MARCH 1984 VOL. 7 NO. 1

a quarterly bulletin of the IEEE computer society technical committee on

Database Engineering

Contents

Database Engineering Bulletin is a quarterly publication of the IEEE Computer Society Technical Committee on Database Engineering. Its scope of interest includes: data structures and models, access strategies, access control techniques, database architecture, database machines, intelligent front ends, mass storage for very large databases, distributed database systems and techniques, database software design and implementation, database utilities, database security and related areas.

Contribution to the Bulletin is hereby solicited. News items, letters, technical papers, book reviews, meeting previews, summaries, case studies, etc., should be sent to the Editor. All letters to the Editor will be considered for publication unless accompanied by a request to the contrary. Technical papers are unrefereed.

Opinions expressed in contributions are those of the individual author rather than the official position of the TC on Database Engineering, the IEEE Computer Society, or organizations with which the author may be affiliated.

Membership in the Database Engineering Technical Committee is open to individuals who demonstrate willingness to actively participate in the various activities of the TC. A member of the IEEE Computer Society may join the TC as a full member. A non-member of the Computer Society may join as a participating member, with approval from at least one officer of the TC. Both a full member and a participating member of the TC is entitled to receive the quarterly bulletin of the TC free of charge, until further notice.

Letter from the Editor

This issue is about statistical databases. It contains tutorial articles that present concensus opinions on the current knowledge, problems, and anticipated research directions regarding statistical databases. This issue differs from previous issues of Database Engineering as its articles are not status reports of the current research of specific groups or individuals. Instead, it contains results of working group discussions which were held at the Second International Workshop on Statistical Databases (September 27-29, 1983). Among the workshop participants were experienced practitioners, leading researchers, and recognized pioneers in the statistical database field. The authors of the papers in this issue were usually the working group leaders; the opinions expressed in each article reflect the concensus of the working group and are not necessarily just those of the authors. To acknowledge the contributions of the group members, their names are listed at the start of each article.

The issue begins with a short note from John McCarthy and Roy Hammond, the general chairman and program chairman of the workshop, respectively. They describe the workshop context and give some insights about common themes that emerged from the workshop as a whole. Next is an overview paper that appeared in the workshop Proceedings. In it, Dorothy Denning, Wesley Nicholson, Gordon Sande, and Arie Shoshani present a concise introduction to the problems and research topics of statistical database management. As noted there and in other papers, statistical databases present problems and requirements that current data management and statistical software do not fully address. The subsequent articles represent summaries from individual working groups on the following topics: user interface issues, workstations and special purpose hardware, connecting heterogeneous systems, time series and econometric database management, special data types and operations, logical data models, metadata management, and physical storage and implementation issues. A keyword index is provided at the end of this issue to facilitate the cross-referencing of major topics.

Readers of this issue will be struck by the enormity of the problems that confront statistical database practitioners and researchers alike. Statistical database research, as a whole, is still in its infancy. Almost all of the major problems can be traced to an inadequate understanding of the fundamental needs and basic tools for statistical database management. It is hoped that this issue will contribute to the improvement of this understanding, and will stimulate further research and better solutions to the problems at hand.

Finally, I thank John McCarthy for his help, enthusiasm, and support. I also thank the other contributors of this issue for all the hard work they put in to make this publication possible.

In the upcoming issues of Database Engineering, Randy Katz is editing the June issue on engineering design databases, Dan Ries is handling the September issue on multi-media databases, and Dave Reiner is organizing the December issue on database design.

D.S. Batory

December 1983
Austin, Texas

The Second International Workshop on Statistical Database Management: Common Themes and Issues

John L. McCarthy, General Chairman
Lawrence Berkeley Laboratory
Building 50B, room 3238
Berkeley CA 94720

Roy Hammond, Program Chairman
Statistics Canada, EPSD
2405 Main Bldg, Tunney's Pasture
Ottawa, Canada K1A0T6

1. Introduction

The Second International Workshop on Statistical Database Management was held in Los Altos, California, on September 27-29, 1983. One hundred computer scientists and statisticians from North America, Europe, and Japan attended. The workshop was sponsored by the Lawrence Berkeley Laboratory and the United States Department of Energy, in cooperation with the Association for Computing Machinery's Special Interest Group on Management of Data, the American Statistical Association's Statistical Computing Section, the IEEE Computer Society's Technical Committee on Database Engineering, and Statistics Canada.

2. Workshop Goals and Working Groups

Like the First LBL Workshop on Statistical Database Management, which was held in December, 1981, the Second Workshop brought together researchers and system designers from both computer science and statistics to discuss current work on problems of statistical and scientific database management. It was intended not only to facilitate interchange of ideas, but also to stimulate some systematic and collective thought about research directions. Although the purpose of the Second Workshop was the same as its predecessor, the format and content differed in a number of ways.

Participants came prepared to work in small groups and to produce the reports summarized in this publication. About half of the time was spent in parallel working group sessions, with each group composed of five to ten experienced practitioners from a variety of backgrounds. Members of each working group were asked individually and collectively to discuss and produce written summaries of questions that need to be addressed and promising research ideas in selected subtopics of statistical database management. Each working group then presented its conclusions at a plenary session to get comments from other participants prior to preparation of the summaries presented here.

We hope that these reports will help focus attention of the larger database community on some of the special problems of statistical database management. We are grateful to Don Batory, Won Kim, and IEEE's *Database Engineering* for providing a forum in which to report the results of our working groups.

3. Common Themes and Issues

At the First Workshop in 1981, definition of "statistical database management" and a working vocabulary that computer scientists and statisticians could both use were major issues. At the Second Workshop there seemed to be more agreement among participants on "statistical database management" and a common vocabulary. In addition, several new themes emerged from written contributions to the *Proceedings*, presentations at the workshop, and working group reports summarized in this issue. Four major ideas which recurred frequently were the importance of metadata, the need for richer semantics, the limitations of current relational systems, and the growing role of microprocessors.

First, there was a widespread recognition of the key role that meta-data, or data about data, can play in different aspects of statistical database management. Meta-data is necessary to specify information about statistical data for both human beings and computer programs. It can provide definition of logical models as well as more mundane documentation details for both database administrators and users. Well-defined and differentiated meta-data is necessary to permit software linkages between different logical and physical representations; between statistical databases, application programs, and user interfaces; as well as between multiple distributed and heterogeneous systems.

A second general theme was the need for richer semantics and operators for statistical data. A number of papers and group reports discussed the need to define and manipulate complex, multi-dimensional data structures. For example, many scientific databases require capabilities for defining and operating directly on vectors, time-series, and multi-dimensional matrices. There also seemed to be widespread agreement on the desirability of using self-describing files for both input and output, with functions automatically using and producing meta-data as well as data.

Although the relational model has become the standard for academic research, a number of groups and individuals noted serious limitations of current relational systems for statistical database applications. Of particular concern are the limited number of data types and operators for both data and meta-data. Some felt such limitations might be overcome by extending the relational model to include complex or abstract data types.

Finally, there was a growing recognition of the wide range of opportunities and challenges for statistical database management inherent in the microprocessor revolution. There is an accelerating trend towards transferring statistical data and meta-data from central databases to microprocessor workstations *and vice-versa,* with many attendant problems of distributed data management. High resolution terminals, large local memory and disk storage, fast local processing, and higher data transmission rates are bringing quantum changes in user interfaces and the way in which statistical analysts work. There promises to be an increasing emphasis on interactive graphical display of *pictures* as well as numbers and words for data, data models, meta-data, control options, and so on.

4. Proceedings for First and Second Workshops

Copies of papers, research reports, and issues outlines are available in *Proceedings* for the First and Second Workshops. For either, contact the Computer Science and Mathematics Department, Lawrence Berkeley Laboratory, Berkeley, CA 94720, or the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.

5. Future Workshops

Preliminary planning has begun for a Third International Workshop on Scientific and Statistical Database Management in Seattle during the fall of 1985, Wes Nicholson and David Hall of Pacific Northwest Laboratories will organize the program and local arrangements. One topic that will probably get more emphasis at the Third Workshop is *scientific* database management, particularly for physical science data. Direct inquires to D. Hall, Math 1137/ 3000 Area, PNL, Box 999, Richland WA 99352; telephone (509) 375-2369.

Research Topics in Statistical Database Management

Dorothy Denning
SRI International

Wesley Nicholson
Battelle-Pacific Northwest Labs

Gordon Sande
Statistics Canada

Arie Shoshani
Lawrence Berkeley Labs

Abstract

This report identifies research topics in statistical database management. These topics are grouped into four major areas: characteristics of statistical databases, functionality/usage, metadata, and logical models.

1. Statistical Databases Characteristics

Computer scientists, especially designers of database systems, commonly ask statisticians and data analysts to identify the characteristics or features of a database that identify it as a statistical database. Searching for a profound answer to this question has perplexed data analysts. Many conclude that there are no characteristics which uniquely identify a statistical database. In principle, any collection of quantitative information residing in a computer is a candidate statistical database. As soon as the body of information is interrogated and statistically analyzed, either in total or by sampling or subsetting, it becomes a statistical database.

There are, however, important characteristics that should be built into a database if it is going to be useful for statistical analysis. These characteristics involve adequate description of the quantitative information in the database (i.e., the inclusion of appropriate metadata as defined in Section 3 below.). Such description is essential to understanding inferences evolving from data analysis. Certain kinds of description or definition are almost always included in the database because it is well known that the particular description is critical to understanding the data. On the other hand, certain other information is almost never included even though a detailed analysis will uncover subtleties that are correlated with such description and often cannot be modeled without it. A simple example will serve to illustrate the point. In a database of hospital records, the subject is always described as male or female. This description is important for prognosis and treatment. Periodic readings of blood pressure are also included in the database. On the other hand, the conditions under which the blood pressure was taken – patient lying down, standing up, sitting; recording made on the left or right arm – are almost never included. If the protocol dictates taking the blood pressure on the left arm with the patient lying down, then that information should be included in the database. If there is a variety of conditions, then each blood-pressure reading should be accompanied with a descriptor. When does such detailed information become important? When blood pressure is correlated with treatment protocol, we wish to minimize the random error in the measurements. Clearly if systematic changes in readings can be associated with the position of the patient or the arm on which the reading was made, then that random variability is reduced and a more precise statement can be made about the effect of a specified treatment.

There are distinct types of quantitative data that may be recorded in the database. For each type, there are general conditions which should be met if the information is to be described adequately for detailed statistical analysis.

1.1. Missing Data

Almost every statistical database has incomplete records. Proper statistical treatment of missing data usually depends on the reason for the missing data. For example, in a seismology file listing individual station seismometer magnitudes associated with particular earthquakes, values missing because a station was not operational should be ignored in an estimate of earthquake magnitude. On the other hand, values missing because the signal was either below the seismometer threshold or beyond the seismometer range and off scale, bound the magnitude of the earthquake and should be utilized in an estimate of earthquake magnitude.

As in the seismometer example, there are several possible reasons for a missing value. A set of tags to identify the particular type of missing value should be included in the file. In the seismology example, the tags would at least include "non-operational," "below threshold," and "offscale."

In some situations, such as with questionnaires, the logical structure may influence the interpretation of a missing value; e.g., whereas for males it is not important

whether a question on the number of pregnancies is answered, for females, it is critical to distinguish between a nonresponse and zero.

Most database management systems identify missing values but lack proper tagging capability. Research is needed to improve missing value treatment, and, in particular, to include sufficient information in retrievals so that missing values (either included or excluded) can be properly handled during data analysis.

1.2. Data Quality

Knowing the quality of data is important for statistical analysis. For example, if data are keyed into a file from a remote terminal, how frequently are typographical errors made? Are the data cross checked before being accepted? If data come from a measurement instrument, what is the resolution of that instrument? What is the reproducibility of independent measurements on that instrument? Has that instrument undergone modification during the time that the total set of data was collected? Or further, is that instrument recalibrated every day prior to data collection? These are all important questions; their answers may well influence the way the data are handled in any statistical evaluation. The file should include such data quality information. If the quality is uniform over the entire file, this information can be included in the file descriptor; if it varies in a haphazard fashion, it may be necessary to attach it to each datum.

Further considerations with respect to data quality involve the frequency of spurious measurements through either a breakdown in the data-generating system or the introduction of a rare physical phenomenon which grossly changes the measurement process. For example, in a chemical analysis for trace constituents a contaminant in the apparatus could cause major variation in the measurement. Here explanatory flags should accompany the data corroborating the presence of a contaminant or suggesting the possibility of a contaminant.

Finally, when data are collected over a period of time, there may be changes in the data-collection process; e.g., in the method of reporting, measuring, validating, or summarizing. To sort out such effects, a time stamp should be associated with each datum giving the time when the data were generated, and the time of the particular file update when the data were included.

In many situations it is useful to have a "degree of believability" associated with data. For example, economic data on developing countries may be obtained by estimates. Using such data for economic forecasts or evaluation should take into account the believability of the data. Another source of imprecise data is introduced by imputation. Imputed data values should be marked as such and not interpreted as reliable data.

Current database management systems do not have facilities for keeping track of data quality. Research is needed to find economical ways of storing information about data quality, and to find ways of passing this information to the data analyst.

1.3. Data Sparseness

In many data sets, there are structured patterns of missing data. This is particularly the case for designed experiments where the "design" is an optimum sparse coverage of the independent variable levels. Here the structure allows encoding which could materially reduce database storage requirements.

