USL / DBMS NASA / RECON

WORKING PAPER SERIES

Report Number

DBMS.NASA/RECON-20

The USL/DBMS NASA/RECON Working Paper Series contains a collection of reports representing results of activities being conducted by the Center for Advanced Computer Studies of the University of Southwestern Louisiana pursuant to the specifications of National Aeronautics and Space Administration Contract Number NASW-3846. The work on this contract is being performed jointly by the University of Southwestern Louisiana and Southern University.

For more information, contact:

Wayne D. Dominick

Editor
USL/DBMS NASA/RECON Working Paper Series
Center for Advanced Computer Studies
University of Southwestern Louisiana
P. O. Box 44330
Lafayette, Louisiana 70504
(318) 231-6308

(NASA-CR-184528) SOME ISSUES IN DATA MODEL MAPPING Final Report, 1 Jul. 1985 - 31 Dec. 1987 (University of Southwestern Louisiana. Lafayette. Center for Advanced Computer Studies.) 26 p CSCL 05B
N89-14568
Unclas
G3/82 0183566

SOME ISSUES IN DATA MODEL MAPPING

Jamal R. Alsabbagh

The University of Southwestern Louisiana
Center for Advanced Computer Studies
Lafayette, Louisiana

May 10, 1985

ABSTRACT

Numerous data models have been reported in the literature since the early 1970's. They have been used as database interfaces and as conceptual design tools. The mapping between schemas expressed according to the same data model or according to different models is interesting for theoretical and practical purposes. This paper addresses some of the issues involved in such a mapping. Of special interest are the identification of the mapping parameters and some current approaches for handling the various situations that require a mapping.

TABLE OF CONTENTS

1. INTRODUCTION

A variety of data models have been proposed since the early 1970s [Kerschberg et al., 76; Tsichritzis and Lochovsky, 82]. Many more are being reported in the literature continuously. The notion of a data model seems to have gained prominence with the proposal of the relational data model [Codd, 71] as a definition rather than an implementation. Earlier database interfaces (hierarchical and network) were abstracted from the then existing implementations to define the hierarchical and network data models. The above three models represent the basis for almost all existing database management systems (DBMS). Few exceptions are emerging for experimental purposes. More powerful models are mainly used as conceptual design and knowledge representation tools [Bracchi et al., 84].

The search for answers to questions like why another data model and what new useful features a proposed new model offers gained importance since the mid 1970s. Attempts were made to draw a framework for identifying the common concepts of the various models [Kerschberg et al., 76]. More recent efforts try to gain insight into the basic notions of data, representation and modelling with attempts to consolidate the approaches to these problems.

Further, such efforts attempt to draw upon the experience and research results on data and knowledge representation in the fields of information systems, programming languages and artificial intelligence [SIGMOD, 80; Bracchi et al., 84].

The prolification of data models have naturally lead to the study of their equivalence and mapping to each other. This problem is of theoretical as well as practical interest [Borkin, 80].

The theoretical interest is due to the fact that a formal study of equivalence can lead to deeper understanding of the basic concepts of modelling. Further, such formality can help in understanding and resolving the fine differences between the different models.

The practical interest exists because of the many situations in which the notion of equivalence is central. Examples of such situations are distributed database systems (DDBS), application conversion, database restructuring, and information system design.

Section 2 provides basic definitions of relevant terminology and concepts. Section 3 examines the parameters involved in the mapping process. In Section 4, we elaborate on the practical situations in which model mapping is a major concern. Section 5 contains a discussion and conclusions.

2. BASIC DEFINITIONS

A few basic definitions are needed before the mapping problem is discussed in later sections. An extensive discussion of the basic concepts can be found in [Tsichritzis and Lochovsky, 82].

The basic entities of the real world that data models attempt to capture are the objects, their properties and the relationships that exist between them.

An object is anything of interest in the real world which needs to be represented by the model. The class of entities that need to be represented as objects is usually limited to those for which the user is interested in describing some characteristics. For example, color can be represented as an object if we are interested in its wavelength. Otherwise, it is more economical to consider it as a property of objects. The reduction in the number of represented objects leads to simpler models [Biller and Neuhold, 78]. An object becomes identifiable by an object name. The characteristics of an object which are of interest to the application are called object properties. Each property may have any one of a set of application dependent values; a property can then have a property value. The name of a property is an attribute. If a certain attribute has a unique value for each object in the system, then that attribute is called the key and may be used to identify the various objects.

