

On Conceptual Database Modelling

Dennis McLeod

University of Southern California
Computer Science Department
Los Angeles, CA 90007

This research was supported, in part, by the Joint Services Electronics Program through the Air Force Office of Scientific Research under contract F44620-76-C-0061.

1. Introduction

The importance of *conceptual modelling*, representing real world objects and events in a computerized database, is recognized in a number of active research areas. In particular, the work that is most relevant to the focus of the workshop is that concerning *conceptual/semantic database models* (in the database systems research domain), *abstract datatypes* and associated specification techniques (in the programming languages area), and *knowledge representation* (in artificial intelligence research). The goals, methodology followed, and results obtained by researchers in these three research domains differ in a number of fundamental ways; perhaps the primary driving force that differentiates these three kinds of approaches is the intended use/application of the conceptual modelling facilities.

This brief paper describes our current research in the areas of conceptual database modelling, database design, and the architecture of database systems. The goals and intended uses of work are described, with emphasis on our specific approach to data abstraction and conceptual modelling. In particular, our current research concerns the design, use, and evolution of computerized database systems and database-intensive information systems.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
© 1980 ACM 0-89791-031-1/80/0600-0161 \$00.75

2. Semantic Database Models and Logical Database Design

At present, the process of designing the logical organization of a database is largely ad hoc, and requires significant database design expertise. The goal of this project is to devise principles, techniques, and tools to support the structured design of databases and the operations that will be applied to them. In particular, this project focuses on the following specific tasks:

- A *semantic database model* is being developed, which is a user-oriented formalism to support the specification of the logical structure of a database. Such a model can be applied to generate a meaning-based description of the content and logical structure of a database. Based on our recent experience in the design and application of semantic database models^{1,2,3}, we are developing the *event database model*. The event database model includes specific mechanisms to model both the objects and events that are relevant in an application environment.
- To accompany the semantic database model, an associated structured, disciplined design methodology is being developed, the purpose of which is to guide designer(s) (experts in the application environment, but not database experts) in specifying and modifying the logical organization of a database (its *schema*). The *event-based database design methodology* and the event database model are very closely intertwined; together they attempt to not only capture the important modelling capabilities that a semantic model must possess, but also to package them in a manner that facilitates the design of semantically expressive and easily evolvable databases.
- An interactive *event-based database design aid* will be designed and implemented, based on the event-based design methodology and database model. The purpose of this design tool is to assist

¹"Applying a Semantic Database Model", with R. King; *Proceedings of International Conference on the Entity-Relationship Approach to Systems Analysis and Design*, Los Angeles CA, 10-12 December 1979.

²"Database Definition with SDM: A Semantic Database Model", with M. Hammer; to appear in *ACM Transactions on Database Systems*, 1980.

³"Abstraction in Databases", with J. Smith; to appear in *Proceedings of Workshop on Data Abstraction, Databases, and Conceptual Modelling*, Pingree Park CO, 23-26 June 1980.

a database designer/maintainer in specifying and modifying the logical organization of a database, including the types of data in the database, the structural interconnections among data items, and the operations that can be performed on the data. The interactive tool will be based on a stepwise, disciplined approach⁴.

In sum, this project focuses on supporting the design and evolution of logical data structures and operations in computerized database systems. The result of applying the event-based design methodology (and model) is an implementation-independent, user-oriented database specification. Such a specification can be used as a communication and documentation aid for both pre-existing and new database systems. As discussed in detail below, such a specification can also be mapped into physical storage structures and access methods to result in a practical database system; alternatively, a logical database specification can be mapped into more conventional database structures, such as flat files (or relations) if a database management system supporting these lower-level structures is to be used to support the database; finally, a general-purpose database management system that supports the event database model at its interface could be developed.

3. Database Federation

One of the most significant trends and important challenges of the next decade in the area of computerized database systems is to effectively support decentralized collections of data. Recent work on "distributed databases" has focused on techniques to store and access data at various nodes in a computer network, as if that data were part of a single logical database; these efforts therefore concern the physical distribution of data. We believe that logical decentralization is an equally important concern, and that existing approaches to logical database decentralization are inadequate. In response to the needs described above, we have developed a new approach to database system (software) architecture, termed *federated database systems*⁵. The goal of the federated database concept is to support the logical decentralization and partial integration of databases.

A *federated database* consists of a number of logical components, each having its own user-level structural specification (*component schema*). The components of such a *federation* are related but independent, and they may or may not be disjoint. Typically, a component corresponds to a collection of information needed by a particular user or application, or a collection of closely related applications. The components in a federation are tied together by one or more *federal schemas* that describe the data that is to be shared by

the various federation components. The *federal controller* is a database system functional module that supports communication and translation of data among the components of a federation, based on the federal schema(s).

A number of important detailed technical issues of the federation concept are currently under study, including the following:

- There are a number of structural alternatives for a federation, concerning the number and organization of the federal schemas, and their relationship with the component schemas. For example, a single federal schema can be formed to link all components, or a hierarchy of federal schemas can be used. A framework of design alternatives is currently under study.
- A mechanism is required to specify the nature, meaning, and structural organization of that portion of each component's data that is to be made available to the federation. Further, a mechanism is needed to support the definition of the (virtual) federal schema(s), and for logically linking the federal and component schemas (to support data communication and conversion via the federal controller). For these purposes, we are employing a semantic database model as a specification technique for both the (sharable portions of the) component schemas and the federal schema(s); a mapping approach based on this model is under development.
- The details of the operation of the federal controller are currently under study, specifically focusing on how both retrieval and update transactions are to be handled when federal schema(s) are referenced.