To reduce storage requirements, designers of databases often change the logical structure of the data. For example, a file may be partitioned into multiple segments, or data values (e.g., year) included with a data element name. This practice can obscure the meaning of the data and complicate retrieval.

Research is needed on the handling of sparse data to find ways to economize storage, to describe metadata, and to optimize retrieval while keeping the logical description independent of storage considerations.

1.4. File Freezing

Many databases are dynamic in the sense that they are continually being updated. If a statistical analysis is to be performed, there will be a natural time cutoff. All data resident in the file as of the cutoff point must be identifiable. Thus there must be a capability to segment on time so that information that comes in after the cutoff will not erroneously get into the statistical analysis and possibly bias the results. As a consequence of file freezing, there may be several versions of the same file in existence.

Research is needed to find techniques that impose proper time constraints on retrievals. Research is also needed to find techniques for efficiently storing multiple versions of large files.

1.5. Imprecise Keys

In statistical analysis, information may be needed from various parts of a single file or from several files. Often, this must be done by making a cross reference linkage using imprecise keys. For example, in a hospital database system, all the information on a patient might be retrieved using the patient's name as an imprecise key to search portions of the same file or several files (name is usually an imprecise key because there may be several people in a database with the same name). A file structure that allows cross referencing with such imprecise keys is very useful for statistical analysis. In statistical databases, subsetting and retrieval using imprecise keys is a difficult question that needs research.

1.6. Security

When a statistical evaluation is to be done on a file that contains sensitive information, the question of privacy protection arises. The confidentiality dilemma is to provide useful summary information while protecting the privacy of the individuals. Suitable mechanisms for protecting information may depend on the logical data model. Research is needed to determine what is obtainable within the constraint of summary information

criteria, and how to provide security mechanisms in a multiuser environment.

2. Functionality/Usage

Several issues were raised regarding the desired functionality or usage of statistical databases.

2.1. Subsetting

The key to successful data analysis lies in finding interesting subsets of the data. This requires the capability for multiple key retrievals or, more generally, for retrieval of any identifiable subset of data (e.g., all PhD's in the age bracket 25-40 living in California and earning more than $50,000 annually). Once a subset of data has been formed and analyzed, it is often desirable to retain the subset for further analysis, for aggregation, or for decomposition into smaller subsets. For example, the salaries for the preceding subset of PhD's may be aggregated by profession or by sex, or the subset of PhD's in the computer industry may be extracted for a more detailed analysis. Because subsets are obtained or retained for the purpose of aggregating or summarizing over certain attributes, they are often called summary sets.

Many commercial database systems have facilities for specifying and retrieving arbitrary subsets. The storage and retrieval mechanisms of these systems are not always efficient, however, for statistical database structures, e.g., sparse data. Research is needed to find efficient techniques for statistical databases; transposed files are a good beginning.

Some commercial database systems support view definitions, which permit subset definitions to be saved and managed by the database system. The data in a view is derived from the current state of the database when the view is retrieved, rather than being stored as a separate data set. With large statistical databases, views may not allow efficient enough access to certain subsets; hence, it may be preferable to store these subsets separately. Additional metadata is then needed for describing the subsets and their relationship to the main database. Research is needed to develop techniques for managing these retained subsets.

2.2. Sampling

In addition to forming identifiable subsets of data, it is often desirable to extract samples of the data. This is particularly true for large databases, where it may be infeasible or impractical to analyze the entire database. Sampling can also provide a means of protecting the confidentiality of sensitive data.

Most existing database systems do not support data sampling. Research is needed to develop efficient techniques for defining, retrieving, and retaining samples, and for combining sampling with other subsetting operators.

2.3. Data Analysis

Many existing database systems have operators for computing counts, sums, maxima, minima, and means. Although full data analysis capability should not be the goal of statistical database management systems (see Section 2.6), research is needed to determine which data analysis operators can and should be included in such systems. For example, it is quite efficient to perform the sampling operations in the data management system. In addition, new methods are needed for accessing complex data structures, e.g., hierarchies, by data analysis programs.

The results of data analysis should be self-documenting; that is, they should contain metadata describing the resulting structure. Existing systems do not provide this capability, and research is needed to develop analysis tools that produce self-documenting structures.

2.4. Adaptive Data Analysis

Data analysis is an adaptive process, where intermediate results determine subsequent steps in the analysis. It is often desirable to go back to an earlier step and try a different path. With appropriate computer graphics, much of the analysis could be done on-line without recourse to hard copy.

Existing database systems do not support this form of adaptive analysis. Research is needed to develop techniques for recording analysis paths, and to develop graphical aids for moving along these paths.

2.5. Historical Data

Traditionally, historical data has been difficult to assemble for analysis. If it is saved at all, it is usually archived on tapes. With on-line database systems, historical data can be retained and retrieved by the database system. Research is needed to determine how historical data is best managed.

2.6. Data Management and Statistical Analysis Interface

The data management software and statistical analysis software should not form a single monolithic system that attempts to provide all capabilities for all users. Even if we could predict what capabilities would be required, it would be difficult to develop and maintain such a monolith. On the other hand, the user interface should provide the image of a single system. The data management and statistical analysis capabilities should be constructed from building blocks that allow their easy interface. Research is needed to determine what building blocks are needed, and to develop a methodology for constructing and interfacing them. Several interfacing styles are possible; for example, the database system may drive the statistical analysis system or vice-versa, or both systems may operate as coroutines.

2.7. Distributed Systems

Local and nonlocal computer networks can provide access to distributed databases and to computing resources not available at the user's personal work station. Several scenarios are possible; for example, data from one or more sites may be assembled at a user's personal work station for analysis; data collected at different sites may be analyzed at the sites (e.g., to reduce the volume), and then transmitted to a central database system for further

analysis; data managed at a personal work station may be sent to a more powerful machine for analysis, and the results returned to the work station, possibly for additional analysis. Before any of these scenarios can be fully realized, research is needed to develop mechanisms for managing distributed statistical data and distributed analysis.

3. Metadata

Metadata is information about data. The panel has repeatedly emphasized the importance of metadata for statistical data. Often data becomes obsolete because the information about its content and meaning is nonexistent or lost. The following is a collection of metadata issues that could benefit from further research.

3.1. Meaning of Data

Most data management systems, as well as statistical packages, have a data definition capability for the specification of a data field descriptors such as type, size and acronym. This type of information is necessary for computer manipulation of the data. However, this information is not sufficient to characterize the meaning of the data to people. A description of the origin of the data, how it was collected, when it was generated and modified, and who is the responsible person for its collection is also needed. The description should include the full names of data entities and an explanation of what they represent. Data types of statistical databases are often complex, such as time series, vectors, or categorical variables. In addition, special types of data values may be required, such as codes for missing, unavailable, or suppressed values.

The lack of metadata is even more acute when data is collected through automatic data systems. Here it is necessary to be able to collect some of the metadata automatically as well.

3.2. Metadata of Subsets

As was mentioned in section 2, a large number of subsets can be generated in the data analysis process. In addition, new data values can be generated by computations over previous data values. The metadata for these newly created data sets include the origin from which the data sets were obtained, the operations (selection, sampling, computations) involved, descriptions of the data elements, who created the data sets, and time of generation.

Most of this information can (and should) be automatically obtained by the system at the time of subset creation. Some additional semantic information must be obtained from the user if he wants to keep these data sets for future use. The open research issues are how to capture and store this information efficiently. In particular, if data sets are generated from each other, they would have much descriptive information in common that should not be stored repeatedly.

3.3. Metadata Management

It is necessary to organize and manage metadata, just as it is the case with data. However, metadata typically contains much text, and its structure can be more complex than just text strings. It is therefore necessary to manage metadata with tools that can handle text. Most data management systems and statistical packages have very limited capabilities in this area.

One should be able to retrieve and search metadata, just as one does with data. For example, it should be possible to ask the system for the data sets generated by John Smith after February of this year, or to search for all data sets that have information about a certain topic in a hierarchical fashion. Research is needed to determine how to organize the (mostly) textual information so that it can be searched, retrieved, updated, and automatically maintained.

3.4. Consistency

Unfortunately, the meaning of terms change over time, and they may be inconsistent across data sets. This occurs often when similar data is collected over long periods of time. For example, the boundaries of a county may be redefined in a certain election year, but the change is not reflected in the name of the county. Clearly, it is invalid to compare data collected for that county over several years which include the change, yet it is commonly done because the corresponding metadata does not reflect the change.

Another reason for confusion is the use of the same terms for different data elements. This occurs often when new data sets are generated from existing ones. For example, one data set may contain information about income generated by an average over the entire set, while another may be generated by averaging over a sample. If both data elements are labeled the same (e.g. income), it is easy to make mistakes in comparing them. These changes should be captured in the metadata, and be readily available when the data sets are used. At the same time there should be a way to indicate that the data elements are related.

The reverse problem is one of using different terms for the same data element. It is particularly important if the same data element, such as "state", is used by more than a single file, since this information is necessary to determine if the files are comparable (joinable) over this data element. Using different terms in the same file requires the support of a synonym capability.

Another related need is the use of metadata for comparing or merging data from data sets whose parameters are similar but not identical. For example, suppose that the partitioning of ages into age groups in two data sets is not the same. In order to compare or merge these data sets on the basis of age groups, one needs the metadata describing the age groups.

3.5. Reformatting

It is not realistic to assume that at some point there will be a standard for data formats over all systems. Therefore, the need for reformatting data is inevitable. Metadata should be used to facilitate the automatic reformatting of databases. Research is needed to determine how to organize the metadata and how to use it for the purpose of reformatting. Perhaps a standard for metadata

specifications can be developed.

3.6. Distributed Data

There is additional metadata that is necessary when databases are distributed over several nodes of a computer network. For example, suppose that data is collected and analyzed at several hospital nodes on patients response to a certain drug. If one was to combine such information, it is necessary to synchronize the state of these databases as well as the correspondence between the items involved. Research is necessary to determine what status information should be kept, and how to coordinate such information for queries that involve several nodes.

There is very little development of distributed systems that can handle statistical data, mainly because the difficulties in implementing such systems seem too great. But, as was discussed by many members of the panel, the trend is indeed towards distributed systems of work stations. As powerful personal work stations come down in price, so it is more likely that future data analysis will be performed on a work station that is connected to other work stations and central machines through a computer network. The central machines are likely to contain data that are of interest and are shared by many users, while the work stations will contain temporary or private data sets that analysts currently work on. Thus, we believe that it is not too early to conduct research in the area of metadata in distributed systems.

4. Logical Models

Logical modeling is that part of database management concerned with the meaning of data collected about the real world. The typical logical model encountered in a statistical textbook is the rectangular array or observation on a case by attribute basis. The current status is that the real world is more complex than the logical models of database systems, but that logical database models are more complex and diverse than the logical models handled by standard statistical algorithms.

4.1. Complexity of Data

The data organizations encountered in statistical textbooks are data matrices or contingency tables. The mathematical machinery used is the matrix and vector algebras or calculus. The traditional interface with computer science has been the numerical analysis of the computational processes needed to implement the arithmetical processes.

When the data becomes more complex, of which the hierarchical relationship of individuals to a family is an example, differing information is relevant in different subsets of the data, and the classical notations quickly loose their elegance and power. In complex situations, the identification of an appropriate unit of analysis, and the collection of data for that unit, may become substantive problems. All of this may have the additional complication of missing and erroneous values. The notation needed to deal with other types of relationships, such as networks, is often weak and has weak associated theory. With complex data structures, the interface with computer science grows to include algorithms and data structures, computational complexity, and database management.

4.2. Missing Data

A common characterization of complex situations is the need to use and identify insightful subsets. In the presence of missing and erroneous data, this may be difficult. The missing data may arise for many reasons - not observed and not defined or relevant are the standard cases. The ability of database systems to approximately deal with the various types of missing data is weak in current practice. The initial machinery typified by the not-a-number symbols (NaNs) of the IEEE floating point standard have not been expanded or integrated into control mechanisms (query languages) of database systems.

4.3. Data Aggregation

The various attributes of data may be more complex than is realized. Hierarchical relationships may be multifaceted in practice. For example, in geographic aggregations, the notion of county and metropolitan area are intermediate between municipality and state and of equal standing; either may be embedded in a strict hierarchy. The form of the aggregation may change over time so that both analysis and representation are further complicated. Simple responses may be either multiple or repeated in practice. The representation of complex data which has been fully and correctly observed is now possible, but the methods to deal with partially or incorrectly observed data have not been developed.

4.4. Documentation

The logical data model is part of the description of the data and should be included in the documentation of the data. The metadata has the role of communicating both the internal technical facts about the data, including the data models used in its representation, and the external information available about the data. The meaning of the data may be derived both from the data models and the external knowledge about the data.

Logical data models should be associated with good analysis methods. The models that are available await analysis techniques, some of which may arise in the interaction of statistics and algorithm design. Some of the known problems with existing models are the identification of appropriate analysis units, and the bringing of data to those units. The current algorithms often are weak in the presence of the various forms of missingness and errors present in data.

Acknowledgements

Mervin Muller joined some of our discussions, and we are grateful to him for sharing with us his experience and insight.

