Suitability of data models as canonical models for federated databases *

F. Saltor, M. Castellanos & M. García-Solaco
Dept. Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
E-08028 Barcelona, Catalonia (Spain)

Abstract

We develop a framework of characteristics, essential and recommended, that a data model should have to be suitable as canonical model for federated databases. This framework is based on the two factors of the representation ability of a model: *expressiveness* and *semantic relativism.*

Several data models are analyzed with respect to the characteristics of the framework, to evaluate their adequacy as canonical models.

1 Introduction

When several databases (DBs) are to interoperate, they form a federation, and a data model must be chosen as the *canonical data model* (CDM) for the federation (we use the terminology of [SL90]). Work on federated or interoperable databases has often used an Entity Relationship (ER) model, or some extension of it, as the CDM; others have adopted an Object Oriented (OO) model. Is any data model equally adequate as CDM? This paper discusses some characteristics of a data model that make it suitable as the CDM of a federation.

Note that we are not trying to unearth the old -and inconclusive- debate about which is "the best" data model. Even if we recall two general factors in Section 2, we then place ourselves in the interoperability environment, and we develop a framework to analyze model suitability in this specific environment, not in a general, all encompassing case. We do not postulate that our framework is complete or definitive.

This paper is organized as follows: Section 2 presents the concept of representation ability as composed of two factors, expressiveness and semantic relativism. In Section 3, we apply these factors to interoperability, to develop a framework of characteristics. Section 4 analyzes several data models with respect to these characteristics. Conclusions are presented in Section 5. Space limitations prevent us from more detailed considerations, contained in [SCG91].

*This research has been partially supported by PRONTIC under project TIC89/0303

2 Representation ability of a data model

DBs are to *represent* conceptualizations that people have about reality, and to be *interpreted* as such conceptualizations. Therefore, an important characteristic of a DB is its *representation ability*, i.e. how well can the DB represent those conceptualizations. The representation ability of a DB is given by its data model. A data model is composed of structures, operations, and integrity constraints.

We see the representation ability of a data model as composed of two factors: 1) expressiveness; and 2) semantic relativism.

2.1 Expressiveness

By *expressiveness* of a data model we mean the degree to which the model can directly represent (express in a natural way) a conceptualization, no matter how complex this conceptualization might be, and which concepts compose it. This is similar to what was called "semantic expressiveness" in [HM81], "conceptual naturalness" in [Sh81], and "modelling support" in [Br82].

Expressiveness may be seen as composed of a structural part and a behavioural part. *Structural expressiveness*, is the power of the structures of the model to represent concepts, and to be interpreted as such concepts. *Behavioural expressiveness* reflects the power of the model to represent behaviors of concepts.

To illustrate the point, a model supporting generalization/specialization between superclasses and subclasses and aggregation/decomposition between complex data objects and their constituent data objects has more structural expressiveness than one that does not. For example, the relational model has no direct support for generalization nor for such an aggregation, i.e. has no constructs that can be directly interpreted as such abstractions, and therefore has less structural expressiveness than some extended ER or OO models that have such constructs. Other classical models are very poor in expressiveness.

A model supporting, not only generic operations of its structures and integrity constraints inherent to the

model, but also the definition of new operations and integrity constraints, has more behavioural expressiveness than a model not supporting this feature.

2.2 Semantic relativism

A DB should support not just one conceptualization, but many. A conceptualization is never absolute, it is relative to the point of view of a user or group of people, because different persons perceive and conceive reality in differing ways. A DB is not supposed to store a representation of each of these conceptualizations separately; it should store a single representation, encompassing all those conceptualizations, and avoiding redundancies. This representation is in accordance with the *database schema* of the DB (in the ISO terminology of [vG82]; the ANSI/SPARC term is "conceptual schema").

For the DB to support all users' conceptualizations, it must support one *external schema* for each conceptualization, and their *derivation* from the single database schema.

The *semantic relativism of a DB* is the degree to which it can accommodate all these different conceptualizations (of the same real world). The power of a DB to derive external schemas from its database schema is therefore the measure of semantic relativism of the DB. This is given by a mechanism of the DBMS to support external schemas (view mechanism), and by the model of the DB, i.e. the model of its database schema.

We call *semantic relativism of a data model* the power of its *operations* to derive external schemas (contrast with what is called "semantic relativism" in [Br82], [SP90], and "relativism" in [HM81]).