Objects may be grouped into classes called categories. A category acknowledges the similarity between its objects and therefore allows economy in the definition of common properties which are shared by members of the same category. Categorization is then a form of data typing as well as data abstraction. Categories may be defined recursively in terms of other categories to build more complex objects.

Objects may participate in a relationship(s) with other object(s), therefore defining a context for object existence.

A data model is a set of rules that can be applied in order to generate a representation of some subset of interest of the real world. Generally, the rules are of two types such that they can together describe the static as well as the dynamic properties of the application; each of the two types is discussed below.

The generating rules allow building a static picture of reality in terms of the data structures that are permitted by the data model. These generating rules may further be divided into two subsets. First, structure specification rules generate the categories and structures within the limitations of the representational features of the data model. For example, the hierarchical model permits the building of records from object designators and object attributes. Records may then be related in a tree fashion only. Second, constraint specification rules allow expressing limitations on the data as dictated by the semantics of the application and which may otherwise be allowed by the structure specification rules.

For example, if two students in a university cannot have the same identification number, then a network model with "duplicates-not-allowed specification" will express such a constraint. In other words, they specify logical restrictions on the data.

Second, a set of operations is defined as part of the data model. They specify the ways in which the generated structures may be manipulated.

A schema is a description of reality generated by the application of the generating rules of the data model to a specific slice of reality. Consequently, a schema is a composition of two parts: a structure schema and a constraints schema.

A database (DB), is a specific collection of application data together with DB control information (e.g., currency indicators) which is structured according to the specified application schema. A specific DB can have any one of many possible DB occurrences depending upon the actual data values and control information stored in it at a particular moment in time. A DB occurrence may be in different DB states at different times depending upon the values of its control information; such state transitions are the direct results of executing the operations defined above.

3. THE MAPPING PARAMETERS

In order to study the mapping problem, the mapping parameters need to be identified. As discussed in Section 2, a data model has three components, they are: structure rules, constraints rules, and operations. These three parameters allow the specification of the mapping between different schemas. It will be shown, however, that structure and constraint specifications can, and have to be considered together as one parameter. More interesting problem can be formulated if we include in the mapping the possibility that an actual database exists and is structured according to a schema which is expressed in some data model. Such new problems are of practical significance, as will be seen in Section 4. The remainder of this section will discuss each of the three mapping parameters.

3.1 Structure and Constraints: As described earlier, a schema describes some part of interest of the real world. If a mapping is to produce an equivalent schema (i.e., a description of the same reality), then the entire source schema should be mapped. Therefore, the mapping will only be meaningfull if both structure and constraints are mapped together. If the structure is mapped only, then a database instance generated according to the target schema can contain data values which would be rejected by a database associated with the source schema.

Structure mapping is generally easier to handle than constraints mapping because the latter can span structural units which may be broken or combined in the mapping process and therefore make the tracing of the original constraints specifications more difficult. Further, implicit constraints are more difficult to map than explicit constraints because they are tightly coupled to the structure. The difficulty with structure and constraint mapping is that it may not be reversible, thus not equivalent. In the relational model, for example, decomposing and then combining relations may cause a loss of some of the original functional dependencies.

Structure and constraint mapping is largely ad hoc due to the informal definition of the source and target models. The procedure is either algorithmic or utilizes an intermediate mapping model.

3.2 Operations: Operations are an integral part of the data as defined in Section 2. The need for mapping operations arises in two cases. First, when the source and target schemas are defined according to different models and the user can only operate directly on the source schema. An example is when a relational view is built on a network database. Second, when the source schema is a subset of the target schema, for example, when a view is provided on an application schema. Both cases represent in fact the same type of situation; the source and target schemas are different in the sense that they either represent the same world differently or they represent different worlds.

The mapping of query operations is generally easier than that of updates. The reason is that updates may imply a violation of some target schema constraints which the source schema is not aware of, as in the situation of view updates, for example.