References

There is an extensive literature covering the different aspects of statistical databases and statistical software. Instead of giving a long list of references, we mention a few surveys and collections of papers, all of which con-

tain many references.

Reference 1 below is an introductory paper to the area of statistical databases. It discusses several problem areas and surveys some existing solutions and work in progress. Reference 2 discusses extensively metadata structures and needs. Reference 3 discusses the security aspects of statistical databases, and surveys existing and proposed controls. Reference 4 contains numerous papers and abstracts presented at a specialized workshop on statistical database management. Reference 5 is a large volume that describes and compares statistical packages and other noncommercial statistical software. Reference 6 is the proceedings of an annual conference that has been held over the last 15 years, and that contains (especially in the more recent issues) several papers on statistical databases.

1. Shoshani, A., Statistical Databases: Characteristics, Problems, and Some Solutions, *Proc. Eighth International Conference on Very Large Data Bases*, Sept. 1982, pp. 208-222. (Copies available from: VLDB Endowment, P.O.Box 2245, Saratoga, CA 95070.)

2. McCarthy, J.L., Metadata Management for Large Statistical Databases, *Proc. Eighth International Conference on Very Large Data Bases*, Sept. 1982, pp. 234-243. (Copies available from: VLDB Endowment, P.O.Box 2245, Saratoga, Ca. 95070.)

3. Denning, D.E. and Schlorer, J., "Inference Controls for Statistical Databases," *IEEE Computer*, (to appear July 1983).

4. *Proceedings of the First LBL Workshop on Statistical Database Management*, Dec. 1981. (Copies available from: Computer Science and Mathematics Dept., Lawrence Berkeley Laboratory, Berkeley, Cal. 94720.)

5. Francis, Ivor (Editor), *A Comparative Review of Statistical Software*, 1977. (Copies available from: The International Association for Statistical Computing, 428 Princes Beatrixlaan, 2270 AZ Voorburg, Netherlands.)

6 *Proceedings of the Computer Science and Statistics: Annual Symposium on the Interface.* (Copies available from different places, depending on the year of the symposium.)

How Do Analysts Work? User Interface Issues

G.A. Marks
Inter-university Consortium
for Political and Social Research
The University of Michigan
Ann Arbor, Michigan, 48106-1248

G. Anderson, McMaster University
A.D. Elliman, Brunel University
A. Goldman, University of Minnesota
J. Klensin, Massachusetts Institute of Technology
B. Meierhoefer, The Federal Judicial Center
P. Svensson, Swedish National Defense Research Institute
J.J. Thomas, Pacific Northwest Laboratory
M. Wartelle, Institut Gustave-Roussy
L. Wong, Lawrence Berkeley Laboratory

1. The Nature of Data Analysis

The data analysis process begins with the identification of questions or problems which to be resolved require factual or empirical information. That information is determined through use of appropriate data in conjunction with display or statistical methods which clarify and focus the issues of concern. One or more people, referred to in this text as data analysts, guide this process and mediate between the reality of what information may be learned from the available data and analytic methods, compared to the abstraction which prompted the quest.

An essential task within this process is the gathering of the needed data. There are many methods of obtaining data. It might be done through sending questionnaires to individuals or organizations through the mail. It might involve collecting readings from seismological instruments. Basic data might come from inspectors at quality control points on a production line. A historian might be utilizing the financial accounts of a 16th Century business. Data might be obtained from some organization that has already done the basic work of collection, such as the national census or economic indicators, or public opinion surveys. The sources of data for analysis are essentially as varied as the imagination, curiosity, and resources of the investigator permit.

Data collection involves substantial thought and care in the selection of what is gathered, in the measurement of the values, in avoidance of transcription errors throughout the process, and in solid documentation of all relevant information. This must be done with clear attention to the ultimate information desired

from the data. Experienced data analysts know that there will be many unanticipated uses for the data, and there have evolved a variety of procedures which improve the prospects of being able to easily accommodate the unexpected. One such procedure is to avoid having only a summarization of any part of the data in the basic data file: this leaves the choice of approach open. Another strong rule is to very carefully document the data, the collection process, the database organization, and the analysis steps already completed. A great deal of time on the part of data analysts is spent trying to uncover what someone else did to the data, or what one's own work of only a short time earlier did.

Data analysis involves use of data in large quantities, because the nature of the scientific method, experimental design, and statistical inference all point to repeated measurements, to replication of the experiment, to a large enough sample of cases to permit valid analysis. Thus a particular statistical computation is likely to involve passing over the data for hundreds, thousands, or even more cases. Furthermore, because the selection of what to measure always means a simplification and abstraction of reality, it is very common to find that many different things are measured for each case, with the consequence that there are many fields or items or variables in the database. Because these are all part of a representation of reality, the analyst typically looks at as many of the plausible relationships in the data as possible.

Thus a data analyst uses an entire database in a very active manner. Even more important is that the manner in which the data are retrieved, i.e. the physical access sequence, is essentially orthogonal to the typical transaction-oriented or case-oriented approach of commercial database management systems. But even that statement does not capture some profound differences in how the analyst approaches the use of data.

It was stated above that an analyst wants to keep data in as raw, detailed, or unsummarized form as possible. This is a critical point. This is intimately related to a very common task during the data manipulation phases of data analysis: the calculation of derived measures, indices, and scales, or the recoding and regrouping of data values into summary measures. Examples of physical ratios such as energy consumption in buildings in BTU/hour/square foot are clear. Less obvious might be a table lookup based on a persons height and age, with a deviation calculated between the ideal value and the actual weight. An even less obvious instance would be the use of five measures which give each person's attitude about the use of handguns under various specific conditions, combined into one overall measure of the person's view of handguns. A typical data analysis process involves a very large portion of time devoted to constructing such derived measures, checking that they were done correctly, examining the analytic consequences, and moving on to try still other approaches based on what was

learned in the last step. This is one of the major places where the analyst wants to keep the raw, unsummarized data available, so that alternative summary measures may be tried as needed. It is also one of the most common situations in which documentation of what was done and what was found ends up being inadequate.

There is another very common aspect of how an analyst uses a database which relates to the need to keep the unsummarized data as well as to more general questions of analysis strategy. Usually the cases of the database may be divided into subgroups. One example would be to create subgroups based on a demographic variable, such as male or female, region of the country, religious or ethnic groups, or some combination of such variables. Other examples would be to split the groupings in a database between productive oil wells and those which are non-productive, or between people who received a specific medical treatment and those who did not. Separate analysis results would be done on each subgroup, possibly for comparison between the groups. Summaries of the subgroups are sometimes produced, such as counting the number of cases in the subgroup, or determining average, minimum, or maximum values on specific variables. These values might then be merged back into the individual cases. For example, this would allow a comparison of the production of an individual oil well with the average production of all oil wells in a specific oil field. Thus a common part of the data analyst's work is often the identification of relevant subgroups, and the generation of simple computational results or statistics which may be used for either within-group or across-group comparisons. This is yet another process where having access to the unsummarized data is essential. It is also unfortunately often a very inadequately documented portion of the analysis work.

The description of the analysis process above hopefully conveys the image of a lot of data manipulation taking place, through a highly iterative and exploratory process, and with a great deal of need to document much of what is done. Most data analysis projects are predominantly involved in data management rather than computation of statistics or generation of reports and graphical displays. Data management may be as much as ninety percent of the overall effort. Thus it is clear why data analysts are interested in systems for database management. The discussion of the nature of the work should also reveal why many existing systems are not very well suited to such applications.

This general discussion of the nature of the data analysis process sets the stage for many possible discussions of the design of statistical database management systems. Of particular interest here is the question of how the user interface ought to be designed. Exploration of this topic begins with some further characterizations and distinctions in the way data analysts conduct their work.

2. Issues Concerning the Nature of the Data Analyst.

2.1 Who is the Data Analyst?

The image presented in this discussion so far is of a single individual handling all aspects of data analysis. This is true in many situations, but there are also many variants. A common one is to have a research team, with one or more senior persons and then a support staff. Often the senior person does little of the actual computer work, while a support person in this situation may spend virtually full time on the computing aspects of the overall data analysis efforts. The computing system that feels efficient and productive to the support person may be unintelligible to the senior researcher. This often means that the senior person may get no chance at direct work with the data, but the even more serious implication is that this situation carries many opportunities for misunderstanding and miscommunication between these people. The senior person may have an unclear or wrong model of either the software or the data, or both, and the support person probably does not comprehend the full research agenda in the mind of the senior person. Resolution of such problems involves both system design which can be quickly grasped, and design of output from the system, including metadata, which both reinforces the model of the system and data evolving in the user's mind, and gives documentation of events and points to potential problems.

An extension of this theme arises with the notion of a person as a consumer of the results of data analysis. Today there is usually a person acting as intermediary in such situations, but the emergence of national networks and information services presents new problems of people who oversimplify a result or simply misunderstand. The database should contain metadata which may be used in at least some situations to warn the consumer of problems. This implies a much tighter linkage between the database system and the analysis software than is common today.

2.2 What is the User's Model of the System and Data?

The data analysis process as already described is plainly a very complex and demanding collection of tasks. Anyone watching a data analyst or a staff of such people will quickly notice many points at which they get lost and confused about what is happening. There is simply a great deal to keep track of, between the nature of the data and the several components of the computing system, and when coupled with the frailties of the human memory and logic, things go wrong. Thus it is very important that the statistical database management system be as readily understandable as possible.

A particularly important implication of this problem of having an understandable system is that there may be a strong need for there to be a relatively tight and direct relationship between

the logical model of the data which the system designer intends the user to employ and the actual physical storage arrangements. Several factors come into play here. One is that the data analyst must know the data intimately. Another is that the analyst has a number of logical models of the world which map from one to the other in order to view the analysis process: the underlying reality, that which was captured by the measurement methods, the database representation of the values and structure of the measurements, the database as it goes through major manipulations over time, the summaries of the data given in statistical output, and the communication of what was found through written word and public presentations. If the statisitical database system creates a large gap or permutation from one layer of the model to the next, the analyst will find the work very difficult and regard it as the fault of the system, quite appropriately.

2.3 What Variations are there in the Orientation of Data Analysis Work?

The overview of the data analysis process presented above contains a description of the most common orientation for the process: the iterative exploration of the data. Another orientation is the generation of routine reports. Here the analysis process is relatively fixed and is simply repeated each time a new collection of data is available. An example of this is many of the periodic government indicators of the state of business and the economy. Yet another orientation is beginning to emerge, that of an automatic analysis by a knowledge-based or expert system. Here the data analysis is in the exploratory mode, but computing systems with appropriate rules and heuristics would perform at least an initial scouring of the data.

2.4 How Do We Treat the User?

A basic question in the design of any software is what level of user competency to presume. Statistical systems have been designed which at least partially prevent statistics from being calculated with the "wrong" type of data. Experience indicates that more damage is done by this in terms of user frustration than is gained by avoiding bad results. Thus there may be good reason to allow "creative misuse," at least in this basic sense of ignoring statistical orthodoxy.

Yet there are circumstances that almost everyone seems to have experienced where data was given to someone else to analyze, with real concern that the other person could use it inappropriately with very serious consequences. These might be political data, information on toxic contamination, or studies of clinical treatment results, or any of a variety of substantive areas. If a subsequent analysis is wrong, whether because of incompetence or strong pre-conceptions of the desired outcome, the impact is equally bad. There is often a strong

sense of need to have a neutral analyst monitoring the work, or of at least placing strong warnings in the metadata about potential pitfalls in the interpretation of the data. The latter seems to be very difficult to do in any practical degree.

2.5 How Much Do We Really Know?

There has not been nearly enough systematic research on how data analysts perform their work. Potential approaches include the automatic retention of on-line histories (a number of these appear to have been collected but never systematically analyzed), the use of controlled observation, having users keep diaries of events and reactions, and tieing of the other methods together with studies of attitudes toward the systems. Among the more notable instances of such work are the reports of the CLINFO design effort ([PAL75]), the Flair system for prototyping a user interface ([WON82]), and the research on user reactions to the evolving designs at Xerox PARC ([SMI82]), and most recently for the Apple Lisa ([TES83]).

3. What Should the User Interface Design Emphasize?

The data analysis effort has already been described as one with enough difficulties to make very close attention to the design of the user interface very worthwhile. The intention here is to range broadly over all the aspects of the system which shape how it "feels" to the user as part of the user interface.

3.1 The User's Model

A fundamental concern in the design of any user interface is the fit between what the user thinks and expects the system will do and how it actually works. If the fit is very good right from the first user trial of the system, the user will find the system easy to use and understand. As already noted, the data analyst actually carries an entire structure of interrelated models around as an overall conception of the task. Really good design of the user interface must recognize and find a solution, a difficult challenge.

3.2 Flexibility and Complexity

The array of data storage and manipulation capabilities needed in a statistical database management system are very substantial, such that there is a clear problem of trying to balance the desire for a complete facility against a readily understandable, coherently modeled design. One approach to resolving this conflict is to partition the system so that functions are clearly separated and do not interact. Then the user can operate with one of them at a time. This of course still requires very careful attention to the overall integration of these partitions.

Another important way of resolving this conflict is to provide

the user with tools for shaping the system to their own work style and needs. In simplest form this could be some form of macro preprocessor as is becoming common for operating system command languages. A more challenging approach would be to identify a rich set of primitives and then build upon them with a user-extensible system. This must be very carefully done or many of the other objectives of interface design will be seriously compromised.

