Feature: Telecommunications
CT2 Developments
Semantic Binary Database Model
Semantic Binary Database Model

By Nahmii Rishie

Semantic data models offer significant advantages, compared to the relational and older data models, in terms of quality and ease of database design and maintenance, data integrity, flexibility, consciousness of languages and ease of application programming. This paper proposes a variant of a semantic model which combines extensive expressive power with simplicity of use and minimalism of the arsenal of semantic constructs, as well as a concise graphical notation. As a consequence, the author believes that this model can be particularly useful to the database practitioner. The model can be used in a stand-alone database management system (DBMS) or as a design tool in top-down development of high-quality databases under the conventional DBMS in the relational, network and hierarchical models.

Semantic Binary Database Model (SBM) - a descendant of the model proposed in [Abrial-74]. This model does not have as rich an arsenal of tools for semantic description as can be found in some other semantic models, e.g., the FDE model [Abiteboul-84], the SDM model [Hamburger-91], the Functional Model [Shipman-81] and the Functional Entity-Relationship model [Chen-76]. Nevertheless, the SBM has a small set of sufficient simple tools by which all the semantic descriptors of the other models can be constructed. This makes SBM easier to use for the novice, easier to implement and usable for delineation of the common properties of the semantic models.

The SBM represents the information of an application's world as a collection of elementary facts of two types: unary facts categorising objects of the real world and binary facts establishing relationships of various kinds between pairs of objects. The graphical database schema and the integrity constraints determine what sets of facts are meaningful, i.e., can comprise an instantaneous database.

An Informal Example

The schema in Figure 1 describes some activities of a dining club, including the following information:

- Patron - any person who is a patron or waiter or both.
- Charge Account - account to which meals can be charged. There is no permanent relationship between an account and a patron. Ad hoc relationships between patrons and accounts are established when the patron's meal is charged to the account. The club does not accept cash for meals. The charge for every meal is always $10, therefore the money amounts do not appear in the database explicitly.
- Party - an ad hoc group of patrons eating on a particular date during the same shift, served by the same waiter and charging to the same account.

**Definition**

Here is a series of definitions of the SBM's concepts.

1) **Databases**
   - **Instantaneous Database** - all the information represented in a database at a given instant of time. This includes the historic information which is still kept at that time. Hence the life of a database can be seen as a sequence of instantaneous databases. The first one in the sequence is often the empty instantaneous database - it is the state before any information has been entered.
   - **Database Model** - a convention of specifying real world’s concepts in a form understandable by a DBMS.

2) **Categories**
   - **Object** - any item in the real world. It can be either a concrete object or an abstract object.
   - **Value or Concrete Object** - a printable object, such as a number, a character string or a date. A value can roughly be considered as representing itself in the computer.
   - **Abstract Object** - a non-value object in the real world. An abstract object can be a tangible item such as a person, a table, a country or an event or an idea. Abstract objects cannot be represented directly in the computer: coding and mapping schemes are needed to physically represent abstract objects in the computer. This term is also used for a user-transparent representation of such an object in the SBM.
   - **Category** - any concept of the application's real world which is a unary property of objects. At every moment in time such a concept is descriptive of a set of objects which possess the property at that time.
Unlike the mathematical notion of a set, the category itself does not depend on its objects: the objects come and go while the meaning of the category is preserved. Conversely, a set does depend on its members: the meaning of a set changes with the ebb and flow of its members.

Categories are usually named by singular nouns. For example, Person is a category of abstract objects. The set of all the patrons relevant to the application today is different from such a set tomorrow, since new patrons will arrive or will become relevant. However, the concept of Patron will remain unaltered.

An object may belong to several categories at the same time. For example, one object may be known as a person and at the same time as a waiter and as a patron.

Some of the categories in the world of our dining club are: Waiter, Person, Account, Patron, Table.

Disjoint categories - Two categories are disjoint if no object may simultaneously be a member of both categories. This means that at every point in time the sets of objects corresponding to two disjoint categories have empty intersection.

Subcategory - A category is a subcategory of another category if at every point in time every object of the former category should also belong to the latter. This means that at every point in time the set of objects corresponding to a category contains the set of objects corresponding to any subcategory of the category. For example, the category Patron is a subcategory of the category Person. The category Waiter is another subcategory of the category Person.