3.3 Databases: This parameter refers to the cases when an actual DB instance is mapped into another DB instance. The DB may be restructured if both DBs are according to schemas expressed in the same data model. If the data models are different, then DB conversion is involved.

4. THE MAPPING PROBLEMS

The general mapping problem can be studied by considering the three parameters of the previous section which are involved in the process [Tsichritzis and Lochovsky, 82].

Structure and constraints may be mapped within the same model or across different models. The operations may or may not be mapped. A target DB may or may not exist. If a target DB exists and is derived from an existing source DB, then this is a constructive mapping. Finally, if a target DB can exist but ,when it does, is not derived from a source DB even if the latter exists, then this is nonconstructive mapping. There are therefore eight possible situations. Each of the eight situations is significant in the sense that each one represents a real practical problem, although they intersect with each other. In this section, each of the above situations will be discussed separately.

4.1 Schema Restructuring Mapping:

This is the case when a source schema is used to derive an equivalent target schema; both schemas are defined according to the same data model. Only the static properties (structure and constraints) are mapped. Operations are not mapped since both schemas are expressed in the same model and are equivalent. No DB instance conversion is involved.

This type of mapping is useful during schema analysis and design. For example, a relational schema may be transformed from one normal form to another to discourage users from making meaningless joins or to improve the physical access efficiency of the stored relations [Date, 82]. In the hierarchical model, such restructuring is used to attack the clustering and placement problems [Teory and Fry, 82].

4.2 View Mapping:

This is the case of operating on a DB through a schema which is a subset (view) of the DB schema. Both schemas are defined according to th same data model. The operations are mapped because the two schemas are not equivalent. The mapping is nonconstructive because the target DB is not derived from a source DB.

This mapping arises when it is desirable to provide various users with logical access to subsets of interest rather than to the whole schema and DB. A view on a relational schema, for example, may allow renaming and permutation of columns, vertical or horizontal subsets from a table, different units for value representation, and/or a join of many relations into one [Chamberlain et al., 75]. Some restrictions on allowable update operations have to be imposed to avoid violations of central integrity rules which are transparent to the view.

Integrity constraint which are applicable at the view level may be defined. In this case, it is important to insure that such constraints are a logical consequence of the central constraint. [Klug and Price, 82] provide an algorithm for such checking which is applicable to views that are derived from the base relations through a restricted set of relational operators.

4.3 Schema Translation Mapping:

This is the case when the static description of an application in some data model needs to be transformed into an equivalent description according to another data model.

An application of this type of mapping is in the area of transforming application requirements into a design. The requirements are first expressed in some semantic-type higher order data model then the resulting description is mapped into the model of an existing DBMS (almost always a relational, hierarchical, or network). Model operations are not mapped since the conceptual model is discarded because the operational interface to the designed system will be through the implementation schema directly. The mapping is nonconstructive because no source DB is transformed into a target DB.

The trend in information systems design methodologies is to map the schemas informally manually. Application specification information (e.g., usage patterns and user procedures) is carried from previous stages to simultaneously restructure the implementation schema and also to produce program specifications [Ceri, 83].

4.4 Operation Transform Mapping:

This mapping refers to the case when a schema according to some data model is provided as a user interface to another schema of a different data model. In the general case where many interface schemas of various models are provided, the situation is the three schema approach of the ANSI/X3/SPARC report. The model operations of the interface model need, of course, to be mapped on those of the base schema. This mapping is nonconstructive.

GEM [Tsur and Zaniolo, 84] is a semantic model interface built on the relational system INGRES. Such an approach is claimed to provide an evolutionary method of extending DBMS capabilities since it relies on the host to provide the usual services of concurrency control, security, integrity, and others. Further, it provides a test-bed for new interface models.

Another example is to provide a hierarchical interface to a relational DBMS [Lien, 80]. It is argued that by decomposing the relations into elementary nondecomposable relations and then organizing these into a tree, some application semantics can be expressed; a feature which is not possible with the relational model.

4.5 Database Reorganization Mapping: This mapping describes the case when a DB is rebuilt due to redefining its schema. The two schemas are in the same model, operations need not be mapped, and the mapping is constructive.

