter of interest, and region of interest. The ability to overlay inventories of different datasets may also be added.

The database of meteorological events (event directory) will serve as a new way to find research cases. A user will be able to search the event directory for a specific event (e.g. tornados), and get a list of times and locations where this event occurred. This list of time and location may then be used to search for datasets that may contain data for the event.

```
NCAR Host Connection Account
US Govt Property: Unauthorized use is a Federal Offense.
++++++++++++++++++++++++++++++++++++++++
+  NCAR Host Connection Account          +
+ (ONLY NCAR HOSTS MAY BE REACHED) +
+                                  +
+   Enter the hostname, or IP #         +
++++++++++++++++++++++++++++++++++++++++

What host do you want to connect to --> storm.ucar.edu
Checking name via domain name system.....

    Enter destination host login name: storm
    Enter password: research

trying  128.117.88.53...
Connected  to  128.117.88.53.
Escape character is '\377'.

<At this point you will be connected to the CODIAC System.  Follow the session transcript in Figure .>
```

Figure 3. Connection to the CODIAC System using a modem.

Expanded project information will be provided in a new project database. In addition to the overall description of the project, this database will contain details on each IOP, and operations logs for the various platforms involved in a project, such as radars, aircraft, satellite RISOP schedules, and polar-orbiter overpasses. The user will be able to search this database to identify projects and IOPs of interest, and then use this information to search for datasets.

The digitized daily weather maps will provide another resource for identifying research cases of interest. The daily weather maps for each day of a research project will be digitized, and stored in a database. The user will be able to specify the day of interest, and the weather map for that day will be displayed in a window.

The software directory will assist users in locating and obtaining software to work with various data and data formats. It will provide descriptions of public-domain software, and allow for direct download of some of the software. This database should facilitate research by helping scientists identify software tools that already exist, hopefully reducing the need to develop new software.

The client-server software is a new implementation of the CODIAC System. It will consist of client software that runs on Unix workstations and IBM-compatible personal computers. This software will interact with database servers running at the various data centers, and will provide seamless interoperability across data centers. Seamless interoperability will be accomplished by allowing the server at each data center to contact servers at other data centers. Thus, an user may connect to any of the data centers, and obtain data from all the data centers running a server.

Additional Data Centers

Another area of development is in expanding the current system to be available at other data centers. Current plans are to add two additional data centers in the next year: The Research Data Program; and The Data Support Section, Scientific Computing Division, both at the National Center for Atmospheric Research.

References

McGuirk, M.P. and Crowe, M., 1991: Wind Profiler Demonstration Network Data Management Activities. Seventh International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology, New Orleans, LA, January 14-18, 1991.

Thieman, J., 1992: Catalog Interoperability/Master Directory Project Plan. National Space Science Data Center, National Aeronautics and Space Administration. April 6, 1992. 98 pp.

```
xhost +storm.ucar.edu

telnet storm.ucar.edu
Trying 128.117.88.53 ...
Connected to 128.117.88.53.
Escape character is '^]'.

SunOS UNIX (storm.mmm.ucar.edu)

login: storm
Password: research
Last login: Mon Aug 31 11:16:47 from cyclone.mmm.ucar
SunOS Release 4.1.1 (STORM) #7: Wed Aug 19 17:24:56 MDT 1992

        <Introductory text deleted to save space>

Are you running X-windows?[y/n/quit]->y

Enter the internet ADDRESS of your X-server->999.999.999.999
999.999.999.999 is alive

Select your KEYBOARD type by number:

  1) PC/AT keyboard
  2) DEC Keyboard
  3) HP 9000/300 Keyboard
  4) IBM System/6000 Keyboard
  5) SUN3 Keyboard
  6) SUN4 Keyboard
  7) Exit

Enter the number of your choice->6
Emergency exit key = CTRL/C
Initializing X-windows sometimes takes up to 3 minutes!
Please be patient.

        <The X-Window should now appear on your display>
```

Figure 4. A CODIAC System login session for an X-Window system user.

User Scientific Data Systems: Experience Report

Elaine R. Dobinson
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

1. Introduction

This paper presents an abbreviated history of NASA science data management system development over the past ten years by selecting two case studies, each representative of a distinct era of science data management systems. The particular problems encountered by each of these systems, and the technical approaches to their solutions, have both taken advantage of and pushed the leading edge of data management technology. The special problems of managing science data and their associated metadata will be discussed.

In the early 1980s, influenced by the National Academy of Sciences Space Science Board CODMAC Reports, NASA funded several pilot data systems development projects to be based upon the key concept of the discipline data management unit. The data systems were organized into systems for separate science disciplines in order to serve the needs of particular science communities, and to provide each community with the data needed for its own research. These data systems included the Climate Data System, the Land Data System, the Oceans Data System, and the Planetary Data System. This paper will focus on the Planetary Data System, but the problems encountered are typical of the others as well. All were designed to service a particular disciplinary group, and all were originally thought of as being self-contained.

All of the initial pilot systems met with varying degrees of success, and are in some form operational today. The second era of data management systems, the era we are in at the moment, deals with the formidable task of integrating some of these stand-alone systems into a single service to provide data to the ever growing inter-disciplinary science research community. The broad community of earth scientists, focusing on the study of global change, is not only highly inter-disciplinary, but inter-agency, and even international. Part two of this paper will discuss some of the special needs of this research community for data, and the resultant challenges to data management technology of the data management system for the NASA Earth Observing System (EOS).

2. Case I: The NASA Planetary Data System (PDS)

Brief Background

The PDS was jointly designed by members of the planetary science community from around the country and data system developers at the Jet Propulsion Laboratory from 1985 through delivery of the initial version in 1989. This original version primarily concerned itself with science data already collected by previous NASA missions over the past two decades. Currently PDS has evolved, and continues to evolve, to archive and distribute data from current planetary missions as soon as the data are available. Work is also underway in the PDS project to plan for the archiving of data from future missions not yet flown.

The Early Vision

From early in its development the vision of the PDS was that of a system whose science data products would be localized at various planetary sub-discipline nodes and whose directory and catalog metadata would be centralized and managed at a central node at JPL. Detailed metadata about individual data granules as well as local physical metadata known collectively as inventories would be kept locally with the data products. The directory and catalog metadata would be be available to the local systems in a client/server mode so that a science user could access any of the relevant metadata from wherever he was located. The data products themselves would be labelled with self-describing metadata in a standard form and distributed by either the discipline nodes or the National Space Science Data Center (NSSDC), depending on the size of the products.

Largely because the prevailing data management technology of the 1980's was relational, the PDS metadata database was designed and built as a relational system. Important information relating to the data products about the spacecraft, instruments, investigators, processing algorithms, etc., was all organized into relations and linked by relational operators to provide an ad-hoc query capability for data access. Testbed data sets from early missions were loaded and the system was released to the community for evaluation.

The Success and the Problems

The PDS was considered quite successful at doing what it was supposed to, i.e., making planetary data available to its community. However, the task of maintaining the system in its operational mode required the loading of many more data sets. This process required the data producers to provide the rich suite of metadata, which made the PDS so useful, in a highly structured form for ingestion into a relational database. In many cases the metadata already exist in the form of documents, journal articles, or data record headers. The scientist has to rework these metadata, and in some cases do some digging, to provide the PDS with its required inputs. This has been loudly complained about. Suddenly, the grand and glorious catalog that provides such a wealth of information is being called too expensive to maintain by the very community of scientists who designed it.

One solution to this problem is the automation of metadata collection. Planning ahead for the archiving of data in the early stages of the flight project would certainly help ensure that all of the required metadata were electronically present. Efforts in this direction are currently occurring with the Mars Observer and Cassini projects. Nevertheless, it is not always practical to carry along all of the metadata required by the ultimate archive system, and it is not always possible to identify all of the relevant pieces of metadata a priori, so the problem of ingesting other metadata as the system operates seems likely to occur even so. New technologies associated with object-oriented and multimedia databases may make the native forms of the metadata (science papers, videos, software, documents) more utilizable within the data system. Other ways of linking the vast amounts of textual information (such as WAIS) and integrating this information into the data system also need exploration.

3. Case II: EosDIS (Version 0)

Data systems belonging to the second generation of NASA systems go far beyond their predecessors of the single-discipline self-contained kind. These new systems, of which the EOS Data and Information System (EosDIS) is a prime example, must necessarily, for several reasons, build upon the systems already in place. These reasons have to do with cost (it's usually too expensive to start from scratch), logistics (most scientists do not want to give up their local capabilities), and the sheer volume of the data to be encompassed.

Version 0 of EosDIS has been chartered to prototype various approaches to interconnecting the underlying data systems without disrupting service to the local users. This experience has brought to light many challenges to current data management technology.

Data System Heterogeneity

Probably the most difficult and challenging problem faced by the EOS data system developers is that of integrating widely distributed, autonomous, heterogeneous data systems into a unified whole. NASA has identified eight institutions to serve as the Distributed Active Archive Centers for the earth science data collected in the past, the present, and the future. The DAACs as they are called are either the earlier discipline data systems built a generation ago or conglomerations of these. Each has a distinct coverage of earth science disciplines. The DAACs will upgrade their own data systems to handle their new data responsibilities, and the Information Management System (IMS) component of EosDIS will integrate these DAACs into a unified whole, providing any of the data to any scientist with complete location transparency. This requirement, known as "one stop shopping" in EOS circles, unveils all sorts of issues stemming from both system and data heterogeneity.

Currently a data dictionary is being developed to document the local DAAC vocabularies so that approaches to the resolution of differences can be worked and true integration of the underlying inventories of data can be achieved.

In addition to integrating with all of the DAACs, the EosDIS also needs to couple with another data system to provide directory information to the earth scientists. The NASA Global Change Master Directory at the NSSDC is yet another source for data heterogeneity problems in that its vocabulary serves an even broader community and needs to be merged with the terminology of the DAACs.

Metadata Generation and Utilization

The problems of automating the collection of metadata and of being able to utilize data and metadata in many different forms identified earlier in the discussion of the PDS are also present in EosDIS and even more critical because of the massive amounts of data to be generated. Multi-media and object oriented technologies to deal with the variety of data forms, and intelligent systems to generate the metadata from the content of the science data are all new technologies that may prove indispensable. In addition, new approaches to spatial and temporal searches, as well as sophisticated graphical interfaces and visualization of metadata, are needed to help locate data of interest from such a huge pool.

4. Summary

Problems of managing science data and associated metadata exist in both generations of data systems, though the new systems pose challenges on a much larger scale. This paper has raised some of the more pressing issues faced by the author currently. Progress in the solutions to these issues will benefit the science data management community as a whole as I'm certain that these are not NASA or space science specific.