The relationship of the logical decentralization of data (in a federation) with its physical distribution (over a computer network) is being investigated; the present focus is on assessing the applicability of recent research on physically distributed databases, specifically concerning multiple copy control and concurrent database access.

A major goal of this project is to develop a *federated database system development aid*, the purpose of which is to assist the designer(s) of federated database systems. This prototype tool will provide facilities to logically interconnect database schemas (component schemas); the tool will interact with a designer and accept a set of schemas that are to be merged into a federation. The same semantic database model will be used to define each component schema, and the federal schema. The result of this merging process will be a virtual federal schema, and a mapping that specifies how the data structures and operations of the federal schema are related to those of the component schemas. It should be noted that the interactive tool focuses exclusively on logical mappings between data structures and operations⁶.

⁴A Database Transaction Specification Methodology for End-Users; Technical Report 79-6, Computer Science Department, University of Southern California, Los Angeles CA; December 1979.

⁵"A Federated Architecture for Database Systems", with D. Heimburger; Proceedings of National Computer Conference, Anaheim CA, 1980.

⁶Issues concerning the physical implementation of federated database systems are addressed in the third research project, described below.

An important use of the prototype federated database system development aid described above is in establishing controlled sharing among pre-existing databases. To accomplish this, a semantic schema is defined for each component database, and the tool is then used to define a federal schema based on these component schemas. A prototype federal controller, an integral part of the development aid, will automatically support data communication and translation among the components, via the federal schema. The last step in logically connecting the component databases is to develop a mapping from the actual database structure of each component to its corresponding semantic schema; this can either be done on a case-by-case basis, or mappings to generic classes of general-purpose database systems can be developed (e.g., to a relational database management system, to CODASYL-DBTG, etc.).

A somewhat different use of the federation concept is as a software engineering approach, to support the design and evolution of data-intensive information systems. The idea is to use the federation concept as an abstraction technique, and as a basis for a software engineering methodology for large, complex systems whose main tasks involve the manipulation of data. In this approach, the structure of a software system is centered on the data and on the operations that can be performed on the data. Two of the most important research issues here are accommodating multiple levels of federal schemas, and connecting the schemas with procedures that implement them (e.g., using an abstract datatype based programming language, such as ADA or CLU). We believe that this approach to the development of data-intensive systems has a number of advantages over existing software engineering methodologies, e.g., in that data and associated operations are highly structured (much more than, say, with the abstract datatype approach).

4. Transformation of Semantic Schemas into Physical Representations

The third major research project currently underway concerns the transformation of databases specified with a semantic database model (such as the event model) into a physical implementation. One approach to studying this issue is to design and build a new general-purpose database management system based on a semantic model and on the federated database concept, e.g., as discussed in⁷. In particular, we believe that the information available in a semantic schema is very useful in optimizing the storage structures and access methods that will be used to store and access the data. Unfortunately, it would be extremely difficult to design and implement a prototype general-purpose database management system based on a semantic model and the federation concept;

for example, facilities supporting recovery, protection, concurrency control, etc. would all have to be provided.

Instead of attempting this monumental task of developing a complete database management system prototype, this project is focused specifically on the mapping of semantic schemas into physical representations. A prototype tool will be designed and implemented, which will take any event-model schema and with the interactive guidance of a designer, and will yield a set of physical storage structures and operation procedures that implement the logical data structures and operations of the schema. In this context, we also intend to study the issues of data allocation (multiple copy control, etc.) and concurrent database access to physical data within a computer network.

The overall approach in this project is to employ a specific kind of *transformation system*, and a particular set of *transformations* for a given semantic database model (the event model). The transformation system accepts a semantic schema and associated predicted performance requirements, and generates as output a physical implementation to support the data structures and operations defined in the schema. The transformation system is interactive, calling on a designer to assist in the stepwise process of refining the schema so that it eventually becomes executable. Each transformation thus introduces additional implementation detail. The advantages of the transformation approach include the ability to take advantage of the designer's expertise and knowledge of current and anticipated database usage, and the ability to modify a schema or performance requirements and easily reapply the transformations.

5. Summary

The three research projects described above are obviously concerned with database abstraction and conceptual modelling. As such, we are drawing heavily on research results from other disciplines, viz., knowledge representation mechanisms in artificial intelligence, and abstract datatypes and associated work in the programming language domain. Moreover, the three projects described above exhibit a unifying overall goal: namely, to develop and demonstrate research techniques to support the design, use, and evolution of data-intensive information systems. In addition to the individual utility of each of the three projects described above, these three related efforts and the prototype tools form a comprehensive *database system development environment*: the semantic model and design methodology are used to specify individual schemas, the federated database approach supports the interconnection of schemas and the unification of database handling facilities with programs, and the transformation approach supports the translation of schemas into executable database systems. We believe that this work has particular applicability in the areas database system design, office information systems, personal database systems, and the architecture of future general-purpose database management systems.

⁷"On the Architecture of Database Management Systems", with M. Hammer; Inteltech State of the Art Report on Data Design, 1980.