Schema redefinition and the consequent DB rebuilding can be an expensive undertaking. Major application changes which might require the addition, deletion, or splitting of relations or keys can affect even the application programs. The writing of conversion programs can be a significant task too, specially if the DML was host embedded.

[Shneiderman and Thomas, 82] have developed a system for automatic conversion of relational databases. The system is based on three types of possible transformations. First, information preserving transformations where no information is lost and the transformation is therefore reversible. Second, data dependent/independent transformations where actual data values have to be checked before transforming; e.g., the deletion of a certain attribute may render the key nonunique.

Third, program dependent/independent transformations since the deletion of a value which is used as a predicate will not preserve mapping equivalence.

4.6 Homogeneous Distributed Database Mapping:

This mapping problem occurres when a global schema is provided as a user interface to a collection of distributed databases. The global and all the local schemas are according to the same data model. An operation on the global schema then needs to be mapped into equivalent operations on the accessed local sites. The factor which makes this mapping different in classification from view mapping is that it is constructive. The reason is that there actually exists a virtual total DB instance and the local DB instances are actually fragments of it. The local DB instances can be regarded as if they were mapped from the virtual global DB, each according to its own schema.

[Rothine, et al., 80] discusses the design philosophy of SDD-1; a distributed DBMS developed at the Computer Corporation of America. SDD-1 is a relational system and has a relational language called Datalanguage. The DB consists of global, but virtual relations (logical relations) which are partitioned into units of distribution (logical fragments). Partitioning is done at the design stage and is a two-step process, consisting of horizontal and vertical partitioning. Horizontal partitioning is a selection process to produce subsets of relations.

Vertical partitioning is a projection to produce shorter tuples. Transactions are compiled into parallel programs. Users interface with the logical schema; the system maps the transactions to the relevant logical fragments through a table look-up process which extracts global and local characteristics.

4.7 <u>Database Translation Mapping</u>: This is the problem of converting an existing DB instance into another DB instance. The source and target schemas are of different models. For example, converting an existing hierarchical DB into a relational DB. The mapping is then constructive. The operations are not mapped since the DB interface will be through the target schema only.

This mapping is necessary when an enterprise needs to take advantage of new hardware or software products and if the shift involves a different-model DBMS. The cost of such a move, in the absence of tools, can be very costly. Massive investment is needed to write conversion programs and rewrite the application programs. It is estimated that the need for DB translation is quite frequent; once every about five years in most organizations [Su, 82].

Some support tools have been developed to aid in the process. One such tool is EXPRESS (data EXtraction Processing and REStructuring System) [Shu et al., 77]. It can extract data from multiple files then restructure and store it in target file(s). It has two main languages; DEFINE for data description and CONVERT for restructuring. EXPRESS does not handle application programs conversion.

4.8 DB Cooperation Mapping: The last type of mapping arises in heterogeneous DDB systems where a local site can communicate with other sites and no restrictions are imposed on the data model of any site in the system. This mapping is constructive according to the argument of Section 4.6. The operations are mapped also since the interface model is different from one or more of the target models.

An example of a heterogeneous DDBMS is MULTIBASE. It was developed as an experimental system by the Computer Corporation of America [Landers and Rosenberg, 82]. The users interface with the system through one global schema; they are unaware of data locations or local schemas. The system decomposes each query into a set of subqueries, formulates an efficient execution plan through proper subquery routing, controls and optimizes data movements, resolves data type representation conflicts, then finally reports a consolidated response. The global language is DAPLEX which is based on the functional model.

5. CONCLUSIONS

The study of data model equivalence and mapping is both theoretically and practically interesting. Theoretically, a basis can be formulated to insure mapping correctness in addition to enhancing the understanding of basic modelling concepts. Practically, many significant situations exist in the areas of DBMS system development and utilization in which mapping is central.

The growing trend towards friendly user interaction requires efficient, easy, and possibly dynamic redefinition of one or more user interfaces at the local stations.

Much of the work on mapping is informal, although exceptions exist [Borkin, 80]. This situation is mainly due to the fact that the models themselves are defined informally. Some formalism is emerging but it is not widely used or evaluated. Of particular promise is the use of logic [Jardine and Reuber, 84], and axiomatic definition [Brodie, 82].