5. Common Problems and Topics for Discussion

Multi-media Databases and Object-Oriented Approaches for Storing and Linking Science Data and Metadata

Planning for Metadata Generation in Mission Design

Evolvable, Extensible Systems

Interoperability between Heterogeneous Database Systems

Standards for Metadata

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Metadata in Archives: The Atmospheric Radiation Measurement (ARM) Archive Experience

Paul T. Singley and Patricia F. Daugherty
Environmental Sciences Division
Oak Ridge National Laboratory*
October 6, 1992

Introduction

Webster's Ninth New Collegiate Dictionary defines an archive as a place where public records are preserved. All too often, modern-day computer archives are places to preserve project data but they fail to provide for the easy access and use of the data by their constituents. Often the major obstacle to broader use of data contained in an archive is a lack of good quality metadata that would allow researchers to use and analyze data from the archive. Metadata are simply data about data. All archives need some sort of metadata to allow access to data. Scientific archives, especially archives where the majority of the data comes in the form of data streams from instruments, have their own types of necessary metadata. In a research archive the metadata must be very broad and complete, because the data are often used to pursue questions other than what it was originally collected for, and the users of archived data will not be as familiar with the data and the circumstances the data was developed under as the initial "target" research community. This paper will discuss issues surrounding the development of the proper and sufficient metadata for an archive to facilitate access and use of its data by scientists who are unfamiliar with the data, and who may be using the data in new and unanticipated ways. We will use our experience in designing and developing the Atmospheric Radiation Measurement (ARM) Archive as an illustration.

The ARM Archive

The ARM Program is a Department of Energy (DOE) sponsored global change research effort designed to improve the modeling of cloud radiative forcing in General Circulation Models (GCM) (Atmospheric Radiation Measurement Program Plan, Feb. 1990). ARM will provide an experimental testbed for studying atmospheric effects important to radiative processes. This testbed, called the Cloud and Radiative Testbed (CART), is expected to consist of five long-term data collection sites located worldwide. The first site, in Lamont, Oklahoma, began operation in June of 1992.

The CART Data Environment (CDE) is the component of the CART that acquires the basic observations from the instruments at the ARM sites, processes the data to meet ARM requirements, distributes the data to the ARM Science Team, and archives the data. The formal logical requirements for the CDE are discussed in Melton, et. al. (1991 and 1992).

The CDE consists of three major physical systems.

- The Site Data System, located at each of the field sites, provides the tools for ingesting data from the instruments into the CDE, controlling the operation of the instruments, and assisting the site operator.
- The Experiment Center (EC), located at Pacific Northwest Laboratory (PNL), is where higher levels of data Quality Assurance (QA) are performed, and where data from the sites and other sources, such as satellites, are incorporated into the measurements required by the ARM Science Team.

* Managed for the U.S. Department of Energy under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

- The Archive, located at Oak Ridge National Laboratory (ORNL), is where data are archived, and provided to both the ARM Science Team and the general scientific community for retrospective studies.

The ARM Archive is similar to many large scientific archives in operation or under development. However, there are several aspects of the ARM Archive mission that make it unique. As with most scientific archives, ARM data and metadata are primarily derived from automated instrument systems, with some critical components of the metadata developed by human input to the data system. Also, in common with most other archives, the ARM Archive has the mission to provide data to both a target user community, in this case the ARM Science Team, and to the broader research community.

The ARM project is somewhat unique in that while the ARM Archive is the direct source of data for the general research community, the ARM Science Team requests for retrospective data will be processed by the EC, which will retrieve the requested data from the ARM Archive for the Science Team member. Another way in which the ARM Archive differs from many environmental archives is that most of the data will not come from imaging instruments, but instead will be numeric data streams. Imagery will dominate the data volume, but the preponderance of the ARM data sources will be generating streams of numbers taken at discrete time points. This will drive up the metadata volume in the ARM Archive, because each time point can have several metadata items, such as multiple QA flags, associated with it. Under these circumstances it is possible to have metadata streams larger than the data they describe. In addition, the ARM data are typically packaged in small files. This may be less of a metadata issue and more of an architectural issue for the design of the ARM Archive, but it will have implications for the association of metadata and corresponding data. Finally, unlike many archives, the ARM Archive will have many data sources. Not only are there a large number of data gathering instruments at the sites, but there are other data sources spread throughout the CDE. A short list of some of the sources and the associated data and metadata will serve to illustrate this point.

- Data files from the ARM Sites contain data generated by the instruments, e.g., Radio Acoustic Sounding Systems (RASS), wind profilers, and radiometers, plus metadata about the data gathered at the ARM Sites.
- The ARM Sites' Site Operations Logs (SOL) contain information about the day-to-day operations of the ARM Sites, e.g., when an instrument went on-line, when an instrument was calibrated. These logs are purely metadata.
- The Technical Database (TDB) includes metadata-only files that contain detailed descriptions of the instruments at the ARM Sites. These will contain technical information about the instruments, e.g., the manufacturer and calibration curve for a particular instrument.
- Data files from the ARM EC include the results of data fusion and analysis performed at the EC, as well as metadata about those data. Some of these files will contain only metadata because the analysis will be a complex QA check of a data stream coming from a site.
- Program and project documentation, including project plans, experiment plans, and other documents describing the scientific purpose of the ARM project, are human generated and not specifically intended as metadata (unlike the SOL and TDB), but are vital for understanding the context in which data was acquired.

In the ARM context, metadata includes metadata contained within the files sent to the ARM Archive and metadata generated by the ARM Archive. This second category of metadata includes such information as:

- file name,

- file size,
- date and time the file was received, and
- date and time when the file was placed on tape in Mass Storage System.

This metadata are most useful in managing the operations of the ARM Archive.

Uses of Metadata in the ARM Archive

We have noted that metadata are essential for future understanding and use of the data. The developers of an archive need to know how the data and metadata may be used so they can select the proper metadata for the archive. In this section, we will briefly discuss the major uses of the metadata in the ARM Archive. These uses are general enough to be common to all archives.

Making Information Out of Data Streams

At a fundamental level, metadata are required to provide meaning to a data stream coming off an instrument. This is a requirement both for immediate use of the data as well as for archival use. However, when the data is fresh and users are very familiar with the system that is producing it, there may be little perceived need to formally record this type of information, and the temptation is to forget that it must be provided for future use. Each type of data can have a different set of required metadata. For most of the ARM observation data streams the required metadata are:

- observation time,
- observation location,
- instrument identification,
- physical phenomenon being measured, and
- units of measurement.

Assuming these metadata are available, other metadata are often necessary for turning bits into meaningful data, such as instrument calibration and instrument operating state.

Describing Archive Contents

Descriptive information about the contents of an archive is necessary for users to understand the type of data that an archive holds. The ARM Archive design calls for a Users Guide that will contain information about the data and metadata that the ARM Archive holds. In addition, it will contain information about available data formats so that users can select a format that is easy to incorporate into their analysis system. Finally, it will contain a Users Manual that describes how to use the available tools to access the ARM Archive. Most, if not all, of this metadata will be developed by humans. Much of the data will be in free-text format with a text querying system to assist the users in finding topics of interest. The description of the data and most of the metadata data will also be in a formal data dictionary, and managed using a Relational Database Management System (RDBMS).

Querying and Analysis

The metadata required for querying and analyzing data blend into each other because the activities of querying and analysis are not truly separate activities. Most ARM metadata will be available to the users for analytical purposes. When a user requests data, there will be a certain set of quantitative metadata automatically shipped with it. Also, the user will be able to request additional metadata such as more extensive quantitative metadata (e.g., a specific set of QA flags) and/or qualitative metadata (e.g., a photograph of the instrument on location). In addition, a user will be able to request only metadata, (e.g., an experiment design or a site map).

Within the metadata held at the ARM Archive we have identified a subset that is specifically useful in helping the user formulate on-line queries for data and metadata from the ARM Archive. These metadata will be used to qualify queries for data, and will contain items such as:

- phenomenon being measured,
- location identifier,
- instrument identifier, and
- QA flags.

These "query metadata" are managed in a RDBMS, and will be used to support an extensive on-line user interface.

Additional Metadata Issues

An understanding of the uses for metadata will assist in determining which metadata are necessary. However, there are other issues in developing metadata that will be truly useful for a broad range of research. This section discusses some of these issues.

Adequate Metadata
The first problem in maintaining adequate metadata is to convince those gathering data to gather metadata at the same time. This is equivalent to maintaining a good lab notebook, one that allows a scientist to understand an experiment years after it was conducted. There is the temptation the generate metadata after the experiment is completed, but no one's memory is good enough to do this adequately.

The Right Metadata
Once the data gatherer is convinced of the necessity of gathering metadata, many issues arise. To take a very simple example, it would be useful to know where an instrument producing data is located. Ways to record location include:

- latitude/longitude, UTM, or State Plane coordinates,
- a dot on a 71/2 minute topographic map,
- a dot on an aerial photograph,
- a shaded area on a county property map,
- a zip code,
- a written description of the location, given in paces from a particular fence post, or
- road directions starting from an exit on an Interstate highway.

Location metadata could also be combinations of the ideas listed above.

When evaluating the possible metadata the following issues should be considered:

- Is it reproducible?, In this case, is it likely that two people given the same directions will end up at the same place?
- Is it precise enough? Are the appropriate units of measure being used? In this case, latitude/longitude coordinates would be good; a zip code would not be.

- Is the metadata identifier used to mark a location unique? Two different instruments having the same identifier is worse than having no identifiers, because in files containing the identifiers, the two instruments would appear as one.
- Who is going to use this metadata? Using a coordinate system that no one else is familiar with will limit the metadata's usefulness.
- How will this metadata be used? If the location metadata is going to be used only to tell the site operator how to get to the instrument, paces from a numbered fence post might be very useful. If this location is also going to be used to provide ground-truthing for satellite imagery, then a coordinate system such as latitude/longitude would be much more appropriate.
- How is this metadata going to be gathered, and by whom? The instrument may be marked on a topographic map by a technician working at the site. This location could then be converted to latitude/longitude coordinates through a Geographic Information System (GIS), or it could be gathered directly through the use of a Global Positioning System (GPS).

As shown above, gathering and maintaining even the simplest metadata can be complex and time-consuming. It is therefore understandable when scientists balk at dealing with metadata. There are concrete payoffs from making the effort to keep good metadata. In the example given above, good location information will save time and money when the instrument must be repaired or visited for some other reason, will allow the instrument to be used for ground-truthing, and most importantly, will lessen the chance of bad conclusions being drawn when data from the instrument is analyzed.

Because these metadata are so important to the usefulness of the associated data, systems should aim to automate the acquisition of these metadata to the greatest extent possible. Automation will help ensure that these critical metadata are collected, and that they are of consistent quality.

Too Much Metadata

It is possible to have too much metadata. This occurs when project participants send all files and documentation, in an effort to avoid not having enough metadata. The problem with this scenario is that it makes it very difficult to find the useful metadata within all the digital and manual files. In this situation it becomes the responsibility of the archive to weed through what has been sent, but archive staff will not have the same understanding of the relative usefulness of what they've received as that of the sender.

Integrating Metadata and Data

Another problem with metadata comes in combining data with metadata. For example, the instrument discussed above could be a thermometer, which creates a digital stream of temperatures in degrees Celsius. This stream of temperatures is divided into separate files, with a single file containing one day's temperatures. These files are sent to the Archive daily.

The location metadata for this instrument could include latitude/longitude coordinates stored in a digital file of coordinates for all ARM instruments. It could also include a topographic map and an aerial photograph of the site which have each instrument location marked with a dot. This location metadata is sent to the Archive once, and then updates are sent sporadically.

In this scenario, how can a single day's temperature file be linked to the digital file of instrument coordinates, or worse, to the manual map and aerial photo? How will a person looking at the file of temperature data know that location data for the instrument even exists?