There might be a separate system for handling the user interface, just as many of the microcomputer systems have standard interfaces, such as in the Apple Lisa system or in VisiOn from VisiCorp ([SEY83]). Such a system would ensure a basic level of uniformity between subcomponents of the system, the degree depending on the sophistication of its design. This system might be broad enough to create a bridge to the statistical software used by the data analyst, a useful addition since many observers feel it is unwise to have a single monolithic system encompassing all of the database management and statistical functions, if for no other than pragmatic reasons about the size of the development task.

3.3 System Efficiency and Predictability

Data analysts often use computing resources at a rate which makes them very sensitive to costs. This in turn makes them alert to anything which appears expensive. Thus the basic cost of execution is certainly an important issue, as is the related question of how fast the system can complete the desired work so that the data analyst may proceed to the next step. There remains substantial argument among the ranks of data analysts about the virtues of interactive data analysis, many feeling that a good analyst should not move too fast, they should think carefully about what they are seeing and doing. Nonetheless, everyone likes to get their routine work done as quickly as possible, so system performance is an important issue. For users who are especially concerned about cost, a system capability to give reasonable estimates before proceeding may be very helpful.

Beyond raw performance is consistency and clarity of action. It is obvious that the system needs to be robust in its implementation, where this should encompass the delivery of clear diagnostics to the user when problems do occur, such as for failures in the hardware containing the database. This also implies that the system includes facilities to ensure the quality and integrity of the data are maintained. This could range from automatic logging of changes and backup, to having validity and consistency easily specified and applied to derived measures. A desirable but seemingly expensive feature would be to allow the user to undo specific work, so that when a user or system error is encountered there is the option of returning to an earlier state.

3.4 Display and Interaction Modalities and Style

The rapid advance of technology offers many new opportunities for presentation of information to the user ([BLY83]) and for obtaining input such as commands for the system ([WAR83]). The most fundamental change is that high bandwidth between the computer and the display is widely and economically available. The possibilities are covered at length elsewhere in this publication.

References

[Bly83] S.A. Bly, "Interactive Tools for Data Exploration," Computer Science and Statistics: Proceedings of the Fifteenth Symposium on the Interface, J.E. Gentle, ed., North-Holland, Amsterdam, 1983, 255-259

[Pal75] N.A. Palley and G.F. Groner, "Information Processing Needs and Practices of Clinical Investigators - Survey Results," AFIPS National Computer Conference, 1975, 44, 717-723

[Sey83] P.B. Seybold, "VisiCorp's VisiOn," The Seybold Report on Professional Computing, 2, 3 (November 21, 1983), 1-25

[Smi82] D.C. Smith, et.al., "Designing the Star User Interface," BYTE, 7,4 (April 1982), 242-282

[Tes83] L. Tesler, "Enlisting User Help in Software Design," SIGCHI Bulletin, 14, 3 (January 1983), 5-9

[War83] R.W. Warfield, "The New Interface Technology," Byte, 8, 12 (December 1983), 218-230

[Won82] P.C.S. Wong, "Flair - User Interface Dialog Design Tool," Computer Graphics, 16, 3 (July 1982), 87-98

Workstations and Special Purpose Hardware

Peter B. Stevens
Bureau of Labor Statistics
Washington, D.C. 20212

R. Becker, Bell Laboratories
W. Benson, Lawrence Berkeley Laboratory
H. Farsi, University of Alberta
S. Fok, Technology Development of California, Inc.
J. L. Raymond, Ohio Bell Telephone
P. Wake, Office of Population Censuses and Surveys, U.K.

1. Introduction

The strong interest in a hardware-oriented topic, e.g., workstations, at a primarily software-oriented workshop is testimony not only to technological progress but also to the potency of an idea whose time has come. This new equipment can dramatically alter the environment within which statistical computing takes place, so we focused our discussions on how the environment can interact with the nature of the task. This approach implicitly rejected the notion that a complex computational task, such as statistical database management, can be usefully defined independently of the computing environment in which it exists. Although efforts to apply new technology to statistical database management are only just beginning, it seems clear that it is now possible to create interactive systems which are truly responsive to a broad range of statistical database management problems. In other words, workstations have become big enough, fast enough, and cheap enough to be really useful.

2. Hardware Requirements

The word "workstation" was used in the title because other terms in use such as "microcomputer" or "personal computer" cover too wide a range of machines to be useful. Here a workstation is a computer to be used by a *single* person that provides a constantly available, high bandwidth connection to *complex* computational processes. Once loaded with programs and data, a workstation does not need to communicate with any other machine to perform important tasks. Computers with the necessary power are not new. The personal and affordable characteristics of current workstations are new.

The following areas were deemed most important among the minimum requirements for a workstation to support statistical database management and

statistical computation: memory size, disk storage speed and capacity, arithmetic precision, graphics resolution, cost, and availability of multiple sources for hardware and software.

To qualify as a useful statistical workstation, we agreed that the minimum memory requirement is now 256K bytes. In the future 512K or 768K will probably be needed as quality statistical and graphics software become broadly available for workstations. The 5 1/4-inch Winchester disks are now widely available and provide higher reliability, better responsiveness and much greater capacity than do "floppy" disks. The most commonly available units now hold about 10 megabytes of data although much higher capacity units (in excess of 100 megabytes) are on the market. These capacities meet all the storage requirements for small surveys and can hold useful working subsets of the data for almost any survey. Long format (64-bit), floating point arithmetic is necessary not only for accurate statistical computations but also because this format is very widely used as a database storage format on mainframes and minicomputers. For workstations, several hardware implementations of 64-bit, floating point arithmetic exist, and most conform to the proposed IEEE standard[Kah79]. The best known implementation is the INTEL 8087 coprocessor which provides 80-bit precision for intermediate results. (The irony here is having more accurate results available on a workstation than on a huge IBM mainframe computer.) Graphics resolution is a difficult area to specify because graphics are not now widely used in statistical database management. We noted that the Apple Lisa computer gets very good performance from a resolution of 720 x 364 dots and decided that should be sufficient. The specifications given here could be met by many minicomputers. Much of the excitement about workstations can be explained by the fact that machines that meet these specifications can be purchased for less than $5,000 from several different manufacturers.

The last, but perhaps most important, requirement is that multiple sources for hardware and software exist. This requirement recognizes that the microcomputer industry has developed in a very different way than did the older forms of the industry. No leading manufacturer, not even IBM or DEC, actually produces the hardware and software that it sells to end-users. These firms only assemble and integrate components and software systems which have been purchased from other sources. Hardware suppliers such as Motorola, Intel, Zilog, and National Semiconductor, and software firms such as Microsoft, Digital Research, and Visicorp sell their products to many manufacturers. A very large, active, and competitive third-party hardware and software industry is responsible for the very rapid development of the microcomputer industry. The conclusions that we drew from these facts is that users can no longer rely on a single supplier for up-to-date hardware and software. Standardization is taken seriously in the microcomputer industry, and we had best prepare ourselves to deal with many different suppliers for our equipment.

3. Areas for Further Research

The dramatic advances in computing power and reductions in cost create only the potential for impact on statistical database management. To say that good software must be created and that the new workstations must be effectively integrated into the statistical computing environment are both obviously true but not very helpful. These questions were considered at length, and attention was focused on four areas that appear to be really new. Three concern the way that people can use computers and the other concerns machine-to-machine communications. For each area much research needs to be done before the potential of cheap, fast hardware can lead to better quality statistical data and analysis. We discuss each area below.

The first is the use of graphics for data management functions. Again, there is some irony in the fact that people use directed graphs frequently in designing and maintaining data bases, but computer support appears rare. Systems flowcharts, data flow charts, Yourdon diagrams are examples, and they are almost always prepared manually, slowly, and at great cost. A workstation can provide a data flow bandwidth measured in the millions of bits per second versus the few thousand bits per second possible over communications lines. This increase in bandwidth, along with the consistency of response that comes from having only a single user, are the primary advantages that a workstation must always have relative to any computer which is shared over communications lines. The result is that a directed graphs system can have virtually instantaneous response time--which is our definition of truly responsive. Graphics systems are not the only ones that benefit from a very high bandwidth and consistent response time. Many systems approaches which have been theoretically possible but not successfully implemented on larger, time-shared computers become practical and economical on workstations. The next two areas are examples.

A second area is the potential for improving data management by allowing data analysts to *perform* actions directly rather than only *specify* them. The potential improvements are most dramatic when the desired actions are very difficult to specify completely. This type of problem is seen frequently in the data preparation and maintenance phases of a statistical analysis project. When data are acquired from different organizations or when survey data have a complex structure, substantial efforts may be required to edit, clean up, reformat, merge, etc. the data into a usable form. The tools now available are typically the traditional batch programming languages, so computer program development is a major cost and time consideration for project planning and budgeting. Worthy questions remain unaddressed because the time and cost requirements for both programming and production are so high. The inadequacy of existing tools for data manipulation and management is a well known problem in the statistical computing community. The point being stressed here is that contemporary workstations offer the potential for new solutions.

When merging two data files, one of the most common data preparation operations, the algorithm to match records is usually fairly easy to specify. What

are hard to anticipate and specify are all of the possible actions to try on near misses. A good example of both the analytic potential and processing problems can be seen in the monthly Current Population Survey (CES)[Mor65]. The results of this survey are extensive labor force statistics, published by the U.S. Bureau of Labor Statistics. The best known CPS statistic is the national unemployment rate. The source data (microdata) for this survey has substantial analytic potential, but the matching and merging of the monthly files presents substantial difficulties.[McI80] A workstation-oriented interactive approach shows promise.

A merge operation in effect partitions the files into three parts: the records that clearly match, those that clearly do not match, and the doubtful cases. If the computer system could route the doubtful cases to an analyst's workstation for review and decision, several benefits would be seen. First, the analyst would be relieved from the burden of attempting to anticipate and provide for all possible doubtful cases. Second, the analyst would be able to directly review and dispose of each doubtful case individually, thereby producing better data. Finally, the total time and effort required would be significantly reduced. For the CPS data, as for many other examples, it is easier and faster to deal with the doubtful cases that actually occur than it is to imagine and specify the handling of all possible cases. The hardware environment for such an interactive file matching system must have the very high bandwidth and instant response that only a dedicated workstation can provide.

A third area is the development of *recognition* as opposed to *recall* systems. In a recognition system the user must select (recognize) the desired action from displayed alternatives. In a recall system the user must construct (recall) a syntactically correct command to cause the desired action to take place. In general, recognition systems are easier to learn and to use than recall systems. Menu-driven systems for word processing or data retrieval are examples, but only primitive ones, of recognition systems. Traditional programming languages such as FORTRAN and PL/1 and all statistical data management or analysis systems such as SAS, SPSS, etc. are recall systems. The high processing bandwidth, graphic capabilities, and consistent response of dedicated workstations allow all sorts of new approaches to recognition systems. In workstation software these new techniques so far have been applied mainly to word processing, document processing, and office automation tasks. Machines such as the Xerox Star or Apple Lisa are the best known examples, but many others exist. The application of these techniques to the processes of statistical database management and usage is just starting. Problems in database design or in navigation through a complex set of relations would seem to be good candidates for graphic- and recognition-oriented systems.

The fourth area is the integration of workstations, in large numbers, into the current statistical computing environment. Local area networks (LAN) appear to offer solutions to most of the problems of managing software and data access in a highly distributed computing environment[Dat83]. A LAN is a very high speed and highly reliable communications link by coaxial cable between workstations and shared network resources, called servers. The servers provide for central software libraries, data file storage and backup, and access to specialized devices such as high

speed printers, etc. Interactive or high-speed batch communications with mainframe computers is possible through communications gateways. Many varieties of LAN have been announced with Ethernet being the best known. The IEEE Project 802 committee is developing standards for Ethernet[Amb82], and many independent hardware and software companies are marketing or developing products.

A full discussion of LAN issues and problems was beyond the scope of the working group's mission and the time available. We noted that a potential for proliferation of incompatible and noncommunicating workstations exists just as has frequently happened with word processing machines. Using a LAN as an essential part of the workstation environment offers great promise for dealing with both the technical and management problems of highly distributed computing.

References

[Amb82] J. Ambrosio, "International LAN Standard Finalized, Finally," *Information Systems News*, (December 13, 1982).

[Dat83] Datapro Research Corporation, *Perspective: Local Area Networks*, Delran, N. J., 1983.

[Kah79] W. Kahan and J. Palmer, "A Proposed IEEE-CS Standard for Binary Floating Point Arithmetic," *Computer Science and Statistics: 12th Annual Symposium on the Interface* (J. F. Gentleman, editor), May 10-11, 1979, 32-36

[Mor65] J. E. Morton, *Analytical Potential of the Current Population Survey for Manpower and Employment Research*, (Kalamazoo, Michigan: The W. E. Upjohn Institute for Employment Research, 1965)

[McI80] R. J. McIntire, "The Mechanics of Matching CPS Microdata: Problems and Solutions," *Using the Current Population Survey as a Longitudinal Data Base*, (Washington, D.C.: U.S. Department of Labor, Bureau of Labor Statistics, 1980).