BIBLIOGRAPHY

[Biller and Neuhold, 78] H. Biller, E. J. Neuhold, "Semantics of Data Bases: The Semantics of Data Models," Info. Sys., vol. 3, 1978, pp 11-30.

[Brodie, 82] M. L. Brodie, "Axiomatic Definitions For Data Model Semantics," Info. Sys., vol. 7, 1982, pp 183-197.

[Borkin, 80] S. A. Borkin "Data Models: A Semantic Approach for Database Systems," MIT Press, Cambridge, MA., 1980.

[Bracchi and Nijssen, 79] G. Bracchi, G. M. Nijssen (eds.) " Data Base Architecture," North-Holland, Amesterdam, 1979.

[Brodie et al., 84] M. L. Brodie, J. Mylopoulos, J. W. Schmidt (eds.) "On Conceptual Modelling: Perspectives From Artificial Intelligence, Databases, Programming Languages," Springer-Verlag, New York, 1984.

[Ceri, 83] S. Ceri (ed.), "Methodologies And Tools For Data Base Design," North-Holland, Amesterdam, 1983

[Chamberlain et al., 75] D. D. Chamberlain, J. N. Gray, I. L. Traiger "Views, Authorization and Locking in Relational Database Systems," Proc. AFIPS NCC, vol. 44, 1975, pp 425-430.

[Date, 82] C.J. Date Introduction to Database Systems, vol. 1, 3rd ed., Addison-Wesley, Reading, MA., 1982.

[De et al., 81] P. De, W. D. Haseman, Y. H. So, "Four Schema Approach: An Extended Model For Database Architecture," Info. Sys., vol. 6, 1981, pp 117-124.

[Fry, 82] J. P. Fry, "Conversion Technology, An Assessment," ACM SIGMOD RECORDS, vol. 12, no. 2, 1982, pp 39-61.

[Fry and Jervis, 74] J. P. Fry, D. W. Jervis "Towards a Formulation and Definition of Data Reorganization," Proc. ACM SIGMOD, 1974, pp 83-100.

[Hawryszkiewycz, 80] I. T. Hawryszkiewycz, "Alternate Implementation of The Conceptual Schema," Info. Sys., vol. 5, 1980, pp 203,217.

[Jajodia and Ng, 84] S. Jajodia, P. Ng, " Translation of Entity-Relationship Diagrams Into Relational Structures," The Journal Of Systems And Software, vol. 4, no. 2/3, July 1984, pp 1231-133.

[Jardine and Reuber, 84] D. A. Jardine, A. R. Reuber "Information Semantics and The Conceptual Schema," Info. Sys., vol 9, 1984, pp 147-156.

[Kerschberg et al., 76] L. Kerschberg, A. Klug, D. Tsichritzis, "A Taxonomy of Data Models," Systems for Large Data Bases, Lockmann and Neuhold, eds., North-Holland Publishing Co., Amesterdam, 1976, pp43-64.

[Klug and Price, 82] a. Klug, R. Price, "Determining View Dependencies Using Tablaux," ACM TODS, vol. 7, no. 3, September 1982, pp 361-380.

[Klug and Tsichritzis, 77] A. Klug, D. C. Tsichritzis "Multiple View Support Within The ANSI/SPARC Framework," Proc. 3rd Int. Conf. VLDB, 1977, pp 477-488.

[Landers and Rosenberg, 82] T. Landers, R. L. Rosenberg, " An Overview of MULTIBASE," Distributed Data Bases, H.J.Schneider, ed., North-Holland Publishing Co., 1982, pp 153-184.

[Lien, 80] Y. E. Lien "Hierarchical Schemata For Relational Models," ACM TODS, vol. 6, no. 1,March 1981, pp 48-69.

[Lochovsky, -81] F. H. Lochovsky, "Review of: Data Models: A semantic Approach For Database Management Systems," ACM SIGMOD RECORDS, vol. 12, no. 1, 1981, pp 16-17.

