The Reference Model for DBMS's

R. Hotaka
University of Tsukuba
Sakura, Ibaraki 305 Japan

ABSTRACT

A reference model for DBMS's is a data model which is referred to as the common base of various activities such as DBMS standardization activities or procurement of commercial DBMS's.

'Equivalent family of self-descriptive base model'-approach was explained which has been edited by Japan as a proposal of the reference model to ISO TC 97/SC 5/WG 5.

The approach does not limit the candidates for the reference model to one, but a family of equivalent reference models are allowed. As a good property of reference models, self-descriptivity was assumed.

1 Introduction

The reference model gives the common reference base of various DBMS's to allow comparisons among them. To this end, it must have the ability to describe necessary details of each DBMS model. Thus sufficient resolution of the model is required. Since the development of the model is in preliminary stage, only the rough direction of writing reference model is sketched now. The details will be supplemented deliberately.

By the same reason, following topics are excluded at first though they are typical topics of DBMS's.

. database recovery

. concurrency controls

. security

. integrity

. distributed database

. detailed discussions of views

. high level user interface

2 Methodological Aspects

How can we establish the DBMS reference model? (DAFTG 82) introduces several approaches to integrating and standardizing multiple data models. Though the situation is a little different, it can be used as the development policy of the DBMS reference model. The following six approaches are discussed in (DAFTG 82).

Independent Data Models

This approach suggests that the models shoud be standardized independently. This approach would allow each model to retain its unique characteristics and evolve independently of the other data models. However, different data models often have some common structures and operators. The adoption of this approach would preclude the adoption, or at least reduce the possibility of recognition and standardization of these common elements.

Base Model

The Base Model approach relies on the development of a more formal data model with powerful expressive capabilities. Other data models can then be expressed in terms of this base model. One could first standardize such a base model and then define (and standardize) the other data models. Powerful base models, such as those founded on set theory, currently rely on some form of symbolic logic and/or abstract mathematics to capture the expressive power required to define the other models.

Views over a Conceptual Schema

An other approach that has been suggested would be to first standardize a conceptual model and then define views in terms of the data models over the conceptual model.

Union of Data Models

Still another approach would be to combine selected data models into a union data model. This model would contain all of the data structures and operations that are associated with the selected data models. Standards for the different data models would then be subsets of this union data model.

Common Intersection

The Common Intersection approach suggests that the common data structures and operations of the models to be standardized should share a common standard. Independent extensions to support the differences in data models could also be standardized. With this approach, common standardization of similar structures and operators would be possible.

Coordinated Family of Data Models

This approach propose standardization of coordinated families of existing data models and continued development of conceptual data models. Under this approach, standardization activities could support and encourage commonality between various combinations of data models without specifying that all data models need to totally conform.

'Independent Data Model' approach gives up the very objectives of the commonality among various DBMS models from the beginning.

The practical usefulness of the 'Base Model' approach was questioned despite of its technical soundness(DAFTG 82).

'Views over a Conceptual Schema' approach presupposed the existence of standardized conceptual model. But conceptual models are still an active research topic. Also whether the conceptual model should come first or not is not certain since it may be possible to establish the conceptual model on top of what we now think of as data models. Clearly, standards for the underlying data models are needed first. Finally the value of mapping an existing data model is not clear. If a conceptual model is supposed to support an operation directly in terms of an enterprise, why would a user want to issue commands in terms of DBMS operations on records or tuples (DAFTG 82)?

(Date 80) is an example of 'Union of Data Models' approach. Two possible problems with this approach has been raised (DAFTG 82). First, in previous attempts to combine data models (Date 80), the definitions of the individual models were modified to form a more compatible and understandable union. Secondly, the individual data models could not be extended without affecting the union of the data models. (DAFTG 82) argues that it would not be possible to add a completely new structure or operation to one of the data models without also adding it to the union model.

'Common Intersection' approach might require changes in the definitions of the models or result in a very sparse intersection. For example, an intersection of some data models might include only "data elements" and "records". However, the concept of a "record" can actually differ from data model to data model. In some data models, refords can have some kind of repetitive elements(e.g. arrays or repeating groups). In others, records must be "simple"-- that is, a collection of attribute-value pairs.

