14 IIPS FA 1.13

A SEMANTIC DATA MODEL FOR ATMOSPHERIC OBSERVATIONS, STORMTRACK DATA, AND WIND ANALYSES

Nirva Morisseau-Leroy* , Mark D. Powell, Sam Houston, and James C. Hendee

1. INTRODUCTION

A database is a model of the real world system. A database is more than just a collection of values, it is a structured collection of objects (elements). The complexity of databases arises from complicated structure and behavioral properties that change through time: concurrent, interactive access by users with different processing needs over a large shared database; and multiple and distinct database views (external schemas).

Conventional data models are not satisfactory for modeling database applications. The features that they provide are too low-level and representational to allow the semantics of a database to be directly expressed in the schema. Semantic data models emerged from a requirement for more conceptual data models. The goal of semantic data models is the design of a higher level data model, which enables the database designer to naturally and directly incorporate in the schema a large portion of the semantics of the database. A database whose organization is based on naturally occurring structures, which is described to the user in terms of concepts familiar to him/her, should make it easy for the user to understand it and use it. Semantic data models attempt to produce a representation of the database that closely parallels the user's perception, a model that serves as a natural application modeling to capture and express the structure of the problem domain. The user's perception plays a major role in the evaluation of the model for a given database.

For any database system, any information system, there are two levels of design: logical and physical. A logical design consists of integrating the requirements of a number of users to arrive to a centrally controlled and maintained logical database structure. The central structure, the schema, must support individual user views of the data and support their processing needs. The physical database design involves the evaluation of possibilities for the implementation of the logical design, a choice of storage structure, placement strategies, and query mechanisms, etc.

This paper details the logical or conceptual model, describes a particular data model, called the Semantic Binary Model (SBM) [10], and summarizes the mapping of the logical design, the implementation of such design, into a physical design using an object-relational database.

The paper is in six sections. Section two provides the fundamental concepts and terminology needed in the discussion of logical and conceptual design. Section three outlines the Semantic Model. Section four provides the notation of a semantic model: the Semantic Binary Model. WANDA schema is presented in section five. An efficient implementation of the logical design is presented in section six.

2. FUNDAMENTALS

At every point of time, the content of a database represents a snapshot of the state of an application system. The state of a database model, at a given instance, represents the knowledge it has acquired from this world. The changes to the database over time reflect the sequence of events occurring in the application environment. The structure or the schema of the database model is a representation of the entities or objects (i.e., the elements) and the interconnections (relationships) between theses objects within this piece of the real world. The structure of a database must mirror the structure of the system that it models. Database models have structures for modeling operations used to manipulate the objects of the database schema. These structures can be atomic operations or more complex transactions. There are, basically, three database modeling levels [8]:

1. External Level: The external level describes the user's logical view of the database without consideration for performance or storage issues.
2. Conceptual Level: This is the level that provides the mapping from the logical to the physical and describes the semantic of the objects and relationships, including descriptions of connections and consistency constraints.
3. Internal Level: An abstract model of the physical database.

3. SEMANTIC MODELING

Prior to the 1970s, early database research concentrated on the physical structure of databases. Little consideration was given to

* Corresponding author address: Nirva Morisseau-Leroy, NOAA/AOML/HRD, 4301 Rickenbacker Causeway, Miami, Florida, 33149; email: moriss@aoml.noaa.gov

the user's perception of the data. The past decade has seen the emergence of numerous data models whose aim is to provide increased expressiveness to the designer and richer set of semantics into the database. These data models have been categorized as "semantic" data models; their unifying characteristics is to provide more semantic content than the current data models. Two important ideas signaled the emergence of semantic data models [9]:

1. Data independence and
2. the capture of additional semantics in the data modeling process.

Existing models were capable of defining some data semantic [1][4]. For example, functional dependencies [4] from the relational theory established some lower level semantics for data models. The entity-relationship (E-R) model [2] is an early semantic data model that unifies the features of the traditional models to facilitate the incorporation of semantic information. From a conceptual perspective, the database is viewed as a collection of entity and relationship types. The E-R model provides strong support for a multiplicity of constraints: Cardinality (many-to-one, many-to-many, and one-to-one relationships, etc.), insertion and deletion (defined on existence dependencies). E-R was the one of the first semantic data model attempting to provide multiple abstraction levels by combining the best features of the earlier models such as the network and the relational model.