For example, the relational model is able to use the whole power of its algebra to derive external schemas (views), and has therefore a high degree of semantic relativism ([Sa86]).

3 Representation ability in the interoperability environment

Following the five level schema architecture of [SL90], when a number of preexisting DBs, called *component* databases, are to interoperate, a *canonical* data model (CDM) common to the whole federation, must be adopted. The database schemas of the component DBs (*local schemas*) are transformed from their native models to the CDM, giving *component schemas*. Each component schema is filtered into one or more *export schemas*. From export schemas of different component DBs, a *federated schema* is constructed — this process is called *schema integration*; several federated schemas can exist in a federation. Finally, from a federated schema a number of *external schemas* are derived.

The use of a common CDM solves the problem of syntactic heterogeneities, consequence of the use of different native data models. Semantic heterogeneities, resulting from different conceptualizations in the minds of the designers of the component DBs, are dealt with in the schema integration process.

Most schemas in this architecture are expressed in the CDM. The characteristics of the CDM are therefore very important. Let us look at some characteristics of a given data model that make it more adequate to be the CDM of a federated database system. Applying to this environment the same two factors of representation ability seen in the previous section, we develop a framework of essential and recommended characteristics.

3.1 Expressiveness

A CDM must have an expressiveness equal or greater than any of the native models of the component DBs that are going to interoperate, in order to capture the semantics already expressed with the native models. Moreover, it should support additional semantics made explicit thru a semantic enrichment process, in case that such a process is applied.

3.1.1 Semantic enrichment

Semantic heterogeneities are difficult to overcome, since there is a need to have a deep understanding of the meaning of the databases in order to detect and solve them. Unfortunately, as pointed out in 2.1, traditional data models have very limited expressiveness. Their overly simple data structures and operations allow the designer to express only a limited subset of his/her knowledge of the applications. As a consequence, the semantically poor local schemas are not a big help for acquiring this understanding. This leads to the need of enriching the schemas through a knowledge acquisition process by which all the semantics extracted is made explicit in an enriched representation of each database ([CS91]).

This is the reason why the CDM must be rich enough to represent the semantics already expressed in the local schemas, as well as additional semantics captured through the knowledge acquisition process. This knowledge may refer to structural and/or behavioural aspects of the applications. In this way, the transformation from local schemas into component schemas is not just a syntactic translation from one model to another, it includes a structural and/or behavioural semantic enrichment in order to upgrade the semantic level of the local schemas. The rich expressiveness of the CDM make it possible to express more semantic information in the component schemas, than the corresponding local schemas. The schema integration process will then be less difficult, by making use of these additional semantics to detect and solve semantic conflicts.

3.1.2 Characteristics of expressiveness

A federation of component DBs should not be closed; it should be open so that other DBs can enter the federation in the future. Anticipating that these new DBs could have native models richer than those initially forming the federation, the CDM should be chosen with a high degree of expressiveness, not just equivalent to the union of the structures of the initial native models and their enrichment.

Schema integration is easier if export schemas are semantically rich, but this is not enough. The CDM must have structures well suited to represent the federated schemas resulting from the integration.

From all these considerations, we find the following characteristics along each of the three *dimensions* in which semantic abstractions lie:

Characteristic **Ea**: In the dimension of *classification/instantiation*, the CDM must support classes and individuals. Support of metaclasses (and of metametaclasses) is recommended, but not essential.

Characteristic **Eb**: In the dimension of *generalization/specialization*, the CDM must support arbitrary many levels of superclasses - subclasses, with inheritance. Multiple inheritance and different kinds of specialization -disjoint, etc.- are recommended.

Characteristic **Ec**: In the dimension of *aggregation/decomposition*, the CDM needs *cartesian aggregation* (also called "aggregation") of objects, and *cover aggregation* (also called "grouping" or "association") of objects, to form a complex object; it must be closed with respect to these constructs. Other aggregations (lists, bags, etc.) may be useful, but are not considered as essential.

On the behavioural side of expressiveness, there is a need to model any behavior, and we formulate:

Characteristic **Ed**: The CDM must support the definition of new operations and of new integrity constraints (extensibility of behavior). Additionally, it is recommended that the implementation of these new operations be encapsulated. This helps in the schema integration process, because it allows, quite naturally, to hide the source(s) of data or operations, their local implementation(s), and the solutions given to semantic conflicts.