In this case, the simplest answer is to make sure that each instrument has a unique identifier that is used consistently. For example, the file of temperature data should include the unique identifier for the instrument that gathered the data, and the dots on the map and aerial photograph should be labeled with this unique identifier. This would solve the problem of linking a temperature file to the location data, but would not solve the problem of letting the user of the temperature data know that location metadata for that instrument exists.

There are also levels of linking. In this case, linking data and metadata may be as simple as letting a user know that a map of the site can be found in a manual map file. At the other extreme, the user may need a report with both the temperature data and a map of the site showing the location of the instrument on the same page.

Cost

Metadata consume valuable resources within the data system, such as transmission bandwidth, storage space, and personnel. These resources can be expensive. Ways to minimize these costs include putting careful thought into the design and development of data systems to handle metadata, and exercising discipline in the recording of metadata, especially those generated by humans.

The alternative would be not to give metadata the necessary attention, which could lead to not being able to use the data in the archive. This alternative is too costly, both financially and politically, for most organizations.

Summary

Modern scientific research archives are charged with storing data and metadata from many sources, with organizing both the data and metadata to allow users to make a broad range of queries, and with describing the archived data well enough to make it useful. This discussion of our experiences with the ARM Archive has illustrated some of the uses of metadata, and some of the issues surrounding developing and managing metadata in an archival setting.

References

Melton, R.B., et al. 1991. "Design of the CART Data System for the U.S. Department of Energy's ARM Program". Proceedings of the Seventh International Conference on Interactive Information and Processing System for Meteorology, Oceanography, and Hydrology. January 14-18, 1991. New Orleans, LA, American Meteorological Society, Boston, MA.

Melton, R.B., et al. 1992. "Clouds and Radiation Testbed Data Environment: Site Data System and Experiment Center". Proceedings of the Eighth International Conference on Interactive Information System for Meteorology, Oceanography, and Hydrology. January 5-10, 1992, Atlanta, GA. American Meteorological Society, Boston, MA.

U.S. Department of Energy (DOE). 1991. "Atmospheric Radiation Measurement Program Plan". DOE/ER-0441, Washington, DC.

Wind Profiler Demonstration Network Metadata Access System

Marjorie P. McGuirk

NOAA Environmental Research Laboratories
Forecast Systems Laboratory
Boulder, Colorado 80303

Susan M. Williams

Cooperative Institute for Research in Environmental Sciences
University of Colorado/NOAA
Boulder, Colorado 80303

Wayne Faas

NOAA National Climatic Data Center
North Carolina 28801

1.0 INTRODUCTION

Metadata, the descriptive information about data, is traditionally collected manually and oftentimes lost altogether. When metadata are missing, data may be useless (W.R. Moninger 89). Capturing metadata in an organized, efficient way becomes more important as NOAA deploys more data collecting platforms. This document gives an overview of how the metadata for the Wind Profiler Demonstration Network (WPDN) are collected, processed and transmitted to a central location, and accessed by the research community. It demonstrates the feasibility of capturing metadata automatically from new platforms.

2.0 BACKGROUND

The WPDN comprise a network of upward-pointing Doppler radars deployed by the Forecast Systems Laboratory (FSL). The radars measure the turbulence in the atmosphere through the use of an electromagnetic wave transmitter and a receiver; the returned signals are processed by sophisticated algorithms that produce measurements of horizontal and vertical winds in the atmosphere above the radar.

Data are received in 6 and 60 minute intervals. The data from the WPDN allow forecasters and researchers to see the structure of the atmosphere in great detail both temporally and spatially. The Management of Atmospheric Data for Evaluation and Research (MADER) project [McGuirk and Crowe, 1991], a cooperative effort between FSL, the National Climatic Data Center (NCDC), and STORM, provides on-line near-real-time wind profiler data and metadata to the research community. It is an efficient, integrated, automated system for handling data and information, responsive and accessible to users. Data are processed and transmitted from the Profiler Control Center (PCC) Hub in Boulder to the NCDC in Asheville, where additional data processing is performed; from there the data and metadata become accessible to the research community though a computer referred to here as the Access Server.

3.0 METADATA TYPES

The types of metadata captured from the WPDN are station history, data dictionary, and inventory. Additionally, a unique feature of MADER planned for late 1992 is a meteorological event directory. Descriptions of each type of metadata follow.

3.1 STATION HISTORY

Station history metadata include the geographical and observational parameters for a wind profiler station:

- Station configuration - the station name, the NWS five digit identifier, latitude and longitude (CLI) of the site, the date the station began operating, and the version number of the software algorithm running at the wind profiler site.

- Beam information - the azimuth angle and elevation angle for the beams of the radar. Each radar has three beams, one vertical and two oblique (east and north).

- Meteorological instruments - the meteorological parameters measured, the instrument type, manufacturer and model number, the placement of the instrument, and what quality control practices, if any, are applied to the instrument.

- Quality Control Element Extremes - global maximum limits for six and sixty minute rainfalls, and limits for the minimum and maximum temperature extremes for all months of the year.

- Clutter flag data - the begin and end date and time for the flag setting, the beam, the power mode, and the flag setting for each gate. Each radar has 36 gates and two power mode settings (high and low). Each gate has a clutter flag setting.

- Physical plant changes - any physical plant changes or remarks regarding the station. For example, if chicken wire were placed around the profiler to suppress clutter, this information would be noted here.

- Line Replaceable Units (LRU) - metadata about items such as beam steering units (BSU), receivers, and power amplifiers are tracked here.

3.2 DATA DICTIONARY

Data dictionary metadata include definitions of all the parameters in the 6 and 60 minute time resolution datasets, such as units, quality control information, and data codes. Data from the WPDN are divided into 8 datasets:

- Moments_60 - the hourly averaged 0th, 1st, and 2nd moment information for all three beams, two modes, and 36 gates of the profiler.

- Moments_6 - the 6 minute 0th, 1st, and 2nd moment information for all three beams, two modes, and 36 gates of the profiler.

- Control_60 - hourly status information on engineering and communications parameters.

- Control_6 - 6 minute statuses for engineering and communications parameters.

- Surface_60 - contains hourly averaged surface data.

- Surface_6 - surface data that is averaged over the previous 6 minute time interval.

- Winds_60 - winds averaged over an hour time period for the components in the u, v, and w directions. Also included are the quality control indicators for all levels of the wind profile.

- Spectrum_6 - a diagnostic spectrum which can rotate any 6 minute period to any beam, gate, or mode.

The eight datasets listed above include all data elements collected by the WPDN.

3.3 INVENTORY

Inventory metadata are counts of all data elements in each dataset by time period and wind profiler. For the WPDN, eight inventories are created, six hourly and two daily. They include the following:

- MOMENTS_60 - a count (0-64) of the minimum number gates across the three beams where the consensus (quality control algorithm) was passed, indicating moment data exist. The eight gates (a total of 36 gates per profiler) in the overlap region between low and high mode are counted if either the low or high mode gate passed the consensus .

- MOMENTS_6 - a count (0-10) of 6 minute periods within the hour when the profiler receiver was on.

- SURFACE_60 - using the BUFR (Binary Universal Form for the Representation of meteorological data [Brazille, W., 1991]) classes and elements, it indicates if the element existed during the hour.

- SURFACE_6 - like SURFACE_60, using the BUFR classes and elements to indicate if at least one reading of each element existed during the hour.

- WINDS_60 - One bit indicates BUFR class indicator. The remaining seven bits are used as a count (0-64) of the number of gates where wind data exist. The overlap region between high and low mode is handled the same way as with MOMENTS_60.

- SPECTRUM_6 - a count (0-10) of 6 minute periods during the hour when the receiver was on and the profiler acquired a spectrum.

The daily inventories include:

- CONTROL_60 - a count (0-24) of 60 minute periods during the day for which control information exists.

- CONTROL_6 - a count (0-240) of 6 minute periods during the day for which the control information exists.

3.4 EVENT

The event directory [McGuirk, 1991] contains metadata as pointers to meteorological events. Often researchers are interested in datasets for a specific meteorological event such as tornadoes, blizzards, floods, etc. The event directory simplifies data searches by allowing researchers to specify a category of events. Three pieces of information comprise event metadata; the type of event, geographic location of the event, and the time the event occurred. Through the relational pointers of the DBMS, these pieces of information are linked to the station history and inventory metadata described. So through a DBMS query, users may find all stations with data in a particular dataset for the time and location of a selected event.

4.0 METADATA COLLECTION

The metadata collection for the wind profiler sites begins as soon as the profiler has been accepted by NOAA and released by the Profiler Control Center (PCC). The PCC evaluates the profiler data and once satisfied that quality control standards have been met, releases data to the National Weather Service (NWS) and begins collection of the metadata. The PCC utilizes the database management system DataEase, a DataEase International product, and manually enters metadata changes as they occur.

In real-time operations, to begin early 1992, a process will transfer files from DataEase to the PCC's Hub computer. From there the files will transfer to the Access Server and be ported directly into an Empress Data Base Management System (DBMS).

All the backlogged metadata collected from January 1, 1991 until real time transferred occurred, was initially imported into the Empress.

5.0 METADATA PROCESSING

Metadata values are updated as they change. Dynamic parameters include station history information, data dictionary, and inventory. The Hub processes the changed information from the profiler and creates an entry in a change file. Information in this record includes the station identification of the profiler affected, the effective date and time of change, the parameter that changed, the Empress table that is affected and the new value of the parameter. The metadata changes are collectively placed on the Hub in a single file in ASCII format, awaiting transmission to NCDC.

The metadata change file is transmitted to NCDC. Through a UNIX C process, the file is divided into multiple files, one file per EMPRESS table. Next Structure Query Language (SQL) commands are added to the file; the commands are executed and the Empress tables are updated.

Current records for metadata are recognized in the Empress DBMS by an end date of 31 December 9999 and an end time of 24:60:00. With these criteria, searching for the current records in a table is simplified. Each time a record is updated in the database from change data transmitted from the PCC, the existing current record for that parameter, site, and selection criteria is updated to reflect the change date and a new record is created with the current data. Each profiler is identified in the database by its Common Location Identifier (CLI), a shorthand latitude and longitude notation. The CLI is the primary station identification as well as the relational key that links the station history tables in the Empress DBMS [STORM Project Office, 1991].

6.0 METADATA TRANSMISSION

The transmission of metadata along with the actual wind profiler data is a new concept of the MADER-91 project [McGuirk and Crowe, 1991]. It gives the user a complete history of the configuration of a wind profiler site, what elements are collected, and inventory information, along with the actual data.

Transmission of metadata from the Hub to the Access Server involves three different computer systems: the Hub in Boulder, Colorado, and the Ingest and Access Server at NCDC in Asheville, North Carolina [MADER 91 Development Team, 1991].

The Hub is a VAX/VMS cluster. It controls each stations's configuration and it formats the data and metadata for output. The Hub is equipped with Multinet communications software. Wind profiler data are transmitted hourly to the Ingest machine over Internet using SUN Remote Procedure Calls (RPC). The metadata change file, described in section 5.0 is transmitted from the Hub to the Ingest machine daily.

The Ingest System is a Sun SPARC Station IPC with 16 MB of memory, a 400 MB hard disk, and a CPU on the order of 15 MIPS. It operates under UNIX and is equipped with a 3480 compatible drive and an Uninterruptable Power Supply (UPS). The Ingest machine collects the files from the Hub, prepares files for the Empress DBMS on the Access Server, produces summary products, and writes files to tape. Files are transmitted from the Ingest machine to the Access Server by means of RCP, Remote Copy.