The main objective of semantic data models is to facilitate the modeling and the use of databases. Semantic data models, through the use of abstraction permit the user to view the data at multiple levels. The principle of abstraction is the suppression of some detail in order to emphasize more appropriate detail. Abstraction provides an approach to reasoning about databases and their development. A data abstraction defines the structure of an object and actions which provide the only means of altering the object. Semantic Models make extensive use of abstraction to address two main problems in the design, development, and evolution of information systems:

- Managing complexity.
- Defining and ensuring a high degree of semantic integrity.

Semantic model is a modeling methodology for the design and development of database-intensive applications in which structural (state or static, entities and their relationship) and behavioral (process or dynamic, state transitions, dynamic properties of operations and their relationships) properties are treated explicitly and abstractly. It provides to the user relationships between data objects that support the manner in which the user perceives the real-world and to the designer the mechanisms, the tools to construct a particular database schema without the need to implement on a low-level. Some of the advantages of the semantic data model are:

1. User's ability to understand the semantics of the modeling constructs provided.
2. Ease of search (query) formulation.
3. Ease of specification and maintenance of the semantics of the modeling constructs.

This paper reports on the Semantic Binary Model [10] and uses this model to capture the structure of the WANDA database.

4. SEMANTIC BINARY MODEL (SBM)

The SBM has been designed as a natural application modeling mechanism that can capture and express the structure (schema) of an application environment. It can serve as a formal specification and document mechanism for a database and can be used as a tool in the database design process. SBM is a high level user oriented data model that has been designed to provide a basis for effective user views of, and associated user interface to databases. In this paper, SBM is presented as a tool for database modeling, which is used to improve the understandability and accessibility of databases, in particular, the WANDA database. The database supporting this application is used to keep track of atmospheric observations, stormtrack data, and wind analyses.

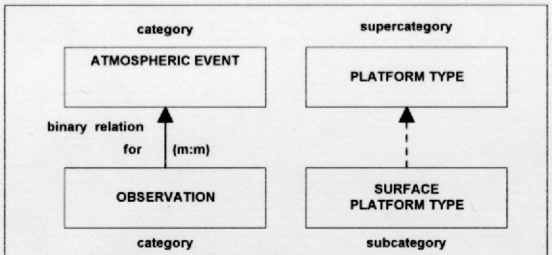
4.1 Semantic Binary Model Notation

The principal objective of database design is to extract as many structural and behavioral properties as possible from the problem domain to define conceptual and external schemas. The design process is composed of discrete steps, many pass processes, each of which deals with distinct aspects of the design and development process, thus, the necessity to design and develop using levels of abstractions. Object schemes such as data structure diagrams, entity-relationship (E-R) can be used to graphically represent objects and structural relationships of database applications. An object scheme is a directed graph in which nodes are strings (integers, real, etc.) denoting objects and edges identifying relationships between objects [1]. The semantic of an object is completely defined by its behavioral and structural properties. Thus, the separation of structure and behavioral can severely complicate the design specification and the semantic integrity analysis. A behavior scheme for an action or transaction is an object scheme which includes each object in the scope plus an operation level on each edge, one label for each constituent action. Using behavior schemes, the design of gross behavioral properties of actions can be done explicitly and abstractly taking advantage of modularity [7]. The result of the combination of object and behavioral schemes produces a conceptual model of the application, a network of data abstractions.

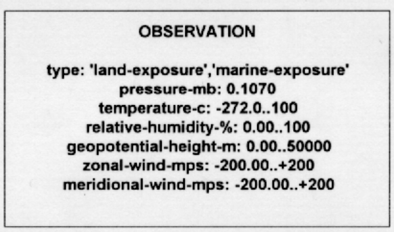
For the design and specification of structural properties of database applications, SBM provides a rich set of modeling constructs: category (abstract and concrete), object (abstract and concrete), binary relations, and other forms of abstractions for relating objects. SBM is more than an ancillary to the existing models (network and relational model). It is an independent model rather than a vehicle for designing relational databases. SBM graphic notation of the forms of abstractions are given in figure 1. SBM categorization of objects is as follows [11]:

- Value of concrete object: A printable object such as a number, a string. For example, latitude is a concrete object. In this example, latitude is also an attribute. An attribute is a functional relation (i.e., many-to-one, or one-to-one) whose range is a concrete category (Figure 2).

- Abstract object: A non-printable object. Such an object can be an item (storm Andrew), or an event (wind analysis of a tropical system), or an idea (WANDA application).
- Category: A category is a precise characterization of all properties shared by each object in the collection. An object category is a catalog, an aggregation, a generalization of a set of object properties. OBSERVATION is an example of a category. OBSERVATION category is the domain for the attributes of an atmospheric observation. The values that constitute this domain are specified via a data type such as Integer, Number, String, and so on. Domain constraints (latitude constraint -90.0000 , + 90.0000) are constraints on the type of values an attribute can have. A category name (written in upper case) identifies a category. Two categories may be disjoint or they may intersect.