Connecting Heterogeneous Systems and Data Sources

Sandra Heiler
The World Bank
Washington, D. C. 20433

A. Timothy Maness
Department of Human Genetics
University of Utah
School of Medicine
50 North Medical Drive
Salt Lake City, Utah 84132

A.N. Arnason, University of Manitoba
Robert Burnett, Battelle Pacific Northwest Laboratories
Michael Fox, Jet Propulsion Laboratory
Andrew Kramar, Institut Gustave-Roussy (France)
Deane Merrill, Lawrence Berkeley Laboratory
Robert Muller, ORACLE Relational Software Inc.
Gordon Schiff, Hoffmann-LaRoche, Inc.
Stephen Weiss, Bureau of Labor Statistics

1. Introduction

This paper is the result of discussions by the working group on "Connecting Heterogeneous Data and Systems." The topic is important for two reasons. First, heterogeneity in data and systems is a fact of life. Existing data and systems do differ from each other, and analysts need to use them together anyway. Second, heterogeneity is here to stay and it may even be a good thing. New systems will continue to be developed and it is unlikely that a comprehensive system, one that does everything on a universal data structure, could be developed. Even if it were, old systems and data would continue to be used. In addition, users themselves will continue to differ from each other in the tasks they want to perform and in their levels of sophistication and interest in the process.

The paper describes some of the reasons for heterogeneity among data systems, the ways in which problems of dealing with such data and systems manifest themselves, and some approaches to avoiding or solving these problems.

2. The Reasons for Heterogeneity Among Data and Systems

Data analysts and computer scientists are familiar with the enormous differences that exist among the data and systems in their working environments. Their available hardware and software resources, the data they are dealing with, and the methods they use to accomplish their tasks are largely determined by factors outside their control -- convenience, availability, the extent of their knowledge, past history and budgetary limitations.

Heterogeneity is the natural result of experimentation and growth in a rapidly changing field. Even when elegant solutions are found for difficult technical problems, the dominant hardware and software manufacturers find it in their best interest to keep changing their products rapidly enough that imitators cannot keep pace. The most important reason for heterogeneity, however, is desire to meet changing needs and to continuously take advantage of advances in technology, experience, and theoretical knowledge.

Not only analysis tools but even the data themselves change over time. Data are meant to describe the real world, which changes over time. For example, advances in medical knowledge prompt changes in disease classification about every ten years. In order for research results to be of use to physicians and clinicians, data collection and analysis methods must adapt to the new classifications, regardless of the inconvenience caused to the analyst.

The reasons for heterogeneity in data and systems are real, and not the result of laziness or poor planning. Equally real is the frustration that these differences cause to programmers and data analysts, who must master multiple techniques for accomplishing simple taks.

3. How Heterogeneity Problems Manifest Themselves

Data

Heterogeneous data may cause problems in the ability to access data, the ability to understand data, or the ability to establish consistency in structure or meaning with other data. The forms and causes of heterogeneity among data may include:

- changes in classification scheme as knowledge increases (e.g. disease classification), scope (e.g. district boundaries changing but names remaining), collection method, or collection intent;
- differences in physical format, units or levels of resolution (for continuous domains such as wind speed) or parameters (e.g. weekly vs. monthly time-series);

- undetected disparity in meaning due to insufficient description (e.g. use of the same name for different items or different names for the same items.

Systems

The analyst is faced with an almost overwhelming array of database systems, statistical packages, etc., and the number of systems continues to increase. Most of these systems were written in an attempt to make life easier for the user by simplifying his access to data or functions, doing the data management or analysis better, or providing him with new functions or combinations of functions. However, none of the systems does everything. The lack of a "super" system -- one that provides all of the functions, equally well -- means that the user must be aware of what packages exist so he can choose the one that is best for his job, and he may have to know the details of how to use more than one system to do a single job.

Different methods of data definition and different data structures add to the complexity of doing an analysis. In addition, data (and metadata) may be lost in going from one system to another because of differences in capabilities. The need to use different computer systems may further compound these difficulties.

Users and Uses

Not only do data and systems differ from each other, users and the uses they make of the data and systems may also differ from each other. It is unlikely that a single user interface could be developed that would work for all classes of users for these reasons. First, users shouldn't have to use systems that are more complex than the problems they want to solve and they will naturally choose the simplest system that will work. Second, they shouldn't have to learn new languages unless the time it takes from their primary function, which is to perform analysis, is made up by being able to do better or faster analysis. Third, the language of a system or tool affects how the user thinks about his problem. Some problems are analyzed and solved better in one language than another.

3. Approaches for Avoiding or Solving Problems

The following approaches may help to avoid or solve problems of dealing with heterogeneous data and systems:

- o Better, more complete metadata
- o Better metadata management
- o Imposition of standards

- o Interfacing
- o Integration
- o Hiding heterogeneity from the user.

Each approach is described briefly below.

3.1 Better, More Complete Metadata

The most obvious way to minimize problems of connecting data that differ in source or structure is to provide complete data documentation. Metadata from each source should include: machine readable code books, errata and caveats (footnotes) for each field value whose interpretation needs clarification, data sources, collection methods, and the (original) intended use of each field, as well as structural information, key fields, link paths, etc. that relate fields and records.

Some or all of this information will be required to establish that fields or records originating in different source databases (possibly at different times) can validly be linked and, if so, how. Where metadata indicates inconsistency, additional metadata or procedures should be provided to map them to a common standard, for example, mappings of population by county onto earlier county boundaries.

When heterogeneous systems are linked, some of the metadata described above become important in minimizing the need to re-provide data descriptions to each system when passing from one to another. It can also facilitate automating data restructuring and other transformations needed to make use of another system.

3.2 Better Metadata Management

Metadata management (MDM) can help solve the problems of handling heterogeneous data by providing the tools to collect the metadata in the first place, to interpret structural differences and to create consistent merged data structures.

The MDM software needs capabilities for creation, maintenance, query and display of metadata, as well as for transforming data from one structure to another. In some cases, the facilities of the data management system itself can be used for MDM. The software should accommodate metadata that are generated when the data are collected, and to automate the addition of the metadata when the data are loaded into a system. It should also make it easy for the user to add additional descriptive information. The ALDS project [Bur83] is an example of current work in this area.

3.3 Imposition of Standards

A somewhat unpopular but often effective approach to avoiding problems of dealing with heterogeneous data is to impose standards. For example, the use of SMSA's (Standard Metropolitan Statistical Areas) is useful in integrating data from multiple economic surveys.

In cases where a set of standards can be imposed and maintained over a long period of time, the problems of heterogeneous data virtually disappear. Unfortunately, such cases are rare. The problem with standards is that they become obsolete. It is unreasonable to require that new data be collected using old classification schemes just to maintain consistency across time. Standards do, however, provide a valuable function even when they are frequently changed. This function is in the area of metadata. When the standard is included or referenced in the metadata, it provides the analyst with a precise specification for the data that can be compared with specifications for other data he uses.

3.4 Interfacing

The most obvious solution to problems of using heterogeneous systems is to build interfaces between them. Since not all systems will be directly connected there will be considerably fewer than n * (n-1) interfaces.

One approach to interfacing is to define a standard for the physical representation of the data which is recognized by the cooperating systems. This was the approach taken in SEEDIS [McC82] which used the self-describing CODATA format [McC82a]. The advantage of this approach is that each new system added only needs to be able to communicate with the standard format in order to communicate with all systems. The disadvantage is that it is possible in only a limited number of systems without changes to the software of the systems involved.

Another approach is to tightly couple pairs of systems. In this approach, individual systems develop independently with the onus of translation being born by the calling system. Examples of this approach are SAS calling BMDP [SAS 82] and the TPL/RAPID interface [Wee81].

A third approach is to develop an executive system that handles the problems of interfacing systems. An analysis can be specified in a special language. The executive will take the specifications and generate the control statements for the statistical package and the DML statements for the DBMS, and execute them in a way that the procedural nature of the process is hidden from the user.

An executive system that supports a high level interface would need to perform the following tasks: 1) validate the operations, 2) generate the retrieval clauses from the retrieval requests, 3) apply these clauses to the DBMS, 4) put the retrieved data in a form suitable for the statistical package, 5) generate the commands for the statistical package, 6) apply these commands to the statistical package and make the data available, 7) analyze any error conditions and take appropriate action, and 8) present the results to the user and return to process the next high level specification.

Examples of this are the Generalized Package Interface [Hol81] and PASTE (Put Application Systems Together Easily) [Wei83]. A particularly nice feature proposed in PASTE is not only generating code to transform the data but also generating code to transform the data definition.

4. Conclusions

Even though work is progressing in the area of dealing with heterogeneous data and systems, we are far from a solution to these problems. Consistent, high level user interfaces are needed, irrespective of the implementation techniques employed and such an environment cannot be provided without a considerable increase in the ability to manage metadata.

References

[Bur83] R.A. Burnett, P.J. Cowley, J.J. Thomas, "Management and Display of Data Analysis Environments for Large Data Sets," Proc. Second International Workshop on Statistical Database Management, 1983, 22-31.

[Hol81] L.A. Hollabaugh and L.T. Reinwald, "GPI: A Statistical Package / Database Interface," Proc. Frist LBL Workshop on Statistical Database Management, 1981, 78-87.

[Loh83] G.M. Lohman, J.C. Stoltzfus, A.N. Benson, M.D. Martin, A.F. Cardenas, "Remotely-Sensed Geophysical Databases: Experience and Implications for Generalized DMBS," ACM SIGMOD Proc. International Conference on Management of Data, San Jose, 1983, 146-160.

[McC82] J.L. McCarthy, "Enhancements to the Codata Data Definition Language," Lawrence Berkeley Laboratory, LBL-14083, February 1982.

[McC82a] J.L. McCarthy, D.W. Merrill, A. Marcus, W.H. Benson, F.C. Gey, H. Holmes, C. Quong, "The SEEDIS Project: A Summary Overview of the Social, Economic, Environmental, Demographic Information System," Lawrence Berkeley Laboratory, Document PUB-424, May 1982.

[SAS82] "SAS User's Guide: Basics," 1982 Edition, SAS Institute Inc., Box 8000, Cary, North Carolina 27511.

[Wee81] P.L. Weeks, S.E. Weiss, P. Stevens, "Flexible Techniques for Storage and Analysis of Large Continuing Surveys," Proc. First LBL Workshop on Statistical Database Management, 1981, 301-311.

[Wei83] S.E. Weiss and P.L. Weeks, "Paste -- A Tool to Put Application Systems Together Easily," Proc. Second International Workshop on Statistical Database Management, 1983, 119-123.

Time Series and Large Econometric Databases

Pamela L. Weeks
Bureau of Labor Statistics
Washington, D.C. 20212

Michel David, OECD, France
David Hall, Battelle Pacific Northwest Laboratories
Gwendolyn Harllee, Bureau of Labor Statistics
Phyllis Levioff, Chase Econometrics
Inger Nilsson, DATACENTRALEN, Denmark
Lars Nordback, Statistics Sweden
Martin Podehl, Statistics Canada
Helen Poot, Data Resources, Inc.
Esther Schroeder, Lawrence Berkeley Laboratory

1. Introduction

Among statistical database systems, an important and growing segment handles economic data. Many such databases originate in government statistical offices and include data on employment and unemployment, price indexes, wage rates, and other measures of economic performance. Users of these data are an increasingly large and diverse community including government and corporate planners and policy makers, researchers, and journalists. Those who participated in the Working Group on Time Series and Large Econometric Databases are either producers, suppliers, or users of economic data. Representatives were included from government and other public statistical offices and a number of private corporations which market databases with their own supporting software.

An econometric database is a collection of economic statistics from which data items can be retrieved for display or for econometric analysis. Most of the data in these databases are in the form of time series. A typical large econometric database contains from hundreds of thousands to millions of time series.

A time series is a series of data items or observations that have been collected at regular intervals. Only one type of data is stored in a particular series. As examples, a time series could contain the national consumer price index as calculated for each month of the last ten years. Another might contain the consumer price index for food as calculated for New

York during each month of the last five years. Thus, each data item can be identified by its date (e.g. month and year) and its type (e.g. New York CPI for food).

Econometric analysis often requires a comparison of different types of data within or between time periods. Some examples of actual questions leading to this type of analysis are:

"What is the relationship between inflation rates and unemployment rates?" "How does the recent recovery from recession compare with other recovery periods?" "Are high tech industries among the fastest growing industries?"

Most of the data in a large econometric database is produced by governmental or other public agencies. An agency that produces data may maintain a database containing principally its own data. In some cases, the public is allowed direct access to the database. In others, the data is exported to secondary sources, often commercial firms, which specialize in building large and varied collections of data acquired from many different sources, and marketing access to the data along with software tools for display and analysis. The degree to which the original data suppliers allow direct public access to their in-house databases varies greatly.

An important aspect of these databases is that they are often accessed by people who are far removed from the data producers. This, combined with the large size of the databases, causes special problems for users in finding the data they want and in getting the right amount of explanatory information about the data.

2. New or Continued Research

The Working Group proposed the following questions for new or continued research:

(1) How should the producers and distributors of statistical data document the data? More specifically, how much documentation should be provided and at what levels?

Time series present special problems in this area, because an important reason for keeping a time series is to have periodic "readings" on a particular type of data. Unfortunately, observations in time series do not always have a consistent basis. In fact, because they report information over a large time span, they are subject to change through revision and redefinition and, thus, may contain gaps or discontinuities.