The Access Server is a Sun SPARC Station 2 running UNIX. It has 32 MB of memory, a 2.0 GB hard disk, and a CPU on the order of 30 MIPS. It acts as a file server and as an access machine for users. All on-line data files reside here and all user interaction will be through this machine.

7.0 METADATA ACCESS

The metadata for the WPDN in the Empress DBMS are accessible directly through Internet or indirectly through the STORM dataset guide [STORM Project Office, 1991]. The MADER system will be available for use for STORMFEST in February 1992. Alternately, users may access information by a data request to NCDC, either on the telephone or in letters.

Accessing metadata through the Access Server directly or indirectly will allow researchers several menu options. The researcher may search the data set guide, station information, inventories, event directory, and data dictionary. In addition, users may browse on-line profiler data. Access through the STORM dataset guide will give STORM users special ordering privileges. [STORM Project Office, 1991].

To utilize the MADER menus, users must have, at a minimum, a monitor, a keyboard, and communications, either a MODEM or access to Internet. To take full advantage of the system, users should have a computer running X-11 Windows.

Secondly, metadata may be accessed by requesting information directly from research assistant in Asheville, who will utilize the MADER system, providing small amounts of information by phone. Larger volume metadata requests will be supplied on removable media.

8.0 SUMMARY

The WPDN provides high spacial and temporal resolution wind data. Metadata from the WPDN enchanced the value of the data itself by giving researchers easy access to information such as station configuration changes, and inventory counts by dataset. The event directory furthermore provides a quick pointer to datasets in existance during specified meteorological cases. Collecting, processing, and transmitting metadata from the WPDN, and providing access to the metadata in the MADER system, points the way for handling metadat for future observing platforms.

9.0 REFERENCES

Brazille, Wayne, 1991: Data Exchange Format Evaluation and Recommendations (Second Draft), STORM Project Office, National Center for Atmospheric Research, Boulder, CO.

MADER 91 Development Team, 1991: MADER 91 Requirements Specification (Draft Revision 0.5, June 6, 1991, Forecast Systems Laboratory, NOAA, Boulder, CO.

Moninger, William R., 1989: Earth Observations and Global Change Decision Making, edited Erving W. Ginsberg and J. A. angelo, pub Krieger Publishing, Malibar Florida, Pg. 233-237.

McGuirk, M. P. and Crowe, M., 1991: Wind Profiler Demonstration Network Data Management Activities,NOAA Environmental Research Laboratories, Forecast Systems Laboratory, Boulder, CO.

STORM Project Office, 1991: MADER-91 STORM / NCDC Metadata Access Module Design Documentation (Sixth Draft), 19 April 1991, National Center for Atmospheric Research, Boulder, CO.

Metadata Applications in Weather Data Archival, Retrieval, and Visualization

Ruth Anne Ross and Stephen Jascourt
Atmospheric and Oceanic Sciences
University of Wisconsin-Madison

Our synoptic meteorology program has recently obtained funds to create a "state-of-the-art" computing laboratory, digital synoptic data archive and interactive case-study facility for instructional (including student research) purposes. Most of the hardware and systems software have been installed--that was the easy part. The difficult hurdle--and exciting opportunity--will be to design an effective system of offline archival and online information about that archive that can together provide a flexible and informative interface for data discovery and location, data access and manipulation, and dataset composition for synoptic case-study development.

Much of the synoptic data that we are ingesting and are now beginning to archive was actually collected for operational purposes (by the National Weather Service, the military). Up until now, that is how we have chiefly used it, keeping only a few days' worth in digital form for analysis and weather forecasting and keeping a year's worth of paper maps and data printouts for constructing case studies, some of the more interesting of these to be kept indefinitely. From this perspective, it could seem reasonable that information about which instruments were used, changes in location of instruments, conditions affecting measurement, methods of data calibration/validation, and other information needed to intelligently evaluate/interpret data is provided in bulletins that are not automatically attached to the data. The weather forecaster learns to keep this information in mind while analyzing recent data in order to produce reasonable results. And it could be observed that the best case studies may tend to be those that are assembled on the spot while all the relevant "metadata" is at hand.

Now consider the situation of archiving the ingested datasets for future research. As has been pointed out by two NOAA researchers [Schwartz and Doswell 1991], "hopeful assumptions about the quantity and quality of operationally collected data are almost never justifiable," and trying later to get station history information, even in nondigital form, can be difficult to impossible. Yet responsible researchers have to make every effort to know about changes in instrumentation, data reduction, reporting and coding practices, frequency and time of observations, location of stations, and data-archiving procedures. In an instructional research setting it is important that students learn to use this knowledge to inform their analyses and conclusions. It has been said that even for the research meteorologist or climatologist, it is only recently that there has been much attention paid to this important issue [Elliott and Gaffen 1991]. Possible explanations include the fact that it has only been since about 1980 that we have had a long enough record of sounding data available for longterm climate or weather studies [Schwartz and Doswell 1991]. And this data, taken largely for weather forecasting purposes, is now being used in research on global warming and detection of climatic change, in spite of a frequent lack of knowledge of the conditions of its collection. As Schwartz and Doswell speculate, this is perhaps because there is no alternative.

With the growth of interest in studying global climate change, the meteorological data archives will be receiving increased attention. For data interpretation to be at all valid, it is essential that these archives hold not only the data but also the context of its collection and that this context can always be found for the data of interest [Keune 1991]. Furthermore, given the expected increase in the quantity of data to be collected, and the difficulty of finding qualified data for a particular study among the large number of datasets available, the potential use of metadata to help us navigate an archive effectively may be a real bonus.

Although these two justifications for maintaining usable metadata (preserving the context of data collection and facilitating the location of appropriate data) are in themselves sufficient for a major effort to be launched, there are still additional benefits to be had. Metadata could also be used to allow the coupling of datasets for graphical display and visual analysis [Hibbard and Santek 1990] or to visually examine the data structures or completeness of the datasets [Hibbard et al 1992]. Metadata would also make it easier to produce, and perhaps to store as metadata, computed summary data of various types. Metadata could be used to support the construction of derivative datasets/variables including transformations into alternate coordinate systems, such as producing datasets in isentropic coordinates from height, pressure, or sigma coordinates. (It should be noted that isentropic, sigma, and pressure coodinate systems vary depending on the data, whereas height coordinate data is in a fixed coordinate system independent of the data. For example, a 500 millibar vertical coordinate will not correspond to any particular height but will represent all heights where the pressure is 500 millibars.)

Looking ahead to future benefits, one can guess that metadata could be used to support some real-time querying of large off-line archives and even be used to provide the semantics for natural language access [Radermacher 1991]. Metadata systems could be used to facilitate the access and use of large datasets over networks or from non-writable devices (as CD-ROMs). Now, users who want to manipulate this data in various ways must download huge datasets. Metadata could be used to "attach" correction/adjustment factors or other types of alterations to datasets without making copies of the primary data. And metadata could be used as a mechanism for organizing a set of entities and of transmitting or transforming that organization, as a sort of virtual dataset, independent of the actual location and ownership of the entities themselves [Ross and McCormick 1990]. Ultimately, metadata could provide knowledge about the data necessary to begin automation of data exploration as software systems are created to sift through terabytes of data to find interesting things for human researchers to study. One can easily conclude that we can simply not afford to proceed without metadata. Its value in the short run will easily repay its cost and its value in the long run may be tremendous.

Getting "back to earth," in our development project we hope to create a system that will ensure that the datasets that we are archiving will not lose their contexts of collection. This is our primary goal. If we fail here, the science in the project is lost. We plan to store this context on each tape that we archive. Some of the information (dimensionality of variables, notes on conditions of collection, quality of data or missing data) will be stored within each dataset. For this we plan to use a self-describing data format to allow embedding some of the metadata in each file. We are currently planning to use the NetCDF format [Rew and Davis 1990], which was based on the CDF format [Treinish and Gough 1987]. We have also been considering the Candis format [Raymond 1988] and the HDF format [NCSA 1990]; see Granger-Gallegos [1992] for a discussion of the use of HDF for large geophysical datasets. Each has its advantages and its limitations. Other information, such as observing station histories (types of instrumentation, elevation, changes with dates) or relevant contemporary events, will be stored separately on the tape and will be referred to by codes in the self-description part of each dataset for which it is relevant.

An important secondary goal of our project is to create an online metadatabase that can be queried to help locate qualified data of interest in the offline archive. Different kinds of data for the same time period or similar events could be linked. The ability to determine data completeness (instrument failure often creates holes in our data) from the online system before going to the tape archive would be a great help. We want the metadata to distinguish between data that we are missing and data that does not exist anywhere. All this could save the researcher's time and make sure that items are not overlooked. Eventually, we want to build an interface to support the construction of case studies for instructional use. Such an interface would be a great help to those teaching synoptic meteorology, an increasingly difficult task which is currently attracting fewer and fewer meteorologists. We envision a system that would ease the burden and make creating good case studies practical and enjoyable.

We also want our system to support the process of scientific visualization. The importance of data management for a data visualization environment has already been elaborated [McCormick et al 1987, Treinish 1991]. Metadata could be used to support a browse mechanism with the visualization of data subsets (reduced resolution with full coverage or full resolution with reduced coverage) to help the user identify which datasets are to be requested from archive. And it could provide default methods for displaying the dataset to allow effective use of the system without a complete knowledge of all the options. These themes of interactivity, ease of use, and versatility have been central to the development of the VIS-5D visualization software [Paul et al 1993, available, as is the VIS-5D software, from Hibbard or Paul upon request], used in our lab for interactive visualization of large atmospheric datasets. Effective use of metadata could further support these objectives.

Some of our metadata, such as descriptive fields, may have to be handcrafted. In the early stages, perhaps much of the metadata will require this. But this would be too time-consuming for anything other than a prototype. Some metadata may be extracted from contemporary datasets, such as using a keyword search of forecasts or severe weather warnings from FAA broadcasts to produce the contemporary event field in other datasets. It should be possible to calculate some metadata, particularly summary statistics, correction factors, coordinate transforms. And some metadata items may be automatically constructed, as lists of missing data or subset datasets for use during online browsing.

We plan to store original data (rather than corrections we may make), but we would want to include (as metadata) correction/adjustment factors (with some context for them). Some types of metadata will go into the header of the dataset itself; some will be in separate files replicated on each tape. And most, if not all, of this metadata will also be available online, probably in a relational database system, so that we can query it.

We will maintain about a week of primary data online before it is archived onto tape. It would be nice to have an online database system with this primary data as well as metadata. The datasets include image data and some very large gridded datasets, and relational database management systems tend not to perform well on such datasets. It may be possible to make use of an extensible DBMS, such as Starburst [Haas and Cody 1991] which aims to accommodate digitized image data and spatial data as in maps as well as sensor data. We are attracted to the idea in Starburst of the

DBMS as "integrator" with data from different sources unified under a common, optimized query language. We hope to set up at least some sort of automated "system" for sharing online data among our Unidata-based "Local Data Manager" [Unidata Program Center 1991b], our McIDAS-X interface, and our own metadata manager. As far as the communication between our metadata manager and our tape archive, for the foreseeable future, that will need to be "sneakernet."