[Manola and Pirotte, 80] "An Approach to Multi-Model Database Systems," Proc. 2nd Int. Conf. on Databases, Cambridge, UK, Dean and Hammmersley (eds.), Wiley-Heyden Ltd., UK, 1983, pp 53-75.

[Navathe, 80] S. B. Navathe "Schema Analysis For Database Restructuring," ACM TODS, vol. 5, no. 2, June 1980, pp 157-184.

[Navathe and Merten, 74] S. B. Navathe, A. G. Merten "Investigation Into The Application of The Relational Model to Data Translation," Proc. ACM SIGMOD, 1974, pp 123-138.

[Rothine et al., 80] J. B. Rothine et al., "Introduction to A System For Distributed Databases," ACM TODS, vol. 5, no. 1, March 1980, pp 1-17.

[SIGMOD, 80] Proc. Of The Workshop On Data Abstraction, Databases, And Conceptual Modelling, Pingree Park, Colorado, June 1980, ACM SIGMOD RECORD, vol. 16, no. 1, 1980.

[Shneiderman and Thomas, 82] B. Shneiderman, G. Thomas, "An Architecture For Relational Database Systems Conversion," ACM TODS, vol. 7, no. 2, June 1982, pp 135-257.

[Shu et al., 77] N.C. Shu "EXPRESS: A data EXtraction, Processing and REStructuring System," ACM TODS, vol. 2, 1977,pp 134-174.

[Su, 82] S.Y.W. Su "Database System Conversion: Problems and Solutions," IEEE COMPSAC 82, Chicago, Ill., Nov. 1982, p 55.

[Teory and Fry, 82] T.J. Teory and J.P. Fry "Design of Database Structures," Prentice-Hall, 1982.

[Tsichritzis and Lochovsky, 82] D. C. Tsichritzis, F. H. Lochovsky "Data Models," Prentice-Hall, New Jersey, 1982.

[Tsur and Zaniolo, 84] Tsur and Zaniolo " An Implementation of GEM- Supporting A Semantic Data Model On A Relational Back-End," Proc. SIGMOD Annual Meeting, Boston, MA., June 1984, ACM SIGMOD RECORDS, vol. 14, no. 2, 1984, pp 286-295

[Vassiliou and Lochovsky, 80] Y. Vassiliou, F. H. Lochovsky "DBMS Transaction Translation," Proc. COMPSAC 80, IEEE, 1980, pp 89-96.

[Zaniolo, 79b] C. Zaniolo "Multimodel External Schema For CODASYL Database Management Systems," in [Bracchi and Nijssen, 79], pp 171-190.

4.21

1. Report No. IN-82

2. Government Accession No. 183066 ~~147396~~

3. Recipient's Catalog No.

4. Title and Subtitle

26P.

USL/NGT-19-010-900: SOME ISSUES IN DATA MODEL MAPPING

5. Report Date
May 10, 1985 DATE OVERRIDE

6. Performing Organization Code

7. Author(s)

JAMAL R. ALSABBAGH

8. Performing Organization Report No.

9. Performing Organization Name and Address

University of Southwestern Louisiana
The Center for Advanced Computer Studies
P.O. Box 44330
Lafayette, LA 70504-4330

10. Work Unit No.

11. Contract or Grant No.
NGT-19-010-900

12. Sponsoring Agency Name and Address

13. Type of Report and Period Covered
FINAL; 07/01/85 - 12/31/87

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Numerous data models have been reported in the literature since the early 1970's. They have been used as database interfaces and as conceptual design tools. The mapping between schemas expressed according to the same data model or according to different models is interesting for theoretical and practical purposes. This paper addresses some of the issues involved in such a mapping. Of special interest are the identification of the mapping parameters and some current approaches for handling the various situations that require a mapping.

This report represents one of the 72 attachment reports to the University of Southwestern Louisiana's Final Report on NASA Grant NGT-19-010-900. Accordingly, appropriate care should be taken in using this report out of the context of the full Final Report.

17. Key Words (Suggested by Author(s))

Database Interface, Conceptual Design Tools, Data Models, Mapping Schemas, Data Model Mapping, Information Storage and Retrieval Systems

18. Distribution Statement

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
27

22. Price*

*For sale by the National Technical Information Service, Springfield, Virginia 22161