3.2 Semantic relativism

We have seen that two of the processes, in the context of interoperability, are the integration of export schemas into federated schemas, and the derivation of external schemas.

Besides the fact that expressiveness is needed for schema integration, this process absolutely requires, in the generalization/specialization dimension, the integration of several specialization lattices into a single, generalized lattice. Thus, a first characteristic in semantic relativism is:

Characteristic **SRa**: The model must allow the implementation of integration operators with an overall power at least equivalent to the *meet, join, fold*, etc. operations in [Mo87], or *gen, addsubtype*, etc. in [MNE88]. Support of *upward inheritance* (from subclass to superclass) as shown in [SN88], is recommended.

The need to support many different conceptualizations is even greater than in stand alone (centralized or distributed) DBs in the sense seen in 2.2. This requires a high degree of semantic relativism, expressed in:

Characteristic **SRb**: the operations of the CDM must have a power to derive external schemas at least as great as the view mechanism using the relational algebra ([Sa86]). In particular, it must be able to produce any kind of structures available in the model.

In addition to this essential characteristics, there are two other characteristics that are recommended for interoperability.

3.2.1 One basic structure

To facilitate schema integration from export schemas to a federated schema, the CDM should have just one basic structure rather than two or more.

Assume, for example, that an ER model, that has two basic structures (entities and relationships), is chosen as CDM. When transforming (converting) a local schema from its native model to the canonical ER model, a decision must be taken for each local class or type to transform it into either an entity or a relationship. For example, a married couple could be modeled as an entity in the component schema of a Marriages DB, or as a relationship in the component schema of a Persons DB (or even, for an Employees DB, as an attribute of Employee). But this is not the main point.

When constructing a federated schema from the export schemas in ER, something that was modeled as an entity in export schema A may correspond to a relationship in export schema B (clash of structures, or structural conflict); in the resulting federated schema F, it may be decided, (after some discussion, or using a methodology such as the one in [SP90]), to model it as an entity, but then special processes must be put in place, to convert from the relationship in B into the entity in F (and back for query processing), and to add new relationships in F –that connect it to the entities that it related in B– (and suppress them), with their corresponding behavior.

Moreover, if a user conceives it as a relationship, he has to be provided with an external schema E, derived from F by converting from an entity into a relationship, taking care of its attached relationships, and adjusting their behavior accordingly.

This example has shown how having to deal with two basic structures augments the already difficult process of schema integration with an added complexity. These conflicts simply do not appear if the CDM has just one basic structure, as happens in the relational, functional and OO models. Note that here the

question is not which basic structure is better -this would correspond to expressiveness-, but that there is only one, no matter which one, from this point of view of semantic relativism.

This is summarized in recommended *Characteristic* **SRc**: The CDM should have only one basic structure.

At first sight, this characteristic appears as opposite to expressiveness: we want a rich set of constructs for expressiveness, but only one structure for SRc. However, a basic structure may adopt many forms -corresponding to different semantic abstractions-, and still have a set of generic operations forming a *homogeneous, closed algebra.* For example, an OO model having just one basic structure: objects, with an homogeneous object algebra, can at the same time support generalization, complex objects, etc. Having two or more basic structures implies either a many-sorted algebra or a set of operations that is incomplete and fails to comply with SRb.

As another example, a model differentiating between attributes and methods will have integration conflicts that would not be present with a CDM using only functions: a given function in a federated schema could correspond to a stored attribute in a component DB and to a computed procedure in another.

3.2.2 Multiple semantics

When integrating export schemas, different users may have conceptualizations that are not subsets of a more general conceptualization, but that diverge in some respect. An example of [SL90] is the integration of colors of shoes from two DBs, DB1 and DB2: userA sees as cream what is cream in DB1 and what is tan in DB2, while userB considers cream what is tan or cream in DB1 and what is tan or white in DB2.

This is called *multiple semantics* in [SL90], and must be supported by the Federated DB system. One way to handle it is by having one federated schema for each semantic conceptualization ([SL90]), i.e. a federated schema for userA and another for userB. An alternative is to allow a single federated schema to support multiple semantics, for example through discriminants ([GS91]), and differentiate the respective semantics at the external schema level, i.e. an external schema for userA, and another for userB. One of the advantages of the second architecture lies in having a smaller number of federated schemas; on the other hand, it requires a CDM supporting multiple semantics at the federated schema level, both in structures and in operations able to derive external schemas for each semantic conceptualization.