The report of a previous workshop on scientific database management [French et al 1990], identifies metadata as a central issue. Its effective use could help address some of the other issues identified, for example, locating data, creating effective user interfaces, reducing dataset transmission requirements, converting datasets to other formats, and assessing dataset quality. Otherwise, we may end up with bigger and bigger "graveyards of data" [Radermacher 1991]. Or we may end up through ignorance misusing the data to arrive at totally incorrect results with misleading implications for action. That is too large a price to pay for not developing the means for preserving and providing access to the metadata that can inform the use of primary data collected at considerable expense.

References

Elliott, W. P. and D. J. Gaffen, 1991: The utility of radiosonde humidity archives for climate studies. Bull. Amer. Meteor. Soc., 72, 1507-1520.

French, J., A. Jones and J. Pfaltz, 1990: Summary of the NSF Scientific Database Workshop. IEEE Data Engineering, 13, 3(Sept), 55-61.

Granger-Gallegos, S., A. Pursch, R. Kahn and R. Haskins, 1992: Automatic data distribution for a large, geophysical data set. The Earth Observer, 4, 4(Jul/Aug),9-22.

Haas, L. M. and W. F. Cody, 1991: Exploiting extensible DBMS in integrated geographic information systems. Advances in Spatial

Databases, SSD '91 symposium proceedings published by Springer-Verlag, 423-450.

Hibbard, W., C. R. Dyer and B. Paul, 1992: Display of scientific data structures for algorithm visualization. Visualization '92 (October).

Hibbard, W. and D. Santek, 1990: The VIS-5D system for easy interactive visualization. Visualization '90, IEEE, 28-35.

Keune, H., A. B. Murray and H. Benking, 1991: Harmonization of environmental measurement. GeoJournal 23, 3, 249-255.

McCormick, B. H., T. A. DeFanti and M. D. Brown, 1987: Visualization in Scientific Computing. NSF workshop report reprinted as special issue of ACM Computer Graphics
21, 6(Nov.).

NCSA Software Tools Group, 1990: Hierarchical Data Format, National Center for Supercomputing Applications, Champagne, IL.

Paul, B., A. Battaiola and W. Hibbard, 1993: Progress with VIS-5D/distributed VIS-5D. To be presented at AMS 93 (January).

Radermacher, F. J., 1991: The importance of metaknowledge for environmental information systems. Advances in Spatial Databases, SSD '91 symposium proceedings published by Springer-Verlag, 35-44.

Raymond, D. J., 1988: A C language-based modular system for analyzing and displaying gridded numerical data. Journal of Atmospheric and Oceanic Technology, 5, 501-511.

Rew, R. K. and G. P. Davis, 1990: NetCDF: an interface for scientific data access. IEEE Computer Graphics & Applications, 10, 4(July), 76-82.

Ross, R. A. and B. H. McCormick, 1990: VUE: a software architecture for televisualization. Technical Report TAMU-90-011, Computer Science Department, Texas A&M.

Schwartz B. E. and C. A. Doswell III, 1991: North American rawinsonde observations: problems, concerns, and a call to action. Bull. Amer. Meteor. Soc., 72, 1885-1896.

Treinish, L. A., 1991: SIGGRAPH '90 workshop report: data structures and access software for scientific visualization. ACM Computer Graphics 25, 2(April), 104-118.

Treinish, L. A. and M. L. Gough, 1987: A software package for the data independent management of multi-dimensional data. EOS Transactions, American Geophysical Union, 68,
633-635.

Unidata Support Staff, 1991a: Site Manager's Guide for the Unidata Scientific Data Management System, Unidata Program Center, UCAR, Boulder, CO.

Unidata Support Staff, 1991b: NetCDF User's Guide: An Interface for Data Access, Unidata Program Center, UCAR, Boulder, CO.

WSR-88D Data Recording and Data Management

Tim Crum
Chief, Applications Branch
WSR-88D Operational Support Facility
1200 Westheimer Drive
Norman, OK 73069

405-366-6530, fax 405-366-6555
OMNET (T.CRUM)

1. Introduction

The network of Weather Surveillance Radar - 1988 Doppler (WSR-88D) Systems provides, as never before, the opportunity to remotely sense and study the atmosphere. Data from these Systems have spatial and temporal resolutions similar to meteorological radar data sets that have been previously collected only during limited field experiments. The NEXRAD agencies, the WSR-88D Operational Support Facility (OSF), researchers, instructors, and other groups will need WSR-88D data for a variety of purposes, but not always in real time. Recent advances in digital recorder technology now permit reliable and affordable recording of the large volume of base data WSR-88D Systems produce. Interdisciplinary applications of WSR-88D data have led to widespread interest in the data. The Level II data archive probably will be the largest single-sensor data set in the National Oceanic and Atmospheric Administration within a few years. Maintaining access to these data and the metadata describing them will be essential. The initial archive capability is now being developed and further changes to the archive to meet user needs are planned.

2. WSR-88D Background

a. The NEXRAD Program

The WSR-88D System is the product of the NEXRAD Program. This joint effort of the Departments of Commerce (DOC), Defense (DOD), and Transportation (DOT) is now a reality as installation of the Program's 159 radars is underway. This System will replace non-Doppler meteorological radars currently employed by the National Weather Service (NWS), Air Force, Naval Oceanography Command, and the Federal Aviation Administration (FAA).

b. The WSR-88D System

The three major functional components of a WSR-88D System are the: Radar Data Acquisition (RDA); Radar Product Generation (RPG); and Principal User Processor (PUP). These are discussed in detail in Federal Meteorological Handbook Number 11 (FMH No. 11), 1991. WSR88D Systems are composed of an S-Band Doppler radar and a data processing System that collects, processes, and displays high resolution and highly-accurate data. From the returned power spectral density, WSR-88D Systems provide estimates of the three basic Doppler meteorological radar quantities: reflectivity, mean radial velocity, and velocity spectrum width (a measure of the variability of radial velocities in the sample volume). The Systems provide mean radial velocity data and spectrum width data out to a range of 230 km. Reflectivity data are provided out to 460 km. The Status and Control Processor in the RDA controls antenna scanning patterns, signal processing, ground clutter suppression, status monitoring, error detection, calibration, and the recording of raw and processed base data (Levels I and 11). The RPG, where most of the data processing is done, executes resident algorithms to convert RDA generated base data into meteorological and hydrological products. The RPG also provides velocity dealiasing, control and status monitoring of the RDA and RPG, Level III data recording of products and algorithm output, and product distribution.

The RPG passes products to the PUPS. PUPs display, annotate, manipulate, and distribute products; control and monitor the PUP system status; and record products (Level IV). The PUPs are where forecasters display and interpret WSR-88D products. WSR-88D antennas continually scan their environment in a sequence of pre-programmed 360 azimuthal sweeps at various elevation angles. A complete sequence of azimuthal sweeps is a "volume scan". The volume scan strategy selected is operator controlled and is determined by operational needs of the NEXRAD agencies. (Data requesters will not be able to influence the volume scan selected.)

3. Overview of WSR-88D Data Recording Capabilities

The WSR-88D can record data at four levels.

a. Level I Data

These data are the analog signals of the radar and are used by engineers to investigate and test various operational characteristics of the signal processor and radar system. There are no plans to record or archive these data.

b. Level II Data

Level II data are the digital base data output from the RDA's signal processor in polar format. These are the same data transmitted over high-speed, wide band communications to the RPG before processing by the meteorological and hydrological analysis algorithms. These data include the three basic Doppler radar moments and system status information/metadata required to properly interpret the data (e.g., RDA status data, maintenance/performance data, RDA to RPG console messages, maintenance log data, RDA control commands, clutter filter bypass map, and antenna scanning pattern). Table 1 contains the spatial and data resolution of these data.
The current Level II recording media are reusable 8 mm tapes that can hold approximately 4.7 gigabytes of data per tape. The data rates vary between 44 Mbytes per hour to 177 Mbytes per hour depending on the scan strategy(ies) used.

c. Level III Data

The Level III data collected by National Weather Service Network sites (ultimately 113) are the base and derived products that are only a small subset of the available data, have been processed through algorithms, and have been quantized into 16 data levels versus the original 256 levels in Level II data. These data are produced by a WSR-88D RPG. Selected system status information, adaptable parameter settings used to create products, and background maps are also part of the Level III data. The current Level m recording media are 5.25 inch double- sided Write Once Read Many (WORM) optical disks (278 megabytes per side, formatted). Data collection rates will be approximately 0.5 Mbytes/hour to 2.1 Mbytes/hour.

d. Level IV Data

Like Level III, Level IV data are the base products and derived products/algorithm output (as in Level III) produced by a WSR-88D RPG, but recorded on a PUP. The PUP operator selects the Level IV products that will be recorded. For a given PUP, these data will be a subset of the data available at the host RPG, since an RPG can generate more products than a given PUP can receive. The current recording media is the same as that used for Level m. Every PUP has a Level IV recorder. There are no plans for centrally archiving Level IV data.

Spatial Doppler Moment	**Resolution**	**Resolution**
Reflectivity	0.95° by 1 km	0.5 dBZe
Radial velocity	0.95° by 0.25 km	0.5 ms
Spectrum width	0.95° by 0.25 km	0.5 ms

Table 1. The spatial and data resolution of WSR-88D Level 11 data.

4. Recording WSR-88D Data

National Weather Service WSR-88D Network sites, upon commissioning, will record Level In data and send them to the National Climatic Data Center (NCDC) for archives The FMH No. 11, Part A, specifies the data to be recorded. Level II data will be collected on a subset of the WSR-88D Systems on a non-interference basis. The Level n recorders are transportable and may be relocated among WSR-88D Systems. The sensitivity of the WSR-88D is great enough that meteorologically useful information is obtained in clear air situations. In order to capture important precursor information and to simplify operations, the recorders will normally record data continuously. Sites will mail recorded media to the NCDC where the data will be quality control checked, inventoried, catalogued, copied, and archived. Requests for entire tapes or selected time subsets of a tape should be sent to the NCDC The NCDC will have a copy of the Level II inventory on line. As of October 1992, Level II data have been collected at seven WSR88D sites.

5. Archiving and Distributing WSR-88D Data

All Level 11 and III data collected on operational WSR-88D Systems will be archived and available for redistribution to requesters. The potential amount of Level 11 data that will be collected (approximately 500 Gbytes per year per site)--and the large number of anticipated requests for data-requires personnel, equipment, and an operation dedicated to the task. The NCDC will fulfill these needs as described below.

a. Level II Data

Beginning in 1993, the NCDC will be the permanent archive for Level II data. To meet the needs of the NEXRAD agencies and researchers for specific event-related Level n data, the NCDC will (by the end of 1993) catalogue the occurrence of specified meteorological phenomena (an event directory). The event directory will enable the OSF, NEXRAD agencies, and other investigators to identify those times and locations where certain meteorological phenomena (e.g., tornadoes, hail, strong winds) occurred. Level II data will be kept at the NCDC in accordance with the Records Disposal Schedules approved by the National Archives and Records Administration. To assist Level II users, the NCDC will publish and maintain a WSR-88D Level II Data User's Guide (available in the first quarter of 1993). This guide will describe the content and format of the data and status information, sample programs to read the data, how to obtain Level II metadata, data reproduction costs, and how data can be requested.

b. Level III Data

The NCDC will be the permanent archive for Level III data. These data will be kept in the NCDC in accordance with the Records Disposal Schedules approved by the National Archives and Records Administration. Requests for color hard copies and acetate overlay copies of Level III data should be sent to the NCDC. A WSR-88D Level III User's Guide will be prepared that describes the data available.

