87-RM

Mathematical Modelling in Science and Technology

THE FIFTH INTERNATIONAL CONFERENCE

Berkeley, California, USA July 1985

Edited by

Xavier J.R. Avula
University of Missouri-Rolla

George Leitmann
University of California,
Berkeley

C.D. Mote, Jr.
University of California,
Berkeley

Ervin Y. Rodin
Washington University

PERGAMON JOURNALS LIMITED

A Member of the BPCC PLC Group of Companies

ON REPRESENTATION OF MEDICAL KNOWLEDGE BY A BINARY DATA MODEL

Naphtali Rishe

Computer Science Department, University of California, Santa Barbara, Ca 93106, USA.

Abstract. A binary data model is shown to be most suitable for representation of medical knowledge, such as description of diseases, symptoms, drugs, drug interaction. The knowledge is represented by a lattice of categories of abstract objects (some of which are included in other categories or intersect them), binary relations among objects of these categories and non-total attributes of these objects. The representation is very flexible for changes both in particular knowledge and in general concepts of knowledge. The model accommodates for uncertainty, conditionality, and incompleteness of information.

Keywords. Data bases, semantic modeling, medical data, binary model, categories, relations.

INTRODUCTION

This paper introduces a binary data model, which is a tool to describe simply and concisely the relationships between the concepts of any universe of discourse or of users' real world's information. A simple medical application is described as case study.

The model can be used to document the conceptual framework of a subject, whether computerized data processing is needed or not. In the former case the model can be used to produce a data base.

The binary data base model represents information of an application's world as a collection of elementary facts of two types: unary facts categorizing objects of the real world and binary facts establishing relationships of various kinds between pairs of objects.

The purpose of the model is to provide a means of simple natural data-independent flexible and non-redundant specification of information emphasizing its semantic aspects.

A variant of the binary model was first introduced in [Abrial-1974]. Since then several other variants appeared in the literature. The concepts of the binary model are close to those of the functional model as presented *e.g.* in [Shipman-1981]. Formal mathematical semantics of the binary model is defined in [Rishe-1985] using the methodology proposed in [Rishe-1986b].

Other major data base models — relational, network, and hierarchic — fail to describe adequately the semantics of a complicated real world. They model the representation of information by computer data instead of modeling the actual information of the real world. If a data base management system supporting the binary model is not available, the binary model can be used to produce documentation and conceptual interfaces between the real world and the technicalities of the data base management in the other models. For this purpose algorithms were proposed in [Rishe-1984a] to translate an arbitrary binary schema into equivalent relational, network, and hierarchic schemata, so that the latter are usually of highest semantic quality possible in the corresponding models for the given application world.

User-friendly languages to manipulate and query the information described by the binary model are proposed in [Rishe-1986a] and [Rishe-1984b].

DEFINITIONS

Data Base — a storage of *complete* information of an application's world *and* managing software, concealing from the users the physical aspects of information storage and information representation.

Schema — an outline of the information structure of an application's world. It can be the data structure of a data base.

Object — Any item in the real world. It can be either a value or an abstract object as follows:

Value — a number or a character string or a date *etc.*

Abstract Object — A non-value object in the real world, *i.e.* a material item (*e.g.* a person, a table, a country), or an event (*e.g.* a sickness of a patient), or an idea (*e.g.* a disease) *etc.*. Abstract objects cannot be represented directly in the computer.

Category — Any concept of the application's real world which is a unary property of objects. At every moment in time such concept is descriptive of a set of objects which posses the property at that time. The category itself does not depend on its objects: the objects come and go while the meaning of the category is preserved in time. Categories are usually named by *singular* nouns.

For example, *PATIENT* is a category of abstract objects. The set of all the patients relevant to the application today is different from such a set tomorrow, since new patients will arrive or will become relevant. However the concept *PATIENT* will remain unaltered.

An object may belong to several categories at the same time. *E.g.* one object may be known as a person, and a patient, and a physician.

A category is a **subcategory** of another if at every point in time every object of the former category should also belong to the latter. For example, the category *PATIENT* is a subcategory of the category *PERSON*.

Binary Relation — Any concept of the application's real world which is a binary property of objects, *i.e.,* the meaning of relationship or connection between two objects. At every moment in time such concept is descriptive of a set of pairs of objects which possess the property at that time. The meaning of a relation remains unaltered in time, while the sets of pairs of objects corresponding to the relation may differ from time to time, *i.e.*, some pairs of objects may cease or begin to be connected by the relation.