Abstract category - a category whose objects are always abstract.
Concrete category, category of values - a category whose objects are always concrete. Patron and Account are abstract categories. String, Number and Digit are concrete categories.

Many concrete categories such as Number, String and Boolean have constant in-time sets of objects. Thus, those concrete categories are actually indistinguishable from the corresponding sets of all numbers, all strings and the Boolean values (True, False).

Finite category - A category is finite if at no point in time an infinite set of objects may correspond to it in the application's world. Every abstract category is finite.

For example, the categories Patron, Account and Digit are finite. The category Number may be infinite.

3) Binary Relations

Binary Relation - Any concept of the application's real world which is a binary property of objects, that is, the meaning of a relationship or connection between two objects. For example, Serves is a relation relating waiters to tables. Birth-date is a relation relating persons to numbers.

The relation is descriptive of a set of pairs of objects which are related at that time. The meaning of the relation remains unaltered in time, which the sets of pairs of objects corresponding to the relation may differ from time to time when some pairs of objects cease to exist or begin to be connected by the relation.

Notation: "x R y" means that object x is related by the relation R to object y. For example, to indicate that a waiter i serves a table d, we write: \( i \text{ Serves } d \).

Types
- A binary relation \( R \) is many-to-one (\( m:1, \text{functional} \)) if at no point in time \( xRy \) and \( xRz \) where \( y \neq z \). For example, Birth-date is an \( m:1 \) relation because every person has only one date of birth.
- A binary relation \( R \) is one-to-many

Figure 1: A binary schema for a dining club application
EMERSON UPS FOR PC

Emerson, the leader in UPS protection for two decades, adds three new members to micro-UPS family for providing a reliable power protection to PC and Workstations.

MICRO ACCUPOWER UPS

Range: 150 VA, 300 VA, 500 VA, 600 VA, 800 VA, 1250 VA, and 1400 VA

SMALL WONDER UPS

Range: 1000 VA (350 VA protected by UPS & 650 VA by RH filter)

ACCUCARD

AccuCard is the first UPS that fits in a single half-size XT/AT expansion slot. In the event of power loss, AccuCard provides enough DC battery power to automatically save a snapshot of system memory to the hard disk before the power down the system. When the power comes back, all system status, registers, buffers, memory and data intact will automatically restore. Just as if nothing had happened.

Invitation. For expanding our distribution network, we invite micro-computer sellers to join us as authorized distributors. For more information, please call Liebert H.K. Marketing at 555 9302.
(1:m) if at no point in time \( xRy \) and \( zRy \) where \( x \neq z \).
- Relations which are of neither of the above types are called proper many-to-many (m:m). For example, Serves is a proper m:m relation because a waiter can work in many tables and a table may employ many waiters.
- A binary relation which is both m:1 and 1:m (always) is called one-to-one (1:1). For example, if accounts are identified by their names, then the Account-Name relation is 1:1, meaning that every account has at most one name, and no character string is the name of two different accounts.

Suppose that in the current situation in our real world, the following is true: every person has at most one name and no two persons have the same name.

This does not mean that Name is a 1:1 relation between persons and strings. Name would be a 1:1 relation if the above condition were true at all times: past, present and future.
- A binary relation is proper m:1 if it is m:1 and not 1:1.
- A binary relation is proper 1:m if it is 1:m and not 1:1.

For example, all of the types of relations mentioned in the previous example are proper.

Since the Account-Name is 1:1, it is also 1:m, m:1 and m:m. Since this relation is proper 1:1, it cannot be proper 1:m, m:1 or m:m.

Figure 2 shows the classification of all relations.

- **Domain and range** - A category \( C \) is the domain of \( R \) if it satisfies the following two conditions:
  i) whenever \( xRy \) then \( x \) belongs to \( C \) (at every point in time for every pair of objects); and
  ii) no proper subcategory of \( C \) satisfies condition (i).

A category \( D \) is the range of \( R \) if:
  i) whenever \( xRy \) then \( y \) belongs to \( D \) (at every point in time for every pair of objects); and
  ii) no proper subcategory of \( D \) satisfies condition (i).