6. Uses and Applications of Recorded WSR-88D Data

Many different user groups will use recorded WSR-88D data. These groups include the OSF, research and development groups developing applications for the WSR-88D System, groups using the data to support other meteorological research (e.g., hydrologic and climatic), and users whose application of WSR-88D data are unforeseen at this time. Level II data can be used on most computer systems--a WSR-88D is not required.

An advantage of using Level III data versus Level II is that no base data processing is required. In addition, these data will be collected continuously at all NWS WSR-88D Network sites. These data will be a valuable resource and will have a wide range of applications.

7. Metadata

Metadata for WSR-88D Systems will have at least three different forms. First, as mentioned in Sections 3b and 3c, some system status information will be written to the recording media. Second, the OSF has configuration management control over the WSR-88D System baseline and will provide NCDC updates on system configuration changes that affect the data in the archives. Third, the NCDC will create station histories, an inventory of the data in the archives, a description of the data/data dictionary, and results of quality control checks.

8. Future Work

In order to fully understand the total needs of the user community, and to determine some of the interdisciplinary uses, a WSR-88D Level II User Workshop will be held in 1993. The workshop will involve the operational and research communities and include government, academia, and private sector potential users of the data. Following this workshop, a draft Strategic Plan for the Archive and Use of WSR-88D Level II Data will be developed concerning all aspects of Level II archiving from on- line services to browse of data sets. Initially, the tools to work with the Level II data sets will not be available at most universities and other locations. The University of Oklahoma and National Severe Storms Laboratory will develop a limited capability to process and display Level II data on at least two different UNIX based workstations. At a minimum this will include the capability to peruse the archive tape and create graphical images of base data. This will ensure that users of the data can work with the Level II data and use it to support their research projects. Future development work will investigate the feasibility of creating an on-line browse capability for researchers, expand the capabilities to meet the increased amount of data collected and requests for archive data, and refine the capabilities already developed to better serve the needs of data requesters.

The Definition Of Station And Management Of Station Metadata Information In Support Of Climatological Data Bases

Anne Viront-Lazar

USDOC,NOAA,NESDIS
Global Climate Laboratory
National Climatic Data Center
Asheville, North Carolina

1. INTRODUCTION

NOAA's National Climatic Data Center (NCDC) is the official national archive for over 200 million original records of weather data. A collection platform for these data is called a "station," and stations can be categorized as follows:

1) surface, fixed point (e.g. airport weather station)
2) surface, mobile (e.g. ship)
3) atmospheric (e.g. aircraft)
4) outer space (e.g. satellite)

In this paper, discussions of station draw primarily from the first category, however the concepts and applications can be applied to all categories. In all cases, weather observations (data) are recorded from a location utilizing certain instruments and observing practices (metadata). The acquisition, organization and access of the collection of data is highly dependent upon the metadata. This paper addresses the concept of station, the NCDC definition of station, and how that definition impacts the management of metadata for some 30,000 stations, with periods of record ranging from single observations to 150 years of observations.

2. THE ONLINE ACCESS AND SERVICE INFORMATION SYSTEM (OASIS)

The NCDC is mandated to provided ever-improving access to the climatological data archive. Our latest development effort, the Online Access and Service Information System (OASIS) provides on-line access, browse and ordering information for several of the Center's major Projects containing principle digital data sets, and has the potential to include inventories of all holdings, both digital and non-digital. OASIS resides on UNIX/Sun Workstations and uses the EMPRESS Relational Data Base Management System (RDBMS).

Oasis was designed to provide online, integrated access to large data sets. Initial datasets contained in OASIS include Wind Profiler, Hourly surface, Daily surface, Upper Air, and Monthly Global Historical Climate Network data sets. The initial precursor system to OASIS was the Management of Atmospheric Data for Evaluation and Research in 1991 (MADER) project. Contained in MADER are an environmental data management package consisting of data formatted by ASCII identifier and packed binary record portions, and a comprehensive metadata module. The metadata module, which is retained in OASIS, consists of numerous EMPRESS relational database tables, with accompanying software designed to store and inter-relate data to support eight functional areas:

1) Station History (location, identification, observational practices and instrumentation)
2) Dataset Inventories (dataset data availability)
3) Event Directory (location, time of specific meteorological events)
4) Dataset Catalogue (features, contents, archival points)
5) Data Dictionary (description of the metadata)
6) Physical Data Directory (storage type, location and format of data sets)
7) On-line Data Access
8) Interactive Order Entry

The MADER system was designed with a three-field station key to link the data with the metadata module. The key field (called the CLI) consists of latitude, longitude and an occurrence number within identical latitude-longitude pairs. The use of this location information as the primary key strongly supports synoptic data retrieval. However, it became quickly apparent that the location key was not efficient in the organization and access of time series data, alternate data types such as satellite scans, other non-point source summaries, and for metadata management.

To improve the environmental data management module, we acquired the Naval Environmental Operational Nowcasting System (NEONS) Environmental Database, developed at the Naval Laboratory (NOARL) in Monteray, California. NEONS offers the capability for management of three basic data types (image data, gridded data, and point data), and selection of datasets based on time, space or data content. The link between NEONS and the MADER metadata module is accomplished through a station identifier number (called the NEONS_ID), making the concept of "station" critical to the management of the entire OASIS system.

3. DEFINITION OF STATION

A weather observing station is a place, either on the surface of the earth, in the atmosphere or above the atmosphere from which atmospheric and other environmental phenomena are observed and recorded. A station can be described in the contexts of place, function and time. Place includes location of the station in traditional reference systems (latitude-longitude coordinates, as well as geopolitical anchors such as place name, county, country, etc.), and encoded identifiers such as station numbers, call letters, etc. Function includes both instrumentation at a station and observing practices. These can range from a person's observation of hail stones, to the twice daily launching of instruments by balloon according to set operational directives, to the most sophisticated automated observation system.

The first issue of complexity comes with the interaction of place and function. If a station is tasked with observing two different types of weather parameters, for example, air temperature and precipitation, is it now one station with two functions or two stations of one function each? Does the distance between the two instruments affect the answer to the question? Perhaps the relation between the two parameters observed is vital to a user. Multiply these concerns by the dozen or so instruments scattered around a typical primary weather observing site.

The second issue of complexity comes with the interaction of place and function with time. To what degree do station or instrument relocations affect the identity of a station? Or how do changes in instrument types or observational practices affect a station? Users of synoptic data (widespread data snapshots) and users of time series data (long term data trends) have differing definitions of stations.

Within the NCDC there have been many definitions of station. For example, the National Weather Service (NWS) has declared that Cooperative Program stations (taking daily summary observations) can relocate up to 5 miles and still be considered the same station if a local manager considers the data compatible with previous sites. On the other hand, the US Historical Climatological Network (USHCN) Project, which is concerned with the construction of highly accurate, long term data sets for the study of climate change, will sometimes combine 2 or more NWS Cooperative stations into one long term USHCN station, creating homogeneity between sites with adjustment routines. Both the Cooperative station data and the USHCN data are derived from the same source, but are organized into two different station identification schemes.

The OASIS station history metadata is designed to not only allow access to these two and other data sets, but to integrate the access to these data sets through a centralized station history data base. The station history must describe the one site that observed the data that ended up in one or more data sets. In order to accommodate these two and many other differing examples of station definition, an arbitrary definition of station must be developed by the data base manager.

4. STATION DESIGNATION IN OASIS

Two main factors influenced our decisions on a station definition for OASIS: 1) the traditional NWS definition of station, and 2) the complexity of the data base designs in the OASIS metadata module.

The traditional NWS station definition is based on administrative units of operation, and includes the Cooperative station example cited above, as well as the practice of calling an airport a station, even though instruments can be located thousands of yards apart. Some arguments have been made in favor of defining each instrument as a station, but the demands of converting source station information into new organizational structures were judged to be too great. Also, the majority, if not all of the NCDC archive is organized in the NWS station definition.

The design of the OASIS station history data bases includes over 80 different tables of information, 18 of which contain the principle station identifier as the key link field. To best optimize the maintenance of and navigation through this structure, we felt it was important to minimize the total number of stations defined, using the data base tables to record those features of stations that may be important to another user's definition of station.

Two station designation rules evolved to best meet the above concerns:

1) all airports shall be considered unique stations throughout time, no matter how or where instruments are relocated, added or deleted; and

2) all NWS Cooperative stations, as organized by the Cooperative station number, shall be considered unique stations throughout time, no matter how or where instruments are relocated, added or deleted. When a Cooperative station, for part of its existence is located at an airport, the airport station designation takes precedence.

Stations not falling into these categories generally will be considered unique stations with each significant (>100 feet) relocation.

5. STATION LINKS IN OASIS

The predecessor to OASIS, MADER, contains the 3-variable station identifier (CLI) that contains latitude, longitude and occurrence number. The CLI is present in all data records, in 18 of the station history tables and in the detailed dataset inventories. Although the concept of instant linkage between data, inventories and station information is a good one, we found it difficult to implement. Take for example a station that operated from two locations historically. This would be stored as two "stations" in the CLI base system. If an error were found in the **date** of station relocation, data records, inventory records and at least 36 station information tables would all have to be changed.

An additional obstacle was discovered when NEONS was acquired for data management. The EMPRESS RDBMS has strong internal software support for linking multiple data bases using one argument (variable). The 3-variable CLI identifier would dictate the development of a volume of supplementary software to connect the two systems.

As a result of these impediments, we created a new key, the NEONS_ID, to provide better linkage in OASIS. The NEONS_ID was added to the 18 station information tables, and all NEONS data records. MADER data records and inventory records are presently still set up with the CLI. Until or if this is changed, we can use the NEONS_ID to better manage the CLI-based data and inventory information by leaving the CLI value frozen, and maintaining latitude and longitude changes in a separate table.

6. MANAGEMENT OF MULTIPLE STATION DEFINITIONS

The preceding sections discussed the need for an arbitrary definition of station as basis for the structure of a station metadata data base. This section discusses the management of multiple station groups as defined by data producers and users.

Climatological data are collected and organized by station networks, i.e. groups of stations observing the same types of data (e.g. National Weather Service surface airways observations; global synoptic transmission, USHCN data set, etc.). Stations are identified in these networks by a number or alpha-code for digital data sets, and by number, alpha-code or station name for non-digital collections. Each network has its own station definition.

The OASIS model is then tasked with the documentation of network-defined stations within its own data base-defined station structure. This can be accomplished with three data base tables: station identifier, station name/alias, and station dataset.

The station identifier table is the principle gateway for retrieving metadata about network-defined stations. This data base table contains the data base station key variable (NEONS_ID), the network-defined station identifier, a coded field describing the type of network identifier, and begin/end dates. Information for a network-defined station can reside in one of more data-base-defined stations for specific time spans. For a given network-defined station identifier, the logical search against the data base will return one or more groups of three variables:

NEONS_ID(1)	begin_date(1)	end_date(1)
NEONS_ID(2)	begin_date(2)	end_date(2)
....	....	....
NEONS_ID(n)	begin_date(n)	end_date(n)

All subsequent date base queries for information about the network-defined station (e.g. location, instrumentation) must be based on these groups of three arguments.