Examples: *TAKES* is a relation relating persons to drugs. *MAY-CURE* relates drugs to diseases. *NAME* is a relation relating persons to character strings. *BIRTH-DATE* is a relation relating persons to dates.

Notation: $x\ R\ y$ means that the object x is related by the relation R to y. *E.g.* to indicate that a patient p takes a drug d we write p *TAKES* d.

A binary relation is many-to-one (m:1) if at no point in time xRy and zRy where $x \neq z$. For example *BIRTH–DATE* is a m:1 relation since nobody was born twice.

A binary relation is one-to-many (1:m) if at no point in time xRy and xRz where $y \neq z$.

Relations which are of neither of the above types are called proper many-to-many (m:m). For example, one patient may take several drugs and one drug may be taken by several patients. Thus the relation *TAKES* is m:m.

Domain/range of a binary relation: A category C is the domain of R if whenever xRy then x belongs to C (at every point in time for every pair of objects), and no proper subcategory of C satisfies this property. A category C is the range of R if whenever xRy then y belongs to C (at every point in time for every pair of objects), and no proper subcategory of C satisfies this property.

No relation needs be *total* on its domain. For example, though the domain of the relation *BIRTH-DATE* is the category *PERSON*, for some relevant persons the date of birth is irrelevant or unknown or does not exist (for unborn children.)

Non-binary relationships — real-world relationships that bind more than two objects in different roles. For example, there is a relationship between a patient, a disease, the time when the patient had the disease, *etc*. Such complex relationships are regarded in the Binary Model as groups of several simple relationships. For the above example we consider an event belonging to the category *SICKNESS*, to which the patient is related by *HAD*, the disease is related by *OCCURRED-AS*, *etc*.

In a schema categories of abstract objects are shown by rectangles. Relations between abstract objects are shown by arrows between the categories' rectangles. (The directions of the arrows are from the domains to the ranges). Relations between abstract objects and values are listed in the rectangles of their domains. Subcategories' rectangles are connected to their supercategories' rectangles by dashed lines. If two categories may have objects in common, but neither is a subcategory of the other, then their rectangles in the schema are connected by <- - -∩- - ->.

A SIMPLE MEDICAL APPLICATION

As a case study, the information aspects listed hereinafter are modeled by a schema depicted in Fig. 1. The concepts incorporated in the binary schema are marked with "☞".

1. A catalogue of names of known diseases.
   - ☞ Category *DISEASE*
   - ☞ Relation ***name*** from *DISEASE* to the category of values *Text* (*m:1*)
2. A catalogue of descriptions of known symptoms:
   - their names and
   - the units in which the magnitude of their intensity/acuteness is measured.
   - ☞ Category *SYMPTOM-TYPE*
   - ☞ Relation ***name*** from *SYMPTOM-TYPE* to the category of values *Text* (*m:1*)
   - ☞ Relation ***magnitude-unit*** from *SYMPTOM-TYPE* to the category of values *Text* (*m:1*)
3. For every disease there is a list of its possible symptoms, in which:

   for every possible symptom

   for some magnitudes of its acuteness

   there is a probability estimation whether

   the symptom should accompany the disease with such magnitude at least.
   - ☞ Category *SYMPTOM'S-POSSIBILITY-FOR-A-DISEASE* (Every object of this category is an event of the possibility of a symptom for a disease.)
   - ☞ Relation **may-have** from *DISEASE* to *SYMPTOM'S-POSSIBILITY-FOR-A-DISEASE* (*1:m*)
   - ☞ Relation **may-indicate** from *SYMPTOM-TYPE* to *SYMPTOM'S-POSSIBILITY-FOR-A-DISEASE* (*1:m*)
   - ☞ Relation ***magnitude*** from *SYMPTOM'S-POSSIBILITY-FOR-A-DISEASE* to the category of values *Number* (*m:1*)
   - ☞ Relation ***probability*** from *SYMPTOM'S-POSSIBILITY-FOR-A-DISEASE* to the category of values *0-100%* (*m:1*)
4. A catalogue of names of known drugs.
   - ☞ Category *DRUG*
   - ☞ Relation ***name*** from *DRUG* to the category of values *Text* (*m:1*)
5. For every disease there are lists of factors which may aggravate, cause or cure the disease:

   drugs, drug combinations, other diseases, contagious patients.
   - ☞ Category *FACTOR-INFLUENCING-DISEASES*
   - ☞ Subcategory *DRUG* of the category *FACTOR-INFLUENCING-DISEASES*
   - ☞ Subcategory *DISEASE* of the category *FACTOR-INFLUENCING-DISEASES*
   - ☞ Subcategory *PATIENT'S-SICKNESS* of the category *FACTOR-INFLUENCING-DISEASES* (The category PATIENT'S-SICKNESS will be elaborated later.)
   - ☞ Subcategory *DRUG-INTERACTION* of the category *FACTOR-INFLUENCING-DISEASES* (Every object of the category DRUG-INTERACTION stands for a combination of drugs which jointly can produce influence.)
   - ☞ Relation **participates** from *DRUG* to *DRUG-INTERACTION* (*m:m*)
   - ☞ Relation **may-cure** from *FACTOR-INFLUENCING-DISEASES* to *DISEASE* (*m:m*)
   - ☞ Relation **may-aggravate** from *FACTOR-INFLUENCING-DISEASES* to *DISEASE* (*m:m*)
   - ☞ Relation **may-cause** from *FACTOR-INFLUENCING-DISEASES* to *DISEASE*

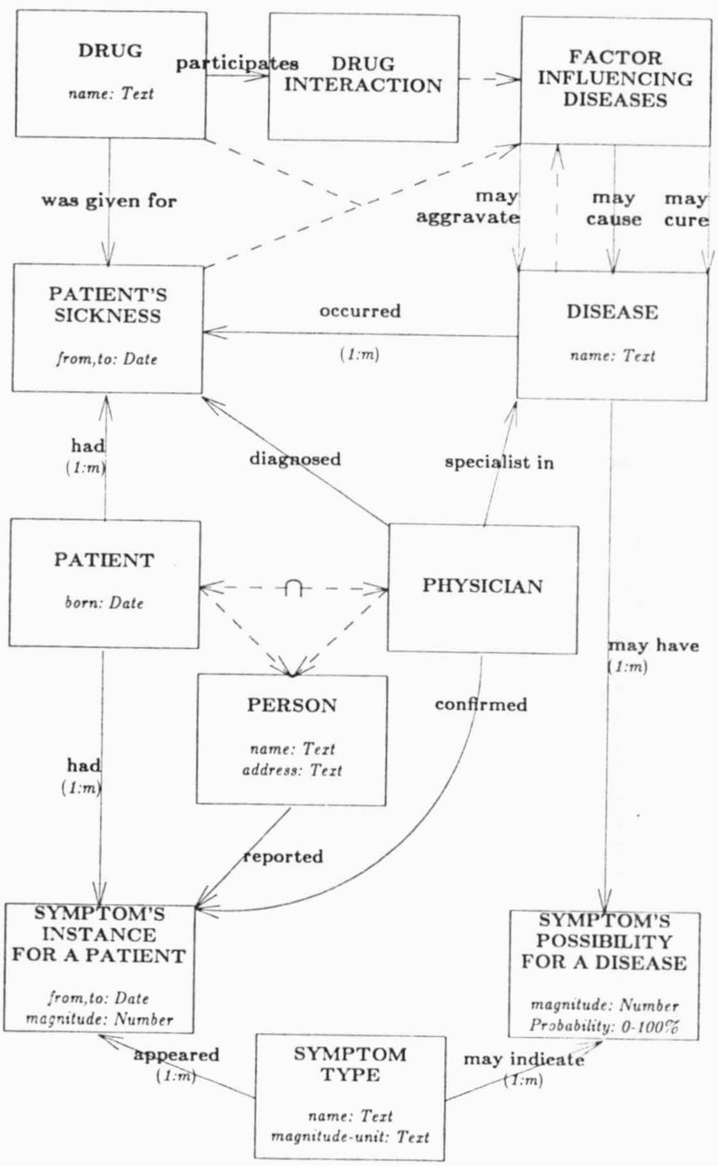

FIG 1. A binary schema for a medical application.

(*m:m*)