SBM Notation: Figure 1



SBM Notation: category with attributes
Figure 2

- Subcategory: All objects of a subcategory belong to another category (the supercategory). PLATFORM-TYPE is an example of a supercategory and SURFACE-PLATFORM-TYPE is a subcategory. SBM uses two means to handle repeated information within the database schema: subcategory, derivation of properties from a supercategory, and semantic connections, thus limiting the degree of redundancy. Specification of constraints is also provided; it is the physical and operational interpretation of the semantic.
- At every moment of time, one needs to know the relation between a set of pairs of objects, i.e., how two objects relate. In the SBM, binary relations express the connection between a pair of objects. A binary relation (Figure 1) is a set of ordered pairs. The first component of the pair identifies an object (element) in the domain set and the second an object in the range set. Objects and categories can be connected by non-binary relations.

5. CONCEPTUAL MODEL - WANDA SCHEMA

SBM has been designed to enable a computerized database to directly model an application environment by supporting database structures that are in close correspondence with the natural constructs of the application environment. Our work focuses on the problems of increasing the understandability of a database and allowing it to be more accessible to the meteorological scientists at NOAA/AOML/HRD.

A conceptual data model, a schema (an aggregate schema facility), is a collection of definitions of permissible assertions; a collection of rules and constraints which governs how assertions are related and what may be asserted. SBM contains three main types of concepts: object type, relationship type, and a data element type. These concepts can however be elementary, derived, generalized or grouped. Conceptual data modeling is the main technique used in requirements specification and decisive in discussions with users. The conceptual data model is also the dominant factor controlling the user interface of a system. In the SBM, a user view is represented in terms of categories, associations, attributes and relations in a view diagram (Figure 3).

5.1 User Perspective and Database Designer Perspective

At NOAA/AOML information needs fall under two major groups: 1) Atmospheric observation information sharing between NOAA laboratory scientists and 2) Data set information sharing between world scientists and the general public.

Before the system could be designed and implemented two major tasks faced the designer:

1. The identification of which data was to be collected.
2. The standardization of that information wherever possible.

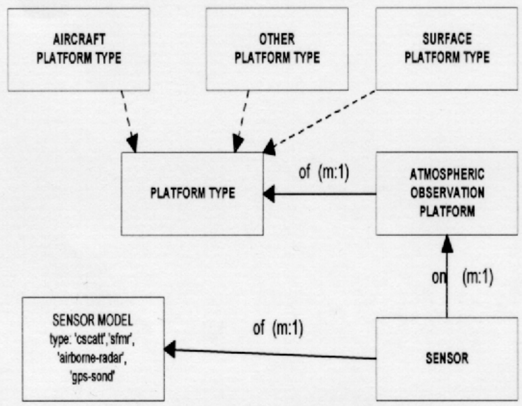
Requirement analysis provides the input to all phases of the logical design process. The input consists of a specification of the data requirements and the processing requirements of the potential users of the system. Using the requirement specifications as input, user views of the real world are extracted as explicitly as possible. The design process involves many pass processes. The designer must interact with the design process to select an appropriate solution.

Data structure diagrams from the SBM are used because they show abstractly and graphically the relationships between objects, the component elements of user views. The process of view modeling involves the extraction from the user the relevant parts of the real world information and the abstraction of this information into a form which completely represents the user view. SBM represents a user view as explicitly as possible in the following sense:

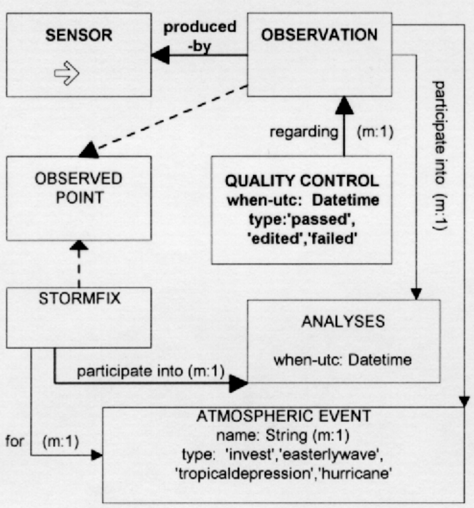
- Distinction among different kinds of associations between objects, categories.
- Allow the associations in which objects or associations or a combination of the two can participate.
- Incorporation of the dependence of categories on one another for the sake of identification.