The station name/alias table, similar in structure to the station identifier table, can also be used to access information about network defined stations, but due to the imprecise nature of place-names (i.e. spelling errors, non-standard abbreviations, multiple station names), searches utilizing this table are subject to interpretation. The utility of this table is directly dependant upon the effort made to organize and populate the tables with all known variations of station names and aliases.

The station identifier and name tables are utilized to retrieve information about known network-defined stations. Other data base users need to query data base-defined stations for their membership in observing networks. This is accomplished through the dataset table, which relates a station to its data holdings. Once it is determined that a data base station contributes to a target dataset, the proper identifier (number, alpha-code or name) can be retrieved from the station identifier/name tables to proceed with data selection/extraction.

7. CONCLUSIONS - FUTURE TRENDS

In the future, successful metadata systems will be accommodating larger and more diverse collections of information sets. Environmental data, linked to points or areas on or above the earth's surface, must be managed in a systematic way if the comparison of diverse holdings is to have any meaning. The analysis of climatological data is dependent upon accurate documentation of weather observing stations. The management of this station information with a data base-dependant station definition can optimize the collection and verification of station metadata from a variety of observation systems.

Metadata and the GENIE Project[1]

I. A. Newman[2]
Department of Computer Studies
Loughborough University,
Loughborough, Leics. LE11 3TU, UK

Abstract:

This paper summarises some of the key features of the GENIE project that may be relevant to environmental scientists designing data management systems. It commences by giving the objectives and assumptions which formed the starting point for the project. The main features of the design are then outlined indicating how these meet the requirements within the assumptions.

1. Introduction (Objectives and Assumptions)

The GENIE project was established in April 1992 to fulfil a perceived need to make existing data more readily available to Global Environmental Change (GEC) researchers. The consortium who were awarded the contract to carry out the project involves academics (computer scientists, human scientists, geographers), computer centre personnel and a GIS supplier (GENASYS II). The main objectives of the project (as taken from the contract) are:

- to create a 'Master Directory' service for enquirers in the natural and social sciences wishing to access UK-held datasets pertaining to GEC;
- to develop a Federal Network which links existing UK Data Centres holding GEC data with the researchers who wish to access that data;
- to provide, develop and promote the infrastructure for coordinating and improving access to, and visibility of, UK GEC data;
- to increase the flows and exchanges of information on GEC datasets within the UK research community;
- to examine data quality standards;
- to monitor: usage; storage and exchange of metadata and, later, data;
- to provide the UK focal point for information on international science and policy developments in GEC data management;
- to acquire, from individual Data Centres throughout the UK, information about the existence and availability of GEC data;
- to facilitate and encourage the entry of metadata by individual Data Centres;
- to develop information retrieval systems to make the metadata available for on-line searching;
- to develop network links to make the metadata accessible nationally and internationally, in as far as is practicable to assist users to gain access to metadata stored in other countries;

[1] The support of the Economic and Social Research Council (UK) is gratefully acknowledged. The GENIE project is tasked with providing the ESRC UK Global Environmental Change Data Network Facility.

[2] Other principal consortium members: David Hogg (Director, Computer Centre, LUT), John Keene (GENASYS II), Paul Mather (Geography, Nottingham University), David Medyckyj-Scott (CS, LUT), Clive Ruggles (Computer Studies, Leicester University), Rob Starling (GENASYS II), Simon Thompson (GENASYS II), David Walker (Geography, LUT).

- to liaise with Data Centres and Archives to promote the provision of good quality documentation and to make it available as part of the metadata;
- to develop systems which encourage the entry of metadata so as to reduce the time taken to make changes and additions to the data visible to enquirers.

The main assumptions underlying the design of the system provided for the project are:

1 very large volumes of data will need to be made accessible;
2 very large numbers (millions) of metadata records will exist;
3 the data are distributed at many sites in the UK (and worldwide);
4 the data comprises a mixture of online and offline sources (flat files, databases, experimental notebooks, rock samples, map libraries, bird song recordings etc.);
5 large numbers of occasional users with little interest in computing;
6 Data Centres already have their own methods of managing data and making it available to scientists in 'their' discipline;
7 most existing metadata is inadequate for use by researchers who are not trained in the appropriate discipline (e.g. there are many implied assumptions, the indexing where it exists uses 'specialist' words and phrases);
8 many datasets do not even have enough metadata to allow a specialist to use them immediately (without some prior 'data mining') and there is normally no money for a major metadata generation exercise;
9 potential users (researchers and Data Centres) will want to use existing equipment to access the service (but all users who wish to use the service must have access to e-mail);
10 equipment (networks, computer hardware, software) will fail, users will disconnect their machines from the network and possibly reconnect at a different physical location;
11 metadata records will, predominantly be text oriented and fairly small (though pictures and audio may need to be supported);
12 there is no requirement for rapid access to data, or even to metadata.

2. Design Considerations - Data & Metadata.

The data are managed by the Data Centres. The GENIE project merely provides a data transport service for a complete dataset or some agreed subset, or product, made available by the DBMS at the Data Centre. Updates would not be generated directly by the user and the GENIE project has no responsibility for managing data (though it might provide a limited service in the future).

At the metadata level, although most of the metadata would initially need to be provided by the Data Centres to describe their holdings, users must be permitted to add metadata (e.g. comments on quality or experimental design; notes on the usefulness of data) at any time if they so wish. Furthermore, every query made by a researcher can be considered to be a piece of 'metadata' in its own right. Examining queries indicates both the data that are being requested most often (which assists with data management) and the data that are wanted but not currently provided (which may encourage experimenters to consider providing them). However, a researcher may wish to keep his, or her, comments and queries private or confine their publication to a small group (some Data Centres may also wish to keep some of their metadata private).

If every user can provide metadata yet an effective service is to be provided even if hardware and networks fail, there is a strong argument for providing each user with their own independent system (software and metadata). This is the design approach that is taken in the GENIE project (i.e. each user has an IMP or Information Management Processor). However, taking this approach means that the problems of communication and co-ordination amongst users become much more important than with a centralised approach. Occasional users and new users will require guidance as to what to do based on 'existing practice'. Organisations may want to encourage some uniformity amongst their personnel. Also, the benefits of providing information about previous queries can only be obtained if the information can somehow be collected from the users making the queries.

At the level of the above discussion, the metadata is just 'data' that needs to be managed in a distributed environment, so the GENIE design is for a totally distributed data management system (DDMS) which happens to manage 'metadata'. The main difference from more conventional DDMSs occurs because of assumption 12. If there is no need for speed, all communication can be asynchronous using e-mail in the first instance. E-mail was chosen because it is widely available and provides a store & forward capability which enables all communication within the system to be 'direct' from one IMP to another, named, IMP.

3. The Design of an IMP

An IMP maintains the metadata known to its user and provides the interface to the remainder of the system. The owner of an IMP must decide what information to store in his, or her, IMP, either by entering it directly or by agreeing to store it after obtaining it from some other IMP. When storing items of metadata the user can identify any number of 'phrases', of one or more words, which characterise the information in some way or can link it to any other existing piece of metadata. The metadata is indexed using the phrases and the links (also, automatically, by the time and date of entry) to facilitate rapid retrieval at a later stage. The user can add extra indexing information to any piece of metadata at any time.

To reduce the workload on the user, the IMP provides:

- a means of linking one or more phrases to a 'concept' (a 'meaning'), so that all subsequent uses of any of the phrases are deemed to be linked to the same concept;
- the possibility of linking the same phrase to two different concepts (the user will be asked to decide which concept applies when the phrase is used);
- a way of sending queries to other IMPs, without requiring the user to explicitly identify IMPs to be interrogated;
- a method of recording messages from, and to, other IMPs and of linking between the concepts identified by the owner of another IMP and the concepts identified by this user (the concepts used in the other IMP are sent with the message).

3.1 Technical Information

In the initial version of the system, all metadata are stored in 'documents' which are made up of several sections each consisting of an ordered collection of paragraphs comprising bytes. At each level there are associated 'interpretations'. An interpretation contains the concepts that are relevant to its associated object (document, section, paragraph). Every concept must be identified by the user either explicitly or implicitly. Mostly, in practice, selection is implicit, either from the interaction (an icon, or menu option, implies one or more concepts) or based on a previous explicit identification. Examples of concepts would be:

- a subject covered by the document/section/paragraph e.g. tropical forest, water pollution;
- a spatial area e.g. North Wales, the area from 10'W to 6'E & 30°36'N to 31°N;
- a time period e.g. 10/1/91 to 18/2/91, Middle Ages;
- formatting information e.g. indented paragraph, main heading, italic;
- type (of metadata) e.g. query, metadata, comment;

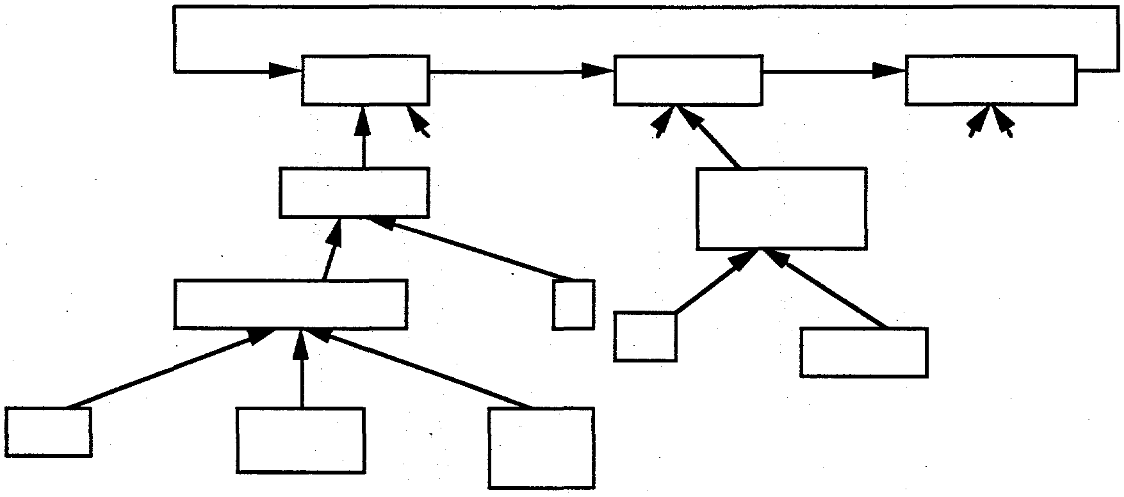
Each concept is assigned a unique number within the system and has associated with it a document containing the 'explanations' of the concept (a piece of metadata which would comprise any, or all, of: text; pictures; audio; phrases for identifying the concept; and a list of all the other concepts to which this concept is linked).

4. Inter-IMP Communication.

Initially, when a new IMP is created it is assigned a 'parent' IMP, and it can only communicate with that parent . If the owner of an IMP makes a query which cannot be answered from the metadata contained within it, or when new metadata is entered for public consumption, a communication is sent to the parent. Either explicitly or implicitly, the person in charge of the parent IMP will have to decide how to process this communication. A query could be passed on to another IMP which is likely to have suitable metadata, published metadata may be passed to the parent's parent or to another IMP which has explicitly requested information of this sort. If a message (query or metadata) is passed on, information about the originator (including their e-mail address) is also passed on (as a piece of metadata in its own right). This enables the recipient to reply directly to the originator, who then gains knowledge of the existence of another IMP in addition to the parent.