6. Names, addresses, and dates of birth of patients; names and addresses of physicians. (Some physicians are also known as patients.)
   - ☞ Category *PERSON*
   - ☞ Relation ***name*** from *PERSON* to the category of values *Text* (*m:1*)
   - ☞ Relation ***address*** from *PERSON* to the category of values *Text* (*m:1*)
   - ☞ Subcategory *PATIENT* of the category *PERSON*
   - ☞ Subcategory *PHYSICIAN* of the category *PERSON*
   - ☞ Relation ***born*** from *PATIENT* to the category of values *Date* (*m:1*)
7. Physicians' areas of specialization (diseases).
   - ☞ Relation **specializes-in** from *PHYSICIAN* to *DISEASE* (*m:m*)
8. Every patient's medical history, including:
   - all his/her present and past illnesses,
     - their duration,
     - their diagnosing physicians,
     - drugs prescribed for them;
   - ☞ Category *PATIENT'S-SICKNESS* (Every object of this category is an event of a patient having a disease during a period of time.)
   - ☞ Relation **had** from *PATIENT* to *PATIENT'S-SICKNESS* (*1:m*)
   - ☞ Relation **occurred** from *DISEASE* to *PATIENT'S-SICKNESS* (*1:m*)
   - ☞ Relation ***from*** from *PATIENT'S-SICKNESS* to the category of values *Date* (*m:1*)
   - ☞ Relation ***to*** from *PATIENT'S-SICKNESS* to the category of values *Date* (*m:1*)
   - ☞ Relation **diagnosed** from *PHYSICIAN* to *PATIENT'S-SICKNESS* (*m:m*)
   - all his/her reported symptoms with
     - the duration of the symptom's occurrences,
     - an indication of the magnitude of intensity/acuteness of the symptom's occurrence,
     - a record of the persons (names and addresses) who reported or measured the symptom's occurrence (*e.g.* the patient himself, his relatives, medical personnel, *etc.*),
     - and physicians who confirmed the symptom's occurrence.
   - ☞ Category *SYMPTOM'S-INSTANCE-FOR-A-PATIENT* (Every object of this category is an event of a a patient having a certain symptom with a certain magnitude during a certain period of time.)
   - ☞ Relation **had** from *PATIENT* to *SYMPTOM'S-INSTANCE-FOR-A-PATIENT* (*1:m*)
   - ☞ Relation **appeared** from *SYMPTOM-TYPE* to *SYMPTOM'S-INSTANCE-FOR-A-PATIENT* (*1:m*)
   - ☞ Relation ***from*** from *SYMPTOM'S-INSTANCE-FOR-A-PATIENT* to the category of values *Date* (*m:1*)
   - ☞ Relation ***to*** from *SYMPTOM'S-INSTANCE-FOR-A-PATIENT* to the category of values *Date* (*m:1*)
   - ☞ Relation ***magnitude*** from *SYMPTOM'S-INSTANCE-FOR-A-PATIENT* to the category of values *Number* (*m:1*)
   - ☞ Relation **reported** from *PERSON* to *SYMPTOM'S-INSTANCE-FOR-A-PATIENT* (*m:m*)
   - ☞ Relation **confirmed** from *PHYSICIAN* to *SYMPTOM'S-INSTANCE-FOR-A-PATIENT* (*m:m*)

DISCUSSION

One of the major advantages of the relational data base model, as compared to the network and hierarchic models, was the independence of the logical *data* from the physical aspects of data storage. The binary model went one step forward towards the independence of the the actual *information* from its logical data representation. Among the semantic advantages of the binary model relative to the relational model are the following:

- All the information is composed of the elementary facts describing the real world, so no normalization of a binary schema is needed;
- No category of objects needs to have a key, *i.e.* a collection of attributes which are never *null* and which universally identify the objects of the category. (Instead, different objects of the category may be identifiable by different attributes or by different relationships with objects of perhaps other categories. In the real uncomputerized world, keys almost never exist.)
- Objects are not logically replaced by their keys, when these exist. So a value of a key is changeable with no influence on the other information about this object in the data base.
- An object may belong to several categories simultaneously.
- Properties which are common to several categories, can be specified just once.
- It is conceptually simple and schemata can be easily explained to owners of the information to be stored in the data base, who may have no computer knowledge but must approve the conceptual schema.

REFERENCES

Abrial, J.R. (1974). Data Semantics. In J.W. Klimbie and K.L. Koffeman (eds.), *Data Base Management,* North Holland.

Rishe, N. (1984a). Algorithms for Design of Relational, Network and Hierarchic Schemata from Conceptual Binary Schemata. Tech. Rep. TRCS84-15, University of California, Santa Barbara.

Rishe, N. (1984b). Data Base Design Fundamentals: Concepts and a Case Study. Tech. Rep. TRCS84-16, University of California, Santa Barbara.

Rishe, N. (1985). Semantics of Universal Languages and Information Structures in Data Bases. Tech. Rep. TRCS85-10, University of California, Santa Barbara.

Rishe, N. (1986a). Postconditional Semantics of Data Base Queries. *Mathematical Foundations of Programming Semantics.* Springer-Verlag. (In press.)

Rishe, N. (1986b). On Denotational Semantics of Data Bases. *Mathematical Foundations of Programming Semantics.* Springer-Verlag. (In press.)

Shipman, D.W. (1981). The Functional Data Model and the Data Language DAPLEX, *ACM Transactions on Database Systems,* v. *6,* no. 1, 140-173