Many time series contain statistical estimates based on observation of a sample of the real world rather than an exact count of all real world instances. For example, when the newspaper reports that the unemployment rate is 9%, we are actually being told that, based on a survey of a sample of all households, the unemployment rate is <u>estimated</u> to be 9%. If all households could be surveyed, as is done in a census, for example, we could get an actual count of unemployed.

Sometimes the estimates are revised based on information that becomes available at a later time. A survey may accept information that comes in after the initial deadline for reporting estimates, then revise its estimates a month later. Thus a database user who is looking at a statistical series from month to month could find that some numbers have changed and must somehow be able to find out the reason for the change.

A different situation arises if the definition of a data item changes. In the case of the unemployment rate, the definition of unemployment might change so that the new rates are not directly comparable with the old ones. For example, in January 1967 the lower age limit for unemployment statistics in the U.S. was raised from 14 to 16. The historical series were revised to the new definition wherever possible. Users need to know this.

In some econometric databases, a change in definition is flagged as a discontinuity within a time series; in others, the old series is ended at the point of change and a new series is established beginning at the point in time when the new definition was introduced. In either case, users must be informed of the discontinuity if they intend to display the data or use it for analysis over a time span that includes the break point.

It is also desirable to explain the reason for zero or missing values. Data can be missing because of non-response, suppressed for confidentiality reasons, rounded to zero (as opposed to having a real zero value), and so on.

Some of the information about the above should probably be available at the time series level, while other information would be more useful if attached to individual data items. In either case, there is the problem of whether to provide the information online or in printed documentation and at what levels of detail. Different levels of detail will be appropriate for different types of users and for different uses of the data. Some data distributors give users the option of "turning down the volume" on the level of detail of documentation on the terminal screen.

(2) What impact will microcomputers have on the way large econometric databases are offered to the public?

Much of the data that is included in these databases has been published at one time or another but has been put into the database for easier access and to facilitate analysis. The access is generally provided through terminals connected to large mainframe computers via telephone lines. All data retrieval and analysis is done on the central computer. The recent rise in microcomputer usage offers new ways for the data to be distributed and analyzed. For example, users may wish to "download" (transfer electronically) subsets of the data from the central database to microcomputers and use microcomputer software for analysis. Some data suppliers are already providing facilities for users to download data in a form that can be used in microcomputer spreadsheet programs. A problem that arises here, is that, with current techniques, most of the documentary information is lost in the downloading process.

Other implications:

Large time sharing services are probably going to start selling software to use on microcomputers with downloaded data rather than providing all of their services solely on the central computer.

Subscription services may offer data on floppy disks on a monthly basis along with the software to work with the data. In this case, we may not be talking about large econometric databases.

Some data producers/suppliers are considering the use of electronic mail as a relatively inexpensive way of distributing data.

Some data producers see a possibility of eventually replacing their paper publications with electronic ones.

(3) Could there be a common language for accessing statistical databases?

Each of the large econometric databases represented in the working group has its own unique user interface. Thus, a person who needs data from more than one database must be able to cope with each of these interfaces in order to retrieve the data he or she wants to examine. Perhaps it would be possible to provide basic facilities in a "universal" language that could be used for any of these databases regardless of organization or country. One problem here is that the commercial data suppliers who have invested much effort in developing proprietary interfaces to their databases may not see an advantage to making it easier for their customers to get data from other sources. On the other hand, such an arrangement might actually increase each supplier's customer base by making it easier for people to use multiple data sources -- which they would be less inclined to do if it involved learning a variety of systems.

(4) Is it desirable to incorporate into the databases mechanisms to keep track of how frequently time series are accessed; could the information be used to automatically decide (a) whether series that are frequently derived from other series should be stored rather than recomputed each time and (b) whether infrequently used or unused series should be archived? In order to keep track of usage in these huge databases, automated logging procedures would be required. Currently, logs which record usage by time series are the exception, not the rule.

3. Conclusion

Much of the Working Group discussion was directed toward the question of how to improve user access to the data while giving users the explanatory information necessary for correct use of the data. These issues increase in importance as the databases grow and as their uses grow in decision and policy making.

Special Data Types and Operators for Statistical Data

J. E. Gentle
IMSL, Inc.
Houston, Texas 77036

and

Jean Bell
Mathematics Department
Colorado School of Mines
Golden, Colorado 80401

G. Barsottini, Commission of the European Communities
V. Brown, Bell Laboratories
H. Farsi, University of Alberta
W. Nicholson, Pacific Northwest Laboratories
F. Olken, Lawrence Berkeley Laboratory
Z. M. Ozsoyoglu, Cleveland State University
D. Swartwout, Bell Laboratories
R. Zak, University of Manitoba

1. Introduction

Existing DBMSs are generally designed for commercial applications such as employee or customer record keeping. These applications involve frequent retrieval and updating. The DBMS generally supports concurrent access, though at the expense of a relatively large overhead. Furthermore, the DBMS can handle fairly complex data sets, as long as they fit the relational model. On the one hand this power and generality of a DBMS that accommodates the more complicated operations is often viewed as overkill. On the other hand, however, relatively simple operations that the statistician may wish to perform are not provided by the DBMS. For these and other reasons [Bra82, Coh82, Sho82] commercial DBMSs are not widely used to manage data for statistical analysis.

In most statistical analyses the data are organized into flat files, not only in the computer but also in the analyst's mind. Reports generated also take this form. For this reason, many of the statistical analysis systems accommodate only this form of organization.

Operators must exist at the appropriate level of abstraction for the user so as to spare the user from having to program operators in an unnatural manner. For example, the user could

use a SAMPLE operator to extract a sample from a database, rather than go through a long procedural description of sampling.

Special operators are also useful to enforce the logical rules of the operation. For example, operators for combining data sets can enforce conformability requirements and can handle the problems of missing values that may be generated.

Optimization can be realized if operators and types appropriate to the use are provided. For example, if the data are really related as in a matrix, it wastes effort for the DBMS to "flatten" the matrix and then have the user "reconstruct" it.

2. Operators and Data Types for Statistical Data

Here we do not attempt completeness, but only to indicate the most commonly required operators and data types. Furthermore, we do not attempt to restrict the discussion to "special" operators and types, since it could be argued that any one of these may find more general uses. Operators seem to fall into three classes: 1) those that would likely not be provided by a general purpose DBMS, 2) those that would likely exist in most DBMSs, but would likely not be efficient enough to support statistical analysis, and 3) others that do not seem appropriate as primitives, but which may be effected through simple programming-language-like features, such as discussed in [Swa83].

Retrieval is the most common operation. There are three kinds of retrieval operations often performed: 1) unconditional, i.e., straightforward, sequential retrieval, 2) conditional, i.e., retrieval based on a selection criterion, and 3) retrieval of a random sample. Of these three kinds of retrieval, only sampling would not be provided in a standard DBMS. The statistician may require various kinds of sampling: simple random sampling, stratified sampling, cluster sampling, probability sampling proportional to some attribute, and subsampling. The sampling may be with or without replacement. It may be with a specified sample size or with a specified sampling fraction. Although if both the population and the sample size are large and the sampling fraction is small, whether the sampling is with or without replacement or whether the sample size and fraction are exact or not is not too important, good online exact sampling algorithms exist (e. g., [Ken80]) and should be used.

Aggregation is another common operation. While a standard DBMS provides some aggregation operators, a system designed for statistical data would need to include such actions as binning, tallying, and computation of some or all of the order statistics. The large amount of summary data is a distinguishing characteristic of statistical data; hence, optimization of aggregation operators is critical.

In addition, various kinds of regrouping operations are

useful, such as sorting, set operations, and relational operations like binary join.

The operators that are used most often must be optimized for efficiency. The sampling operation, for example, should be performed as close as possible to the physical data; whole databases should not be moved between systems or subsystems just to take a sample.

The most useful general data structure for statistical databases is the multidimensional array. Array operations, such as matrix addition, are useful for statistical data management, but are not provided in a standard DBMS. For a large portion of all statistical analyses, lists and case-by-variable matrices are the appropriate structures. An important special case of these structures is the time series, in which the cases represent points in time. Data relationships are important in demographic databases and many of these fit the relational model. The summary table, which can be viewed as an array of trees, is an important special data structure in statistical applications [Ozs83]. Other data structures finding occasional use are maps and functions. These latter are particularly useful in organizing and analyzing spatial data, such as on air pollution. The statistical DBMS must provide for these various data structures, but not at the expense of a complex user interface. The common array structures that are appropriate for most users should be built with very little explicit "data definition" required of the user.

3. User Interface for Statistical DBMS and Analysis System

Although there has been a great deal of discussion of where the "boundary" should be between a data management system and a data analysis system, this consideration is not really relevant to the statistician, i.e. the list of operators that the data analyst sees in the single system includes not only those above, but also correlation operators, regression operators, etc. It is a "single" system in that the user has a reasonably consistent interface and no special interaction with the operating system is required to use both "data management" and "data analysis" operators in the same "program" (analysis session). The important distinctions perhaps lie in the nature of the algorithms: the algorithms for the operators listed above do not involve any considerations of numerical analysis, whereas an algorithm for regression analysis does. There may be some consideration of "efficiency" of the implementation in the discussion of where the boundary between a data management system and a data analysis system, but such a consideration would be so state-of-technology dependent as to be of no real concern. It would be desirable for the data analyst to tell the system to

REGRESS weight ADJUSTED FOR height ON chlor FOR sex=female

The analyst should not be concerned with the fact that FOR implies

a "data management" activity of selection and REGRESS ... ADJUSTED FOR...ON implies "data analysis" activities.

More research is needed to determine the appropriate level of abstraction for special data types for statistical database management. Currently, there is lack of agreement among database researchers and statisticians about specific data types or operators for statistical database management. Distribution of and experimentation with prototype systems are needed. Although several statistical database management systems have been developed and described in the literature, much of the system development work has not reached the data analyst, who might be willing to give the system a trial. Developers and users must investigate the tradeoff between providing so many special types the the DBMS is too complicated and providing so few special types that retrieval of statistical data is too cumbersome. Many of the answers will come from the experience of the users.

References

[Bra82] Bragg, A. Data manipulation languages for statistical databases -- the Statistical Analysis System, Proc. First LBL Workshop on Statistical Database Management, 1982, 147-50.

[Coh82] Cohen, E. and R.A. Hay. Why are commercial database management systems rarely used for research data? Proc. First LBL Workshop on Statistical Database Management, 1982, 132-3.

[Ken80] Kennedy, W. J. and J. E. Gentle. Statistical Computing, Marcel Dekker, New York, N.Y., 1980.

[Ozs83] Ozsoyoglu, Z. M. An extension of relational algebra for summary tables, Proc. Second International Workshop on Statistical Database Management, 1983, 202-211.

[Sho82] Shoshani, A. Statistical databases: Characteristics, problems, and some solutions, Proc. Eighth International Conference on Very Large Data Bases, 1982, 208-22.

[Swa83] Swartwout, Don. How far should a database system go? (to support a statistical one). Proc. Second International Workshop on Statistical Database Management, 1983, 220-222.

Data Models for Statistical Database Applications

Sue M. Dintelman
Department of Human Genetics
University of Utah
50 North Medical Drive
Salt Lake City, Utah 84132

Satki Ghosh, IBM Research Laboratory
Rick Greer, Bell Laboratories
John Long, University of Minnesota
Mauro Maier, IBM Italia S.p.A.
Gultekin Ozsoyoglu, Cleveland State University
Maurizio Rafanelli, National Research Council of Italia
Hideto Sato, Economic Planning Agency (Japan)
Arie Shoshani, Lawrence Berkeley Laboratory
Stanley Su, University of Florida
Robert Wiederkehr, King Research, Inc.

1. Introduction

The topic of data models for statistical database management systems and the related topics of metadata and data transformation were discussed in light of the special needs of statistical database users. A data model provides a formal means of representing information and a formal means of manipulating such a representation. A data model to be used with statistical applications should provide object types and operators that correspond to the elements of statistical analysis. The following section outlines some of these elements and section 3 describes briefly some proposed data models. More complete presentations of the special characteristics of statistical databases may be found in [Bat82], [Lit83], [Kre82] and [Sho82].

2. Requirements of statistical applications

A commonly expressed view of statistical analysts is that they need only tables (i.e., flat files) as a modeling construct. However, observing even a simple statistical analysis where a table is assembled from information from more than one existing table, then transformed by simple or complex functions into a different table which is then used to produce a variety of summary tables, etc., leads to the conclusion that more complex constructs are necessary.

For raw data, i.e., the original observations or unit records, the same types of features provided by traditional data models are useful. One example of this is the ability to relate entities

to one another, such as individuals to their spouses, children, grandchildren, etc. The flat files usually provided as the only object type in statistical systems are not sufficient for this. It is necessary for the model to represent hierarchical and network relationships so that statistical analysis involving multiple data sets can be specified.

In addition, an object which is an individual from one point of view may denote a whole class of individuals from another perspective. Objects that appear in one view as attribute values may appear as entities or relationships or metadata in another view. For example, the units of a measurement which may be metadata in one view may be considered part of the basic data in another. This relativism implies that the data model should allow data objects to be multiply labeled as required by different applications.

A data model for statistical applications should allow complex data types such as matrices, time series, sets, vectors, variable length text strings, dates, etc. as primitives. The inclusion of these commonly used constructs as primitives greatly simplifies the task of defining and using a database.