The initial logical structure is a combination of a set of hierarchies and a ring as illustrated below. IMPs which are on the ring are 'trusted'. They are expected to be online and available the majority of the time and are all known to one another. They provide two services: a repository for the metadata that has been published by the IMPs in their hierarchy (so that it can be made available to enquirers without requiring the originating IMP to be online); a copy of an abstraction of the metadata that is known about by all the other trusted IMPs. Whenever a trusted IMP receives new information which the owner/manager of the IMP feels should be publicised it is sent round the ring. Eventually all the trusted IMPs should have the same set of abstracted metadata.

As IMPs communicate, this 'formal' structure becomes overlaid with a network of accredited IMP-IMP links. An IMP may request a 'notification of update' to a piece of metadata or about a particular subject from another IMP. When an update occurs the second IMP sends a message indicating that this has happened and cancels the request. If & when the originating IMP requests the update it can also request notification of a subsequent update. These mechanisms are designed to minimise network traffic where updates may occur frequently relative to the likelihood that the metadata is actually requested (but work acceptably if metadata is usually requested after a change).



5 Other Issues

5.1 Lazy Entry of Metadata

The GENIE system is designed to make use of any information that is available in machine readable form (e.g. project proposals, reports) by reading it into an IMP, storing it as metadata and indexing it appropriately. If sufficient indexing information can be made available, queries addressing the subject can be routed to that IMP. Provided that the IMP owner replies to the query using the system the reply will be recorded as additional metadata. Metadata is, therefore, added about the records and/or topics that are of most interest to researchers (a service can be provided without, first, needing a major metadata collection exercise).

5.2 User Interface Uniformity

A user who has an interest in a particular topic can provide user interface templates for either, or both, querying and metadata entry. Since, by definition, the 'topic' will be a concept, the interfaces can be part of the document that is associated with the concept. This permits a person performing a query to be shown both the main sub-topics that 'belong to' the topic (in the form of labeled icons, menu entries or data capture boxes) and also the amount of metadata that is currently associated with each of these sub-topics. Similarly, someone offering to supply metadata about the subject can be shown the main headings under which metadata has been supplied in the past and the frequency with which queries have been made on each of the headings. In each case, on most occasions, the minimal effort for the user will be to use the pre-existing format. However, the user is given tools which enable a new option, a new menu, or a new data capture field to be added to the interface. Furthermore, since the interface is itself a concept and so are the fields within it, the user who created the interface template can request to be informed of any changes to any of the concepts associated with the interface. This user can then decide whether to integrate any change into the 'standard' interface and can notify interested users when such a change is made.

Appendix A: List of Workshop Participants

Name	Address	Electronic Mail
Gerry Barton	NOAA/ESOM EX2 Universal 506 1825 Connecticut Ave NW Washington, DC 20235 (202) 606-4548	OMNET G. Barton barton@esdim.noaa.gov
Tom Boden	CDIAC Oak Ridge National Laboratory P.O. Box 2008 Oak Ridge, TN 37831-6335 (615) 574-0390 (615) 574-2232 (Fax)	Omnet: CDIAC tab@ornl.gov
Wayne Brazille	NOAA, N/C/SPO 325 Broadway Boulder, CO 80303 (303) 497-8169	brazille@ncar.ucar.edu
Jim Brown	PNL P.O. Box 999, MS/K7-22 Richland, WA 99352 (509) 375-3626 (509) 375-3641 FAX	jc_brown@pnl.gov
Tim Crum	WSR-88D Operational Support Facility 1200 Westheimer Drive Norman, OK 73069 (405) 366-6500 ext 252 (405) 366-6550 FAX	OMNET: T.Crum
Judy Cushing	Computer Science Department Oregon Graduate Inst. Beaverton, OR 97006-1999	Cushing@cse.ogi.edu
Lois Delcambre	Dept. of Comp. Sci. and Eng. Oregon Graduate Institute 19600 N.W. von Neumann Dr. Beaverton, OR 97006-1999 503-690-1689	lmd@cse,ogi.edu
Mike DeVaney	Pacific Northwest Laboratory P.O. Box 999, K1-87 Richland, WA 99352 (509) 375-2435	dm_devaney@pnl.gov
Elaine Dobinson	JPL M/S 525/3610 4800 Oak Grove Drive Pasadena, CA 91109	elaine_dobinson@isd.jpl.nasa.gov
Judy Feldman	Hughes Aircraft 7375 Executive Place, Suite 401 Seabrook, MD 20706 (301) 805-0332	omnet: j.Feldman judy @ eos.hac.com

Jim French	Dept. of Computer Science Thornton Hall, University of Virginia Charlottesville, VA 22903 (804) 982-2213	French@Virginia.edu
Jim Frew	Univ. of California Computer Sci. Div, 571 Evans Hall Berkeley, CA 94720 (510) 642-0267	frew@postgres.berkeley.edu
Goetz Graefe	Portland State Univ. Computer Sciences Dept. P.O. Box 751 Portland, OR 97207-0751 (503) 725-4036	graefe@cs.pdx.edu
Yannis Ioannidis	Computer Sciences Dept. 1210 W. Dayton St. University of Wisconsin Madison, WI 53706 (608) 263-7764 (608) 262-9777 (Fax)	yannis@cs.wisc.edu
Marcos Lester	Battelle Environmental Planning and Social Research Center 4000 NE 41st Street Seattle, WA 98105 (206) 528-3308	mk_lester@pnl.gov
Dave Maier	Dept. of Comp. Sci. and Eng. Oregon Graduate Institute 19600 N.W. von Neumann Dr. Beaverton, OR 97006-1999 (503) 690-1154	maier@cse.ogi.edu
Marjorie McGuirk	NOAA 325 Broadway Boulder, CO 80303 (303) 497-3090	OMNET M.McGUIRK mcguirk@fsl.noaa.gov
Dan McKenzie	U.S. EPA 200 SW 35TH Street Corvallis, OR 97333 (503) 754-4625 (503) 754-3615 FAX	McKenzie.Dan@EPAMAIL.EPA.GOV
Blanche W. Meeson	NASA/Goddard Space Flight Center Code 9022 Greenbelt, MD 20771 (301) 286-9282	meeson@pldsg3.gsfc.nasa.gov
Ron Melton	Pacific Northwest Laboratory P.O. Box 999, K7-02 Richland, WA 99352 (509) 375-2932 (509) 372-4761 (Fax)	rb_melton@pnl.gov
Bill Moninger	NOAA, R/E/FS1 325 Broadway Boulder, CO 80303 (303) 497-6435	Moninger@FSL.NOAA.GOV
Ian Newman	The Genie Project, Dept of Computer Studies Loughborough University Loughborough, Lfics LF 113TU, UK 44 509 222687 44 509 211586 - FAX	I.A.Newman@lut.ac.uk

Calton Pu	Dept. Comp. Sci. & Eng. Oregon Graduate Institute 19600 NW von Neumann Dr. Beaverton, OR 97006-1999 (503) 690-1214	Calton@CSE.OGI.EDU
Ruth Ross	Atmospheric and Ocean Sciences 1225 W. Dayton St. Madison, WI 53706 (608) 262-3086 (office) (608) 262-2827 (sec'y) (608) 262-0166 (Fax)	ruth@meteor.wisc.edu
Edward Sciore	Computer Science Dept. Boston College Chestnut Hill, MA 02167	SCIORE@BCUXS2.BC.EDU
Bob Shepanek	EPA HQ/RD-680 401 M Street SW Washington, DC (202) 260-3255	shepanek.robert@epamail.epa.go v
Arie Shoshani	Lawrence Berkeley Laboratory mail stop 50B-3238 Berkeley, CA 94720 (510) 486-5171	shoshani@lbl.gov
Paul Singley	Oak Ridge National Lab P.O. Box 2008, MS 6490 Bldg. 0907 Oak Ridge, TN 37831-6490 (615) 574-7817 (615) 574-4665 FAX	sin@ornl.gov
Tim Smith	Hughes STX EROS DAta Center Sioux Falls, SD 57106 (605) 594-6091	tsmith@glis.cr.usgs.gov
Don Strebel	VERSAR, INC. 9200 Rumsey Rd. Columbia, MD 21045 (410) 964-9200	strebel@ltp.gsfc.nasa.gov
Jim Thomas	Battelle, PNL P.O. Box 999/ms k7-34 Richland, WA 99352 (509) 375-2210	JJ_Thomas@PNL.gov
Joyce Tichler	Brookhaven National Lab Building 051 19 W. Brookhaven Ave. Upton, NY 11973 (516) 282-3801 (516) 282-3911 FAX	OMNET: J.Tichler tichler@bnl.gov
Anna Viront-Lazar	Data Base Mgmt Branch-Stop 33 Nat'l Climate Data Center 37 Battery Park Avenue Asheville, NC 28801 (704) 259-0380	alazar@ncdc.noaa.gov
Michael Woodford	NOAA/NODC E/OC24 1825 Connecticut Ave NW Rm. 415 Washington, DC 20235	Woodford@NODC2.NODC.NOAA.gov Omnet: NODC.Pollution.INFO

Appendix B: Glossary of Terms and Acronyms

Annotative Metadata

Metadata that provide context for the primary data are referred to as annotative metadata. Examples include information in scientific notebooks, instrument logs, manuals, and reports that document the platform and instrument conditions, the operational environment, interfering sources of noise, and that uniquely identify the software and computer platforms used for analysis, modelling, and simulation.

Denotative Metadata

Within the database community the term metadata has often been used to refer to the information that describes the structure of a database. We refer to this as denotative metadata.

HDF

Heirarchical Data Format is a multi-object file format that facilitates the transfer of various types of data between machines and operating systems. Machines currently supported include the Cray, Convex, HP, Vax, Sun, IBM RS/6000, Silicon Graphics, Macintosh, and IBM PC computers. HDF allows self-definitions of data content and easy extensibility for future enhancements or compatibility with other standard formats; includes Fortran and C calling interfaces; utilities to prepare raw image of data files or for use with other NCSA software. The HDF library contians interfaces for storing and retrieving compressed or uncompressed raster images with palettes; an interface for storing and retrieving n-Dimensional scientific datasets together with information about the data, such as labels, units, formats, and scales for all dimensions.

Metadata

We use the term metadata in this report to refer to annotative metadata.

NetCDF

Network Common Data is an interface for scientific data access and a library that provides an implementation of the interface. The netCDF library also defines a machine-independent format for representing scientific data. Together, the interface, library, and format support the creation, access, and sharing of scientific data. The netCDF software was developed at the Unidata Program Center in Boulder, Colorado. The freely availablesource can be obtained by anonymous FTP from: ftp.unidata.ucar.eduin the pub/ netcdf/directory.

Pedigree

A term often used in lieu of provenance.

Primary Data

We have chosen to qualify the term data with primary to refer to the basic information produced by an instrument or calculation.

Provenance

Provenance means "place of origin" or derivation. It is used within some communities in the context of establishing the authenticity of an item such as a work of art. Within the scientific community provenance is sometimes used to describe the history of data. In the context of metadata the question is, "what metadata is required to establish the provenance of a data set?"

SDM

scientific data management