We have therefore recommended *Characteristic* **SRd**: the CDM should support multiple semantics.

4 Analysis of data models

Let us now analyze a number of data models with respect to the essential characteristics for a CDM of Ea, Eb, Ec, Ed, SRa and SRb, and the recommended characteristics of Ea, Eb, Ec, Ed, SRa, SRc and SRd.

Pre-relational models such as the hierarchical and network models do not even satisfy essential characteristics: Eb, Ec, Ed, SRa, SRb.

The basic relational model does not satisfy Eb, Ec, Ed and SRa, even if it complies with SRb ([Sa86]), the essential part of Ea, and SRc (relations are the only structure), and can easily be extended with discriminated operations to satisfy SRd ([GS91]). All this applies to version 2 of the relational model ([Co90]), except that Ed is satisfied. The RM/T model of [Co79] is an improvement over them, because it satisfies the essential parts of Eb and Ec, and of SRa; it complies with SRb ([Sa87]); however, it now fails to satisfy SRc, because it has two basic structures (E-relations and P-relations, not counting catalog relations).

The basic ER model lacks in Eb, an essential part of Ec (relationships between relationships are not allowed), Ed and SRa; also in SRc, as has been shown in 3.2.1; even when equipped with an algebra, such as in [PS85], it does not satisfy SRb (because the operations can produce entities, but not relationships), nor SRd. Extensions to the ER model (EER [TYF86], ECR [EWH85], ERC+ [PS89]) improve upon ER, particularly in Eb, and also in Ec, but not in Ed; however, they are lacking in semantic relativism: even when equipped with some operations, no complete satisfaction of SRb, that is essential, nor of SRc and SRd. Among them, only ERC+ would comply with the essential part of SRa.

Functional models such as DAPLEX ([Sh81]), on the other hand, are good in semantic relativism: SRa, SRb, SRc (considering functions as their only structure, including entity types), and SRd, and satisfy all essential characteristics of expressiveness, but not those recommended.

Besides extensions to the ER model and RM/T, other models have incorporated generalization and aggregation abstractions, and have been called *semantic models* ([PM88]); examples are SDM ([HM81]), and TAXIS ([MBW80]). These models satisfy characteristics Ea, Eb, Ec, not only the essential parts, but recommended features as well; they do not comply with SRa, they may be considered to comply with SRb, it is arguable to which point they support SRc, and they do not comply with SRd. TAXIS, but not SDM, would support the essential part of Ed.

There are many OO models ([Ma89]) and models that claim to be object oriented. Considering those which satisfy [At89], they are very expressive: generalization, cartesian and cover aggregations, definition of new methods; they also satisfy the essential part of SRa. Some modern OO models are equipped with an adequate language to support views ([AB91], [SS91]), and comply with SRb, and therefore with all essential characteristics. Furthermore, since objects are their only structure, they satisfy SRc. Some of these models also comply with others of our recommended characteristics; VODAK ([KNS90]), for example, supports upward inheritance (SRa) thru metaclasses.

5 Conclusions

We have developed a framework of desirable characteristics for the canonical data model of interoperable databases, and applied it to a number of models.

From our analysis, it may be concluded that data models differ widely in their suitability as canonical data models for interoperable databases. Pre-relational models are not adequate at all, while, at the other extreme, functional and some OO models, supporting views, are the only ones which satisfy all essential characteristics of our framework, and seem best placed for the job.

In particular, the ER model, and its extensions, that have often been used as CDMs, are clearly inferior for this purpose to modern OO models. Extended ER models would need a complete algebra, producing all kinds of entities and relationships, to comply with SRb, and therefore with all essential characteristics; even so, they would be inferior to OO models at least in SRc.

Acknowledgments

We are indebted to A. Sheth and W. Litwin for their comments on an earlier version of this paper. We are grateful to the anonymous referees for their comments.

References

[AB91] S. Abiteboul & A. Bonner: "Objects and Views". Proc. ACM SIGMOD Conference. ACM SIGMOD Record vol 20, #2.