The operations provided by a data model for use with statistical applications should provide the usual database management functions such as retrieval, update, insertion and deletion. Statistical databases are often not being actively updated, but there is still the need to delete certain cases or modify values which have been found to be incorrect. In other words, the same functionality is required even though the pattern of usage of these operators may be different for commercial and statistical applications.

Routine statistical analysis involves a large variety of often complex operations, some of which may be being developed. It should be possible to add new operations for object types. Associated with the use of aggregation (summary) operations is the problem of specifying groupings. In current systems it is often difficult to specify the groupings to use to perform the aggregation (i.e. by income group, by geographic area, by family size, etc.). The ability to specify category and summary attributes, where a summary attribute is one which contains the measured data on which statistical summaries may be done and a category attribute describes the measured data, would facilitate the specification of aggregations.

It is the need to represent and manipulate derived data in which statistical database applications differ most from traditional commercial database applications. [Gey81] illustrates some of the difficulties that exist when trying to manage summary data with standard data models. Because summary data files are inherently redundant, i.e., the same data appears in multiple places in the data file for different levels of counting, aggregate functions found in most existing systems will produce inaccurate

results. This is because they make the implicit assumption that the data they manipulate are non-redundant individual case data.

Many statistical analysis procedures have results that differ in size and shape from the input data. A linear regression requires a matrix of independent variables and a vector of dependent variables and produces vectors of coefficients, residuals, and leverage values, a covariance matrix and a scalar residual sum of squares.

In addition to the problems of representing and manipulating derived data, there is a need to have some way of preserving the actual derivation of results.

3. Specific Approaches

There are many proposals for data models to be used with statistical database applications (see for example [Cha81], [Gre83], [Kre82], [Mai83], [Ozs83], [Raf83], [Shi83] and [Su83].) These represent a wide range of approaches for including the components necessary to model the statistical database application.

Two approaches to providing primitive object types are taken in these models. One is to provide a few basic primitives that allow a great deal of flexibility in creating special objects and operations. The other approach is to provide the special objects and operations as primitives in the system. One problem mentioned regarding the second approach is that by increasing the number of constructs, the model becomes more difficult for the user to learn. The other side of this argument is that specialized primitives can increase the ease of use of a model for application development.

A data model may be considered as a rather abstract programming language and the two approaches for data models described above find analogies in programming languages in the late 60's and early 70's [Ghe82]. There are programming languages containing many constructs which make the language more difficult to learn but, once learned, make it easier to use to develop applications (e.g. PL/1). The other involves the support of a few primitive constructs with which users may build their own application specific constructs (e.g. PASCAL). In any case experimentation with systems that implement the proposed models will help determine which constructs are actually useful and practical.

The ALDS project [Tho83] and the work described in [Wie82] are examples of work in the area of modeling the devrivation of results. The ALDS project is developing a model of the data analysis process and facilities that will allow users to maintain, save or restore all of the components required to completely describe or reconstruct a data analysis environment.

References

[Bat82] D. Bates, H. Boral, D.J. DeWitt. "A Framework for Research in Database Management for Statistical Analysis." Proc. ACM SIGMOD International Conference on Management of Data, Orlando, Florida, June 1982, 69-78.

[Cha81] P. Chan, A. Shoshani. "Subject: A Directory driven System for Organizing and Accessing Large Statistical Databases." Proc. of the International Conference on Very Large Data Base (VLDB), 1981, 553-563.

[Gey81] Fredric Gey, "Data Definition for Statistical Summary Data or Appearances Can Be Deceiving", Proceedings of the First LBL Workshop on Statistical Database Mangement, 1981, 3-18.

[Ghe82] C. Ghezzi, M. Jazayeri, Programming Language Concepts, John Wiley and Sons, Inc., New York, 1982

[Gre83] Greer, R. "An Introduction to Sampling to Estimate Database Integrity." In Proc. of the Second International Workshop on Statistical Database Management, Los Altos, California, September 1983, pp. 360-367.

[Kre82] P. Kreps, "A Semantic Core Model for Statistical and Scientific Databases." A LBL Perspective on Statistical Database Management, Berkleley, California: University of California, Lawrence Berkeley Laboratory, 1982, 129-157.

[Lit83] R.J. Littlefield, P.J. Cowley. "Some Statistical Data Base Requirements for the Analysis of Large Data Sets." Computer Science and Statistics: Proc. of the Fifteenth Symposium on the Interface, Houston, Texas, March 1983, 24-30.

[Mai83] M. Maier, C. Cirilli. "SYSTEM/K: A Knowledge Base Management System." Proc. of the Second International Workshop on Statistical Database Management, Los Altos, California, September 1983, 287-294.

[Ozs83] G. Ozsoyoglu, Z.M. Ozsoyoglu. "Features of a System for Statistical Databases." Proc. of the Second International Workshop on Statistical Database Management, Los Altos, California, September 1983, 9-18.

[Raf83] M. Rafanelli, F.L. Ricci. "Proposal of a Logical Mode for Statistical Data Base." Proc. of the Second International Workshop on Statistical Database Management, Los Altos, California, September 1983, 264-272.

[Shi83] K. Shibano, H. Sato. "Statistical Database Research

Project in Japan and the CAS SDB Project." Proc. of the Second International Workshop on Statistical Database Management, Los Altos, California, September 1983, 325-330.

[Sho82] A. Shoshani, "Statistical Databases: Characteristics, Problems, and Some Solutions." Proc. 1982 International Conference on Very Large Data Base, Mexico, City, Mexico, September 1982, 208-222.

[Su83] S.Y.W. Su, "SAM*: A Semantic Association Model for Corporate and Scientific-Statistical Databases", Information Sciences, 29 (1983), 151-199.

[Tho83] J.J. Thomas, D.L. Hall. "ALDS Project: Motivation, Statistical Database Management Issues." Proc. of the Second International Workshop on Statistical Database Management, Los Altos, California, Sept. 1983, 82-88.

[Wie82] R.R.V. Wiederkehr, "Methodology for Representing Data Element Tracings and Transformations in a Numeric Data System." Special Issue on Numeric Databases, Drexel Library Quarterly, Summer-Fall, 1982, 18 (3 & 4) 161-176.

Metadata Management

Robert T. Lundy
DIALOG Information Services, Inc.
3460 Hillview Avenue
Palo Alto, California 94304

Y.M. Bishop, U. S. Department of Energy
S.R. Childs, Data Resources, Inc.
P.J. Cowley, Pacific Northwest Laboratory
R. Cubitt, Statistical Office of the European Communities
J. Dolby, Dolby Associates
S. Ghosh, IBM Research Laboratory
R. Hay, Northwestern University
H. Holmes, Lawrence Berkeley Laboratory
R. Lundy, DIALOG Information Services, Inc.

1. Introduction

The principal focus of this group was the management of Metadata, which may be loosely defined as data about the data rather than the data itself.

The term 'metadata' has a multiplicity of definitions, descriptions, and uses. For this reason, it is clear that most of the people who have dealt with the problem of metadata management have been strongly influenced by the problems immediately confronting them. Part of the difficulty arises because it is not always clear in any particular context where the 'data' stops and the 'Metadata' begins. For example, in a relational database describing a group of people, the attribute 'Sex' is part of the metadata of the file (it describes something about the contents of the file) and the values that it may take ('male' or 'female') may likewise be looked upon as metadata in that it further delineates the attribute. The values of the attribute in each case may be looked upon as the real data itself from some perspectives. If the database consists of records about individuals, and the 'Sex' attribute is given for each case, then the 'male' or 'female' code might be regarded as part of the data rather than the metadata. Others argue that the real 'data' item is the existence of the case with the indicated sex, and that in consequence the value of 'Sex' in each case record is metadata there as well.

The group agreed that the definition of Metadata could best be approached from two perspectives:

* Functional (based on usage), and

* Operational (based on the objects actually used and stored).

2. A Functional Definition of Metadata

Metadata is used for the following purposes:

* Storage and Retrieval of data

  For example, if one wishes to analyse the census tracts of all coastal counties in California, the metadata concerning county names and geographic levels must be examined in order to select the data items of interest, as well as the definitions of the variables to be examined as data (e.g. Population counts).

* Presentation and Display of data

  This category might include the documentation and display of a table (or a composite table derived from several primary data tables). The relevant metadata for this purpose would include notes concerning the sample universe, variable definitions, case names, and other title and footnote information including notes indicating the source of the data.

* Analysis of data

  This category includes any calculations that might be performed on the data, such as aggregation, comparison, various transformations, the derivation of correlation coefficients, etc..

* Physical Description of the data file

  This covers such areas as the physical medium on which the data is stored (e.g. tape or cards), the representation of the data (character or internal binary), relative location of the fields containing the various data items, and so forth. As this is one of the few instances where the use, importance, and terminology are already of necessity reasonably well defined, the working group did not address this aspect of metadata.

3. Metadata as Operationally Defined

The other approach to defining metadata is to look at those kinds of information most commonly included in various database manipulation systems (or recommended for inclusion). This is the operational approach to definitions.

A brief list of those differentiable items that were agreed by the working group to be examples of metadata constructs is as follows:

1. Short Name - usually a 1-8 character mnemonic based on the Long Name, e.g. 'FEM20_49'.

2. Long Name - usually a 20-80 character phrase explaining the data item, e.g. 'Females between 20 and 50 years of Age'.

3. Definition or Computational Formula

   This usually consists of a sentence or at most a paragraph describing the source of the data or the manner in which it was computed, e.g. 'Derived from a sample of returns from the U.S. Census of 1980', or 'Calculated as BIRTHS / FEM20_49'.

4. Title

5. Column Headers (and Generic header above those)

6. Row labels (and generic header for these)

7. Footnotes

8. Keywords (usually from an associated thesaurus)

9. Data Descriptors

   * Units of Measure
   * Scale
   * Missing value codes
   * Data Quality codes
   * Universe (Description)
   * Allowed or defined data values permitted by the Universe.
   * Data Type (integer, real, character-code, etc.)

10. Default Editing & Display Specifications

11. Summary Statistics

12. Abstracts, textual description & history

The above list is by no means exhaustive. Some systems were brought up in the course of the discussion in which over 700 different possible items of metada-

ta are defined. Note, however, that not all of the items are applicable in general to all imaginable statistical databases.

4. What Should Be Done with metadata

The use of metadata for the purposes of enhancing storage and retrieval capabilities, as well as its use in the construction of displays, is not an area of controversy. Most systems handle these functions adequately. Metadata management, however, is not as well defined, either in principle or practice.

Just as data can be derived from preexisting data (e.g. base tables), so too should metadata for derived data be generated from pre-existing metadata (e.g. base table metadata). Unfortunately, generating metadata is a difficult task. For example, suppose we have two data items A and B in our database. If we create a third variable C, as

$$C = F(A,B)$$

then we would like the software to define the metadata for variable C in an automatic fashion, such that

$$M(C) = G(M(A),M(B))$$

where M(x) represents the metadata for variable x, and G is a transformation function for metadata associated with the data transformation function F. The nature of M for any F is not always clear. For example, if F is simple addition, so that C = A + B, then the various metadata transformations might be arranged as follows:

* Short Name - Taken from the expression entered by the user.
* Long Name - Taken from the functional definition and the short names of component variables. (With optional user override).
* Definition or Computational Formula - Similar to Long Name.
* Title - Similar to Long Name
* Column Header(s) - Taken from Short Name(s)
* Row Labels - Should be identical for A and B, hence C.
* Footnotes - Footnotes from A and B should be Footnotes for C also.
* Keywords - Logical OR of keywords associated with A and B.
* Data Descriptors - These can be derived in a fairly straightforward fashion.

* Editing specifications - Also mechanically derivable.

* Summary Statistics - Can be derived directly.

* Abstract - The system should create an automatic note referring to the operations performed and directing the user's attention to the text associated with the original variables.

Furthermore, it was generally agreed that the system should at the very least warn the user of attempts to perform possibly incorrect operations, such as the aggregation of quantities that are incompatible, such as gallons and meters.

The generation of meaningful titles, labels, and footnotes in the general case is not simple. Dolby([Dol83]) recommends the use of a faceted classification scheme for data which seems to work well when the data units are tables, but which becomes unwieldy when data come in the form of scalar items. We conjecture that in order to do an adequate job in handling titles and footnotes, the system may require the use of techniques commonly associated with Expert Systems.

5. Existing Statistical metadata Management Schemes

Most systems for manipulating statistical data include some provision for handling some form of metadata, often in the form of a data dictionary, schema, or other construct. However both the items of metadata handled and the degree to which the handling is automated and propagated through as the database is modified differ widely between systems.

In general, systems that are heavily oriented towards computation seem to be less capable in the area of metadata management than are systems that are oriented more heavily towards data management. Some systems attempt to use the strengths inherent in the two contrasting types of packages by attempting to combine them in a unified environment. For example, the SAS system, which has few metadata management capabilities, has the ability to take data by means of interfacing procedures from files maintained by more metadata-oriented packages such as OSIRIS or DATATEXT. This approach is only partially successful, however, since changes made to the database once it is transferred out of the control of the metadata-oriented packages cannot be reflected in the database as they would have been had the alterations been made in the original database.

Database systems that are oriented primarily towards textual data have generally better metadata capabilities, although they usually compensate by having little in the way of analytical or computational capabilities.

