

PROCESSING DATA MODEL ABSTRACTIONS

W. Terry Hardgrave
Donald R. Deutsch
National Bureau of Standards

INTRODUCTION

A number of data abstraction, database, and conceptual modeling issues border on the three technical areas that this workshop addresses: artificial intelligence, database management, and programming languages. We assume that the workshop is dedicated to removing the boundaries between these three fields in order to shape research and development of common interest. While enthusiastically supporting this objective, we believe that each area has a unique perspective and emphasis worth preserving. The three fields can truly benefit from each other only if each one appreciates the diversity of these research viewpoints. In this spirit we describe a research project on Abstract Data Models conducted by the Institute for Computer Sciences and Technology (ICST) of the National Bureau of Standards (NBS). Before doing so, however, we wish to discuss from our perspective both some problems that database management shares with the areas of artificial intelligence and programming languages, and some possible mathematical solutions to these problems.

SHARED RESEARCH PROBLEMS

In many areas of artificial intelligence research a large amount of data is needed to support intelligence. Resulting AI systems could benefit from the support of database management facilities. But for years the primary tool for AI research has been LISP, a tool with only limited data storage and retrieval capabilities. DBMS research, on the other hand, has produced powerful query answering systems based on the predicate calculus. AI research could utilize these advances in query languages if the AI target was some form of predicate calculus; whatever an AI system accepts as input, its output could be predicate calculus.

Programming language research generally ignores the fact that data often outlives programs. Database management, on the other hand, is predicated on the belief that one should not try to describe and control data within programs whose execution time is short relative to the lifespan of databases they reference. A data description stored apart from programs that manipulate it is a DBMS concept that should be recognized by more research efforts in programming languages. One initial step toward acknowledging the long life of data is PASCAL/R [SCHM77], which writes out symbol table information for use as a stored data-description.

A second look at these areas where database management overlaps artificial intelligence and programming languages results in an awareness of problems shared by DBMS and the other fields. Both database management and AI need a common precise formalism. The DBMS needs for formal precision are being at least partially satisfied by mathematically based solutions such as the predicate calculus; database and artificial intelligence researchers should work together to develop this and/or other formalisms that meet their common needs. The difference between database management and programming languages is primarily one of emphasis. Programming languages focus on operations and procedures, while database management is viewed by many as the study of mass-storage data structures. Neither view is entirely correct; structures and their associated operations are inseparable.

SEEKING SOLUTIONS IN MATHEMATICS

Database management shares with the disciplines of artificial intelligence and programming languages the need for a precise formalism and the need to recognize that data structures and operations on those structures are equally important. Research at NBS/ICST is looking to mathematics for solutions to both of these problems. Mathematicians commonly define new terms and concepts using previously defined terms or long-standing and generally accepted concepts. Computer scientists, however, often assign terms to concepts without concern for previous definitions or theoretical foundations. Indeed, the research literature on database management often uses long-standing mathematical terms and notations incorrectly.

Notwithstanding deficiencies in its application to date, mathematics promises a level of precision that is needed to promote understanding among professionals in related areas of computer science. The NBS/ICST Abstract Data Models research, for example, is addressing the problem of representing both structures and operations on structures in a formalism soundly based in mathematics.

DATA MODEL PROCESSING

NBS/ICST is charged with developing Federal standards for database systems, and with pursuing basic research in support of Government standardization objectives. Since database systems can be viewed as implementations of data models, ICST is interested in developing a structured approach to the study of data models. One objective of the Abstract Data Models project is to design and build a prototype of a Data Model Processor (DMP). Many of the ideas for the DMP effort come from Rothnie and Hardgrave [ROTH76]; in particular, the DMP accepts a semi-tabular, parametric description of a data model and emulates the operation of an implementation of the model.

Descriptions of data models require precise formalisms to represent the syntax for DBMS interface languages as well as the semantics for operations. Positional Set Notation (PSN) is our current candidate for expressing DBMS semantics. A data model description must include a high-level, set-theoretic specification of the algorithm for executing each operation. The data model processor further provides a capability for defining query languages by translating phrases of query syntax into invocations of the primitive operations of the data model. Eventually, a capability for defining mappings either within a single data model or across data models may be provided.

Previous research resulted in prototype implementations, and performance measurement and prediction tools for set processor software systems necessary to support development of a DMP [DEUT79]. We are developing second generation versions of these tools. The DMP will be built on top of the resulting improved set-theoretic software. We have completed a preliminary user's manual for the DMP and developed initial specifications for the CODASYL, TDMS and relational models. All of our current implementation efforts are on UNIX; these newest set processors have been designed as extensions of the UNIX command language.

REFERENCES

- DEUT79 Deutsch, D. R., Modeling and Measurement Techniques for Evaluation of Design Alternatives in the Implementation of Database Management Software, NBS Special Publication 500-49, July, 1979, 244pp.
- ROTH76 Rothnie, J. B., and Hardgrave, W. T., "Data Model Theory: A Beginning", Proceedings of Fifth Texas Conference on Computing Systems, October 1976. (Due to mail delay, only an abstract appears; paper available from authors).
- SCHM77 Schmidt, J. W., "Some High Level Language Constructs for Data of Type Relation", TODS, Vol. 2, No. 3, Sept. 1977, pp. 247-261.