For example, the domain of Account-Name is the category Account and its range is the category String. The domain of Serves is Waiter and the range is Table.

- **Total binary relation** - A relation \( R \) whose domain is \( C \) is total if at all times for every object \( x \) in \( C \) there exists an object \( y \) such that \( xRy \). (At different times, different objects \( y \) may be related to a given object \( x \).

**Note:** Most relations are not total on their domains. For example, though the domain of the relation Birth-Date is the category Person, the date of birth of some relevant persons is irrelevant or unknown. Thus, the relation Birth-Date is not total.

4) **Non-binary Relationships**

- **Non-binary relationships** - Real-world relationships that bind more than two objects in different roles. For example, there is a relationship between a waiter, an account and a shift during which the waiter works for the account.

Such complex relationships are regarded in the SBM as groups of several simple relationships.

For example, the non-binary relationship of the previous example is represented in the SBM by a fourth object, a party and three binary relations between the party and the waiter and the account.

In general, the SBM represents any non-binary relation as:
  i) An abstract category of events. Each event symbolises the existence of a relationship between a group of objects.
  ii) Functional binary relations, whose domain is the category (i). Each of those functional binary relations corresponds to a role played by some objects in the non-binary relation.

Thus, the fact that objects \( x, \ldots, x \) participate in an n-ary relation \( R \) in roles \( R_1, \ldots, R_n \), is represented by:

- an object \( e \) in the category \( R' \),
- binary relationships \( eR_1x, \ldots, eR_nx \).

5) **Instantaneous Databases**

- **Formal representation of an instantaneous binary database** - as a set of facts, unary and binary.
Uses of the Model
The SBM can be used as a semantic database management system and as a database design tool in conjunction with conventional DBMSs.

*Semantic DBMS*
We have developed the model as an experimental semantic DBMS [Rishe-88-EOI, Rishe-88-TMI, Vijaykumar-87], [Jain-87]. The semantic DBMS is capable of rendering better services to its users than services offered by the relational and other conventional DBMSs, in terms of ease of database design, flexibility, ease of application programming in fourth-generation languages and in non-procedural languages and enforcement of integrity constraints. This is not surprising because a system that knows more about the semantics of the user’s data can certainly offer better services. In addition, our experimental implementation has shown that the semantic models have potential for much more efficient implementation than the conventional data models. This is due to two reasons:

- All the physical aspects of representation of information by data are user-transparent in the semantic models. This creates greater potential for optimization: more things may be changed for efficiency considerations without affecting the user programs. This is a continuation of a trend in the relational model. The relational model has more data independence than older models. For example, the order of rows in the tables (relations) is transparent to the user. The semantic models have even more user-transparent structure.

- For example, the representation of real-world entities by printable values is transparent to the user. One may recall that not long ago the relational model was criticized as less efficient than the network and hierarchical models. However, it is clear now that optimizing relational database systems have potential of much higher efficiency than the network and hierarchical systems due to its data independence.

In semantic models, the system knows more about the meaning of users’ data as well as connections between such data. This knowledge can be utilized to organize the data so that meaningful operations can be performed faster at the expense of less meaningful or meaningless operations.

Currently we are working on an experimental massively-parallel database machine to support the SBM [Rishe-88-AM].

*Use in logical design of conventional databases*
We have developed a methodology for logical design of relational, network and hierarchical schemas and integrity constraints using semantic binary schemas [Rishe-88-DDF], [Rishe-88-MT]. This is a top-down methodology. In this methodology, a conceptual design of an enterprise is designed using a SBM. Then this description is converted into the relational database design. The result is a high-quality conventional database schema and integrity constraints.

We have also developed a tool which automates all the busy work of the methodology and provides graphic output [Rishe-88-MT], [Jain-87]. With respect to the intelligent design decisions, the tool accepts instructions from its user who is a database designer, or, when the user defaults, makes decisions itself based on “rule-of-thumb” principles.

The author gratefully acknowledges the advice of Li Qiang, Nagaraian Prabakaran, Doron Tal and David Barton.

Bibliography

| [Rishe-88-TMI] N Rishe, “A Methodology and Tool for Top-down Relational Database Design” Submitted to Knowledge and Data Engineering. |