No widely available system for the organization and manipulation of statistical databases has comprehensive and adequate facilities for handling metadata.

6. Recommendations and Conclusions

* We recommend that a formalized set of metadata items be promoted as a standard minimum that all Statistical Database management or analysis packages should handle in a consistent fashion, regardless of the design and purpose of the database.

* The metadata management scheme should be logically independent of the management scheme used for the database as a whole. The metadata definition language proposed by McCarthy([McC82]) is suggested as a reasonable starting point for standardizing such an effort.

* Standard definitions of metadata operations associated with various data operations should be developed.

References

[Dol83] Dolby, James L., and Nancy Clark, The Language of Data (unpub. draft)

[McC82] McCarthy, John L., "Metadata Management for Large Statistical Databases", in Proceedings of the Eighth International Conference of Very Large Data Bases, Mexico City, Mexico, Sept. 8-10, 1982, pp. 234-243

Physical Storage and Implementation Issues

D.S. Batory
Department of Computer Sciences
The University of Texas at Austin
Austin, Texas 78712

M.A. Bassiouni, Univ. of Central Florida
J. Dixie, Office of Population Censuses and Surveys, U.K.
F. Gey, Lawrence Berkeley Laboratory
W. Kim, IBM Research
K.A. Hazboun, Penn State University
K. Shibano, IBM Japan
P. Svensson, Swedish National Defense Research Institute
H. Wong, Lawrence Berkeley Laboratory

1. Introduction

The present state of statistical database software reflects a varied integration of database management and numerical analysis technology. Popular statistical analysis packages, such as SAS ([SAS79]) and SPSS ([Nie75]), are based on rudimentary file management systems. Facilities for backup, recovery, data definition, concurrency control, and processing high level (nonprocedural) queries are limited or nonexistent. At the other extreme, some DBMSs have been designed specifically for statistical databases. Examples are RAPID ([Tur79]), ALDS ([Bur81]), and SEEDIS ([McC82]). However, these systems must interface to existing statistical packages for the numerical processing of data. An intermediate approach, one that is gaining popularity, is to extend the data management and interface capabilities of existing DBMSs to support special-purpose applications. Extensions to System R ([Has82]) and INGRES ([Sto83]) to handle complex objects and abstract data types are recent examples.

In this article, we will examine some of the implementation issues of statistical databases from the perspective of database management. Specifically, we cite limitations that are common to many commercial DBMSs (limitations which, we feel, make most DBMSs unsuitable for statistical databases), and identify essential features that a data management component of a statistical DBMS (SDBMS) must support. We also have recommendations for future research in data compression, a significant component in SDBMS implementations.

2. Data Management Features of an SDBMS

An SDBMS consists of two components: a statistical analysis (SA) component and a data management (DM) component. The DM handles the storage and retrieval of data. It also supports some elementary statistical operations and is responsible for formatting retrieved data for further processing by the SA. The SA handles complex statistical analyses and relies entirely on the DM for data storage and access. This separation is a practical one, for it coincides nicely with the domains of numerical analysis and database management. It reinforces the preferred separate (if not coordinated) development of mathematical and database software.

The boundaries between the SA and DM are rather fuzzy. For performance reasons, it seems best to define elementary (and frequently requested) statistical operations as primitives in the DM, although conceptually these operations might otherwise belong to the SA. Examples are random and clustered sampling, and the simple aggregation functions that are common to most commercial query languages, such as maximum, minimum, average, count, and cross tabulation ([Dat82]).

Even if a conceptually clean separation were possible, it is evident that the DM components of existing commercial DBMSs are inadequate to satisfy the requirements of common statistical database applications. Among the more important limitations are:

Size. Statistical records often have hundreds and sometimes thousands of attributes. Individual records can be tens of thousands of bytes long and files can have hundreds of millions of records ([McC82]). Commercial DBMSs cannot handle the dimensions of some statistical databases.

Storage structures and compression algorithms. Commercial DBMSs support a set of file storage structures and data compression algorithms that work well for business data processing. However, it is known that other storage structures, such as transposed files ([Hof75], [Tur79], [Bur81], [Mar83]), and compression algorithms, such as index encoding ([Als75], [Bat83]) and the SEEDIS compression algorithm ([Gey83]), are much better suited for statistical files. Thus, for reasons of performance, commercial DBMSs are not well suited for statistical processing.

Data modeling and formatting. Matrices, time-series, and G-relations, among others, have been found useful in expressing the logical organization of statistical data ([Su83]). Existing DBMSs are based on data models that support only simple data types (e.g., scalar, repeating groups), and uncomplicated data relationships (e.g., owner - member). Although a current trend in DBMS design is to enlarge the data structuring - data relationship capabilities of DBMSs (e.g., [Has82]), it is not clear whether the proposed extensions are sufficient to handle the needs of statistical applications. It is clear, however, that the relational model, as described in popular database texts (e.g., [Dat82]), is inadequate for modeling statistical databases. Supporting complex data types is the responsibility of the DM.

An interesting and unsolved research problem is how to support complex data types, especially in connection with file storage structures. It seems possible to extend the idea of transposition (i.e., transposed files) beyond the storage of a relational table of records. However, it is not clear how useful these extensions would be, or whether something other than transposed files would be better. The way to address this particular problem is to determine what complex data types and their attendant operations should be supported. Presently, there is no agreement on what the primitive types should be, let alone what DM operations (as opposed to statistical or mathematical operations) are performed on these types. Until these issues are better resolved, progress on how to support complex data types will be slow.

Data processing characteristics. In addition to the above limitations, there are special characteristics of statistical processing that existing DBMSs do not handle properly. Statistical files, for example, often utilize several types of null values, each of which must be processed in a special way. SAS and SPSS have facilities for handling more than one type of null value, but most commercial DBMSs have no such facilities.

Statistical file processing can be divided into an exploratory phase and a confirmatory phase. During the exploratory phase, file updates may occur. As statistical analyses often process significant portions of a file, concurrency control may be effectively and efficiently managed by using locks that have a large granularity (e.g., file locks or subfile locks rather than record locks). During the confirmatory phase, there are no file updates, so concurrency control is not a problem. Therefore, it may be the case that concurrency control mechanisms for SDBMSs need not be as complex as those found in todays DBMSs. Future research is needed in this area.

3. Research Directions on Data Compression

Unprocessed statistical data, as a general rule, has enormous amounts of redundancy. Seismic monitoring data consisting of long periods of low activity (generating long sequences of small numbers or zeros) ([Egg81]), sparse matrices of financial activities and census databases ([Sho82]), and satellite transmission data ([Loh83]) are examples. Such files are processed sequentially, in whole or in part. The expense and speed of sequential processing is proportional to the file size. Data compression is quite valuable in this connection, for it can not only reduce a file's storage volume, the file's effective processing speed can increase.

Many data compression techniques for statistical (and nonstatistical) databases have been proposed over the last few years. This trend will continue. From a practitioner's viewpoint, it is difficult to know which compression techniques are suitable for particular applications. Because there is no existing methodology or metric by which different compression techniques can be assessed and compared, the value of new techniques is often difficult to judge. Questions like 'under what conditions will a compression technique work well and does a particular set of files and applications satisfy these conditions?' are difficult to answer, but such answers are essential for determining what compression techniques *(if indeed any)* should be used.

Reducing theory to practice is an important and serious problem in database research. We feel that the development of practical methodologies for evaluating data compression techniques will be valuable contributions to both practitioners and researchers alike. Such methodologies should encorporate 'hard' comparison metrics, such as compression results on actual files (rather than simulation results), and should avoid subjective rankings as much as possible. (Dujmovic's Logic Scored Preference (LSP) methodology for ranking systems may be useful here (see [Duj79], [Su84])). A possible result of applying such methodologies would be a catalog of various compression techniques indicating their performance and usability.

Existing SDBMSs, such as RAPID, ALDS, and SEEDIS, normally utilize several data compression techiques. Examples include run-length encoding, index encoding, and zero and blank suppression. As a general rule, an SDBMS applies all of its compression techniques automatically; there are no mechanisms for enabling or disabling the use of any technique in particular. An almost certain result which will come from evaluations of different compression techniques is that no single technique will be universally good in all applications. Future SDBMSs, therefore, should support several different compression techniques which can be optionally selected. This would provide a convenient mechanism for introducing new compression techniques (which is difficult, if not impossible, to do in present SDBMSs) and for optimizing database performance.

A promising area of research is the development of algorithms for manipulating, searching, and processing data in a compressed form. As examples, the Hu-Tucker ([Knu73]) and index encoding algorithms enable sorting and searching operations to be performed directly and efficiently on compressed data. (The key to such algorithms is to assign compressed codes to symbolic values in a way so that the numerical ordering of the codes preserves the lexical ordering of their symbolic counterparts. In this way, operations like sorting, searching, and inequality testing can be performed directly on compressed codes without requiring data expansion). Eggers, Shoshani, and Olken have developed the Header Compression technique that supports fast random and sequential accessing of data elements in a compressed file ([Egg81]). Another promising research direction is the development of algorithms to process compressed matrices. Algorithms to transpose compressed matrices is a good example.

The value of this line of research is that data compression and expansion introduces nontrivial overheads in data processing. By eliminating the cost of expanding data in processing common file operations, a significant gain in performance may result. Such gains will help amortize the overhead associated with data compression and expansion.

Besides algorithm development, there is a growing recognition of the importance of data compression in reducing file transmission costs in a distributed statistical database environment. Research on this promising topic is just beginning ([Haz83]).

References

[Als75] P.A. Alsberg, 'Space and Time Savings Through Large Database Compression and Dynamic Restructuring', *Proc. IEEE* 63, 8 (August 1975), 1114-1122.

[Bat83] D.S. Batory, 'Index Encoding: A Compression Technique for Large Statistical Databases', *Proc. Second International Workshop on Statistical Database Management,* 1983, 306-314.

[Bur81] R.A. Burnett, 'A Self-Describing Data File Structure for Large Data Sets', in *Computer Science and Statistics: Proceedings of the 13th Symposium on the Interface,* Springer-Verlag, New York, 1981, 359-362.

[Dat82] C.J. Date, *An Introduction to Database Systems,* (Third Edition), Addison Wesley, Reading, Mass., 1982.

[Duj79] J.J. Dujmovic, 'Criteria for Computer Performance Evaluation', *Performance Evaluation Review* 8 #3 (1979), 259-267.

[Egg81] S. Eggers , F. Olken, and A. Shoshani, 'A Compression Technique for Large Statistical Databases', *Proc. VLDB 1981,* 424-434.

[Gey83] F. Gey, J.L. McCarthy, D. Merrill, and H. Holmes, 'Computer-Independent Data Compression for Large Statistical Databases', *Proc. Second International Workshop on Statistical Database Management,* 1983, 296-305.

[Has82] R. Haskin and R. Lorie, 'On Extending the Functions of a Relational Database System', *Proc. ACM SIGMOD 1982,* 207-212.

[Haz83] K.A. Hazboun and J.L. Raymond, 'A Multi-Tree Automaton for Efficient Data Transmission', *Proc. Second International Workshop on Statistical Database Management,* 1983, 54-63.

[Hof75] J.A. Hoffer, 'A Clustering Approach to the Generation of Subfiles for the Design of a Computer Data Base', Ph.D. Dissertation, Dept. of Operations Research, Cornell Univ., Ithaca, New York, 1975.

[Knu73] D.E. Knuth, *The Art of Computer Programming, Vol. 3: Sorting and Searching,* Addison Wesley, Reading, Mass., 1973.

[Loh83] G.M. Lohman, et al., 'Remotely Sensed Geophysical Databases: Experience and Implications for Generalized DBMS', *Proc. ACM SIGMOD 1983,* 146-160.

[Mar83] S.T. March, 'Techniques for Structuring Database Records', *ACM Comp. Surveys,* 15, 1 (March 1983), 45-80.

[McC82] J.L. McCarthy, et al., 'The SEEDIS Project: A Summary Overview of the Social, Economic, Environmental, Demographic Information System', Computer Science and Mathematics Dept., Lawrence Berkeley Laboratory, University of California, Berkeley, Calif., 1982.

[Nie75] N. Nie, et al., *SPSS - Statistics Package for the Social Sciences,* McGraw-Hill, 1975.

[SAS79] *SAS User's Guide,* SAS Institute, Inc., Cary, North Carolina, 1979.

[Sho82] A. Shoshani, 'Statistical Databases: Characteristics, Problems, and Some Solutions', *Proc. VLDB 1982,* VLDB Endowment, Saratoga, Calif., 208-227.

[Sto83] M. Stonebraker, B Rubenstein, and A. Guttman, 'Application of Abstract Data Types and Abstract Indices to CAD Data Bases', Report UCB/ERL M83/3, Electronics Research Laboratory, University of California, Berkeley, 1983.

[Su83] S.Y.W. Su, 'SAM*: A Semantic Association Model for Corporate and Scientific-Statistical Databases', *Information Sciences,* 29 (1983), 151-199.

[Su84] S.Y.W. Su, et al., 'A Cost-Benefit Decision Model: Analysis, Comparison, and Selection of Data Management Systems', to appear.

[Tur79] M.J. Turner, R. Hammond, and P. Cotton, 'A DBMS for Large Statistical Data Bases', *Proc. VLDB 1979,* 319-327.