[At89] M. Atkinson et al: "The Object Oriented Database Systems Manifesto". Proc. Int. Conf. on Deductive and Object Oriented Databases, Elsevier, Amsterdam, 1990.

[Br82] M. Brodie: "On the Development of Data Models". In Brodie, Mylopoulos & Schmidt (eds.): On Conceptual Modelling, Springer Verlag, TIS, 1984.

[Co79] E. F. Codd: "Extending the Database Relational Model to Capture More Meaning". ACM TODS, vol4, #4.

[Co90] E. F. Codd: The Relational Model for Database Management Version 2. Addison Wesley, 1990.

[CS91] M. Castellanos & F. Saltor: "Semantic Enrichment of Database Schemas: An Object Oriented Approach". Proc. 1st Int. Workshop on Interoperability in Multidatabase Systems, Kyoto.

[EWH85] Elmasri, Weeldreyer & Hevner: "The category concept: An extension to the entity relationship model". Data & Knowledge Engineering, vol 1, #1.

[GS91] M. Garcia & F. Saltor; "Discriminated Operations for Interoperable Databases". Proc. 1st Int. Workshop on Interoperability in Multidatabase Systems, Kyoto.

[HM81] M. Hammer & D. McLeod: "Database Description with SDM: A Semantic Database Model". ACM TODS, vol 6, #3.

[KNS90] W. Klas, E. Neuhold & M. Schrefl: "Metaclasses in VODAK and their Application in Database Integration". In Arbeitspapiere der GMD, No.462, Darmstadt, Germany.

[Ma89] D. Maier: "Why Isn't There an Object Oriented Data Model?". In Information Processing 89 (Proc. IFIP WCC, San Francisco 1989), North Holland. Extended version as TR CS/E-89-002, OGC.

[MBW80] Mylopoulos, Bernstein & Wong: "A Language Facility for Designing Database Intensive Applications". ACM TODS, vol 5, #2.

[MNE88] M. Mannino, S. Navathe & W. Effelsberg: "A Rule-based Approach for Merging Generalization Hierarchies". Information Systems, vol 13, #3.

[Mo87] A. Motro: "Superviews: Virtual Integration of Multiple Databases". IEEE Transactions on Software Engineering, vol SE-13, #7.

[PM88] J. Peckham & F. Maryanski: "Semantic Data Models". ACM Computing Surveys, vol 20, #3.

[PS85] C. Parent & S. Spaccapietra: "An Algebra for a general Entity Relationship Model". IEEE TOSE vol 11, #7.

[PS89] C. Parent & S. Spaccapietra: "About Entities, Complex Objects and Object oriented Data Models". In Falkenberg & Lindgreen (eds.): "Information System Concepts: An In-depth Analysis". North Holland.

[Sa86] F. Saltor: "On the Power to Derive External Schemata from the Database Schema". 2nd IEEE Int. Conf. on Data Engineering, Los Angeles.

[Sa87] F. Saltor: Semantic Relativism in Databases: The Case of RM/T. Facultat d'Informatica RR 18/87. UPC, Barcelona.

[SCG91] F. Saltor, M. Castellanos & M. Garcia-Solaco: A Framework to Analyze Data Models as Canonical Models for Federated Databases. Dept LSI, RR. UPC, Barcelona.

[Sh81] D. Shipman: "The Functional Data Model and the Data Language DAPLEX". ACM TODS, vol 6, #1.

[SL90] A. Sheth & J. Larson: "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases". ACM Computing Surveys, vol 22, #3.

[SN88] M. Schrefl & E. Neuhold: "Object class definition by generalization using upward inheritance". Proc. 4th IEEE Int. Conf. on Data Engineering, Los Angeles.

[SP90] S. Spaccapietra & C. Parent: "View Integration: A Step Forward in Solving Structural Conflicts". LBD, DI, EPF Lausanne.

[SS91] M. Scholl & H-J. Schek: "Supporting Views in Object-Oriented Databases". IEEE-CS bulletin on Data Engineering, vol 14, #2.

[TYF86] Teorey, Yang & Fry: "A logical design methodology for relational databases using the extended entity relationship model". ACM Computing Surveys, vol 18, #2.

[vG82] J. van Griethuysen (ed): Concepts and terminology for the conceptual schema and the information base. ISO/TC97/SC5/WG3.