- The effect of insertion and deletion of objects and relations on one another at the schema level.
- Incorporation of user-defined rules about instances of data.

Transactions are modeled. Structure modeling at the transaction level involves identifying objects and relationships needed by a transaction. For example, an "arrival" transaction



SBM Subschema: ATMOSPHERIC PLATFORM
fIGURE 3



SBM subschema: OBSERVATION
Figure 3 (continued)

⇨ indicates category defined in another schema

represents the event by which an observation becomes known to the database system. It also provides the operational link between PLATFORM-TYPE (for example, satellite), ATMOSPHERIC-EVENT (storm Andrew), and OBSERVATION (an observed point described in terms of latitude, longitude, date time, etc.).

5.2 Data Dictionary

A data dictionary contains all the names of the categories, objects, and relations of the application's schema, the structure of the schema, the definition of all objects of the application. The specification of every concept in the Semantic Binary Model (SBM) consists of:

- Concept name: OBSERVATION is the concept's name.
- Technical characteristics of the concept: Category is the technical characteristic of the concept OBSERVATION.
- Comment: A comment defines the meaning of the concept. For example, "(A catalog of persistent sources of platform observations: virtual, physical, etc.)".

This paper presents the data dictionary for the Atmospheric Platform and the Observation subschemas.

5.3 Atmospheric Platform Subschema

⇒ ATMOSPHERIC OBSERVATION PLATFORM – category (A catalog of persistent sources of platform observations: virtual, physical, etc.)

⇒ PLATFORM-TYPE – category (A catalog of persistent source of platform type)

⇒ SURFACE-PLATFORM-TYPE – subcategory of PLATFORM-TYPE (A physical observation platform that collects observations in the near surface layer)

⇒ OTHER-PLATFORM-TYPE – subcategory of PLATFORM-TYPE (A catalog of other platform types)

⇒ AIRCRAFT-PLATFORM-TYPE – subcategory of PLATFORM-TYPE (A physical observation platform that collects observations at the flight level, multiple levels, and/or the surface layer)

⇒ SENSOR – category (A catalog of instruments that collects observations. These instruments are located on physical observation platforms)

⇒ SENSOR-MODEL-TYPE – category (A catalog of sensor model types. For example, a buoy sensor that collects wind observation data is of sensor model type: buoy cup anemometer)

These attributes below are integral parts of the schemas, but all of them are not represented in the semantic schemas.

⇒ type – attribute of PLATFORM-TYPE, range string (m:m) (An English name to identify the type of a specific platform)

⇒ type – attribute of SENSOR-MODEL (An English name for a sensor model)

⇒ on – relation from SENSOR to ATMOSPHERIC-OBSERVATION-PLATFORM (m:1) (A sensor is located on a specific platform)

⇒ of – relation from SENSOR to SENSOR-MODEL (m:1)

⇒ of – relation from ATMOSPHERIC-OBSERVATION-PLATFORM (m:1) (A platform of a specific type)

5.4 Observation Subschema

- ⇒ OBSERVATION – subcategory of OBSERVED-POINT(An observation object. A report by a platform or by an instrument on a platform)
- ⇒ SENSOR – See subschema ATMOSPHERIC PLATFORM
- ⇒ STORMFIX – subcategory of OBSERVED-POINT (An atmospheric event of type stormfix. A stormfix is the position and time of an atmospheric event; for example, at the center of a storm, this is the location where the wind is = 0 at that point and time.)
- ⇒ ATMOSPHERIC-EVENT – category (A catalog of atmospheric events, e.g., storm Andrew, Dolly, etc.)
- ⇒ QUALITY-CONTROL – category (Observations undergo quality control. All determining actions (e.g., failed, passed, edited) constitute a quality control event.
- ⇒ OBSERVED-POINT – category (A catalog of coordinates for observations and stormfixes.)
- ⇒ when-utc – attribute of OBSERVED-POINT, range: Datetime, total "(Timestamp of an observed point).
- ⇒ latitude-degree – attribute of OBSERVED-POINT.
- ⇒ longitude-degree – attribute of OBSERVED-POINT.
- ⇒ pressure-mb -- attribute of OBSERVATION.
- ⇒ wind-speed-mps -- attribute of OBSERVATION.
- ⇒ name – attribute of ATMOSPHERIC-EVENT (A name given to a tropical system).
- ⇒ type – attribute of ATMOSPHERIC-EVENT, String (The tropical system name/identifier.)
- ⇒ when-utc – attribute of ANALYSES .
- ⇒ analysis-exposure-type – attribute of ANALYSES