Thus one common intersection might force only "data items" into a common standard. By forcing the common intersection of many current and future models for standardization, the data models would be signigicantly modified or the intersection would turn out to be practically empty. An empty intersection would result in standards similar to those developed under an independent data model approach.

The approach taken by 'Coordinated Family of Data Models' is very vague. In a sense there still remains the necessity of the other DBMS reference model in that this approach encourage commonality between various models and it seems that aim can only be achieved by establishing the alternate reference model.

We have seen several approaches to establish the DBMS reference model. Contemplating drawbacks of each approach, the following approach is proposed.

Equivalent Family of Self-descriptive Base Models

'Base Model' approach is technically sound, but we cannot uniquely determine the base model. There can be various candidates for the base model. We can avoid this difficulty as follows. Every DBMS model and function should be explained by using concepts of a reference model. Suppose there are two reference models M_1 and M_2. Then M_1 should be explained by using the concepts of M_2 and vice versa. In a sense, each reference model must be shown equivalent to each other. The euivalent reference models constitute a family of reference models any of which can be used as a DBMS reference model.

For the purpose of the reference model, multiple reference models would be inadequate. But we cannot determine the best one from the beginning. All we can do is to propose a family of reference models and allow users choose the proper one. The more frequently used reference model would become the de facto standard reference model. Note, however, a reference to a reference model will remain valid even if the different reference model becomes the standard reference model. This strategy is analogous to that of the definition of real numbers. The continuity of real number can be defined various ways: diminishing intervals (Cantor), cut (Dedekind), convergence of bounded monotone sequence and the existence of supremum (Weierstrass) etc. From each of the definitions, we can define the continuity of the real number and demonstrate each of the above definitions are equivalent. The real number theory is equivalent as far as it is based on one of the mutually equivalent definitions. In a sense, though the reference model of the real number is not unique, we can unambiguously discuss the real number.

<u>DBMS model independence</u>

In order the above scheme to be successful, every reference model must provide the fundamental descriptiptive power so that any possible (both present and future) DBMS model can be described. Otherwise, reference models will confront the situation where the model must be modified to adapt to the new DBMS model. This may make the previous reference to the older version of a reference model unstable and endanger the evolvability of the reference models. I.e., for the reference model, DBMS model independence is required.

<u>Self-description</u>

Though we allow multiple DBMS reference models, we can diminish varieties of the reference models by imposing the following axiom.

<u>Axiom</u> (Self-descriptivity)

Each DBMS reference model is self-descriptive.

A reference model can make the meaning of the concepts of a DBMS clear. At the same time, the meaning of the concepts of the reference model must be explained somehow. Yet the history shows that no models can explain the meaning of itself completely. If a concept should be explained, another different concept is necessary, but if we introduce a new concept to explain a concept, the new one remains to be explained. If we take this approach, we will have to introduce infinitely many concepts. Therefore, if the model has only finite concepts and if each concept must be explained in the model, recursive explanation is the only answer. We must content ourselves with using as few concepts as possible and establishing several relationships between these concepts. Some concepts may be left undefined and only relationships with other concepts stated. Since a concept is described by the concepts in the model itself, it is called self-descriptive (Abrial 74). (Abrial 74) says on page 9, "Notice that it is not only a 'theoretical game' but could be also very useful in helping human beings to modify the model by asking for information about its own behavior."

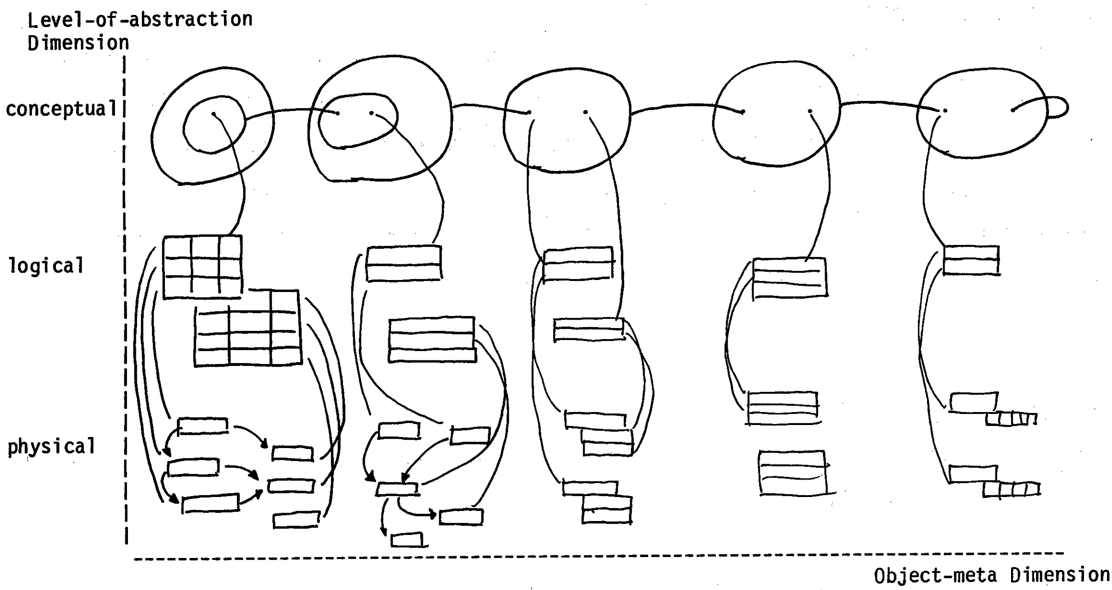
3 Data Analysis

We observe three dimensions of data models.

(1) Object-meta (or Extension-intention (Mark 84)) dimension

(2) Level-of-abstraction dimension

(3) Base-view dimension



object role	Application Data	Data Dictionary Data	Data Model Data	Data Model Designing Data	-
meta role		Application Schema	Data Dictionary Schema	Data Model Schema	Data Model Designing Schema

Fig. 1 Level-of-abstraction dimension vs. Object-meta dimension

Object-meta-dimension

If an object data is to be accessed, its meta data (or schema) must be specified at the same time. Thus in accessing application data, application schema is needed. Similarly, in accessing data dictionary data, data dictionary schema is needed. Object- and meta- data are relative concepts. When accessing 'Application data', 'Application Schema' plays the role of the meta data , whereas the same data plays the role of object data when it is accessed as the 'Data Dictionary' data (Lemke 83). In Fig. 1, one thing of two roles is written at the same level of the object-meta dimension.

In Fig. 1, we exhibit multiple levels of the object-meta dimension. The number of levels may differ from DBMS data models to models according to how far the model covers the universe of discourse. For example, 'Data Model Data' level and above may be missing in a DBMS model.

Though there may be several levels in object-meta dimension, it can be mapped into 2 level hierarchies as shown in Fig. 2 if the DBMS model itself is self-descriptive, which in fact contracts into 1 level objects.

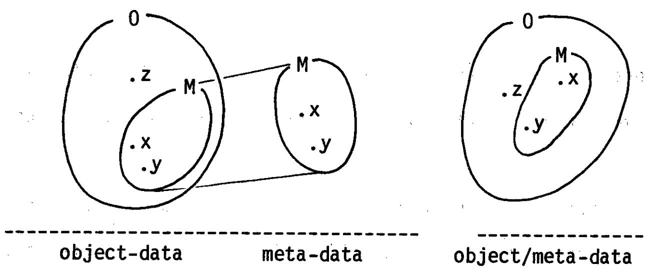


Fig. 2 Self-description mechanism

Level-of-abstraction dimension

Three levels are assumed in this dimension. Conceptual level deals with concepts which is not directly dealt by a DBMS but only resides in human mind. Logical level deals with names (Kangassalo 83). Physical or internal level deals with the particular representation of logical level objects.

Base-view dimension

Base data is the set of fundamental data. Fundamental data is the data which cannot be decomposed further. Each base data (conceptual, logical or physical) can be viewed several ways as views. In ordinary DBMS architecture, two dimensions level-of-abstraction dimension and base-view dimension are somewhat confused. Therefore they are distinguished here. A view is a set of compound data. Compound data is the data which can be integrated from fundamental data.

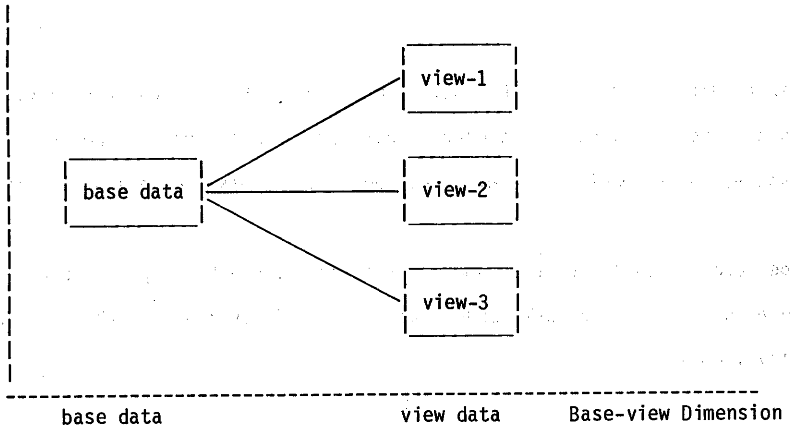


Fig. 3 Base-view dimension

4 Functions

Functions of a DBMS are operations which users of DBMS can perform on the database. In issuing a DBMS operation, a proper level of object data of each three dimensions must be specified.

There are various kinds of functions available in the existing DBMS's. But a reference model should first introduce the set of primitives. A primitive is a fundamental operation to be applied to the fundamental data. The reference model assume the set of fundamental operations (e.g. creation, deletion, reference). Next, the model states the way primitives are integrated to form compound operartion. The set of fundamental oprations might not be unique again but they should at least be minimal within the proposed reference model. A compound operation can be defined along two directions (Fig. 4).

Data / Operation	fundamental data	compound data
fundamental operation	primitive (fundamental operation against fundamental data)	fundamental operation against compound data
	compound operation against fundamental data	compound operation against compound data

Fig. 4 Fundamental vs. compound operarion

Since application schema is nothing but an object data in Data Dictionary Data Level, so-called

DDL functions can be viewed as the ordinary fundamental of compound operations but 1 level up in the object-meta dimension (Abrial 74) (Hotaka 77) (Jefferson 83) (Lemke 83) (Mark 84).

Each operation will specify the levels of each dimension by specifying the meta data which at the same time tell the range of scope of the object data.

5 Functional Interfaces and User Classes

Various user classes will need particular compound operations for their use. Typical tools for them will be identified (future tasks).

Examples of users:

- Database designer
- Database administrator
- Host language programmers
- User language programmers
- Parametric users

6 Description of Interfaces

First, an example of reference model interface of both data and functions will be supplied (future tasks).

If another reference model is proposed, their equivalencies are shown.

References

(Abrial 74) J. R. Abrial: Data Semantics, in 'Data Base Management' edited by J.W.Klimbie et al.. Proc. IFIP TC-2, Working Conference, 1974, North-Holland, pp.1-60.

(DAFTG 82) The Database Architectural Framework Task Group: An Architectural Framework for Database Standardization (Draft) July 7, 1982 ISO/TC97/SC 5/WG 5 N57.

(Date 80) C. J. Date: An Introduction to the Unified Database Language (UDL), VLDB, 1980, pp.15-29.

(Hotaka 77) R. Hotaka and M. Tsubaki: Self-descriptive Relational Database, VLDB, 1977, pp.415-426.

(Hotaka 81) R. Hotaka: Discipline for the Adoption of any (conceptual or not) Data Model, TC 97/SC 5/WG 5 N12.

(Hotaka 84) R.Hotaka: On the policy of writing Reference Model Specifications, TC 97/SC 5/WG 5 N124.

(Jefferson 83) D.K.Jefferson: Reference Model Priorities and Definitions, TC 97/SC 5/WG 5 N106.

(Kangassalo 83) H.Kangassalo: On the selection of the approach for the development of the reference model for DBMS standards, TC 97/SC 5/ WG 5 N104

(Lemke 83) J.Lemke: Summary of the Ad Hoc Group's Discussion on a Reference Model for DBMS, Appendix 4 of TC 97/SC 5/WG 5 N120.

(Mark 84) L. Mark and N. Roussopoulos: The Reference Model, in TC 97/SC 5/WG 5 N150 by D. Jefferson.

(Masunaga 84) Y. Masunaga: On the uniqueness of base reference models, personal communication, May, 1984.

(Sato 84) H. Sato: On the reference model, personal communication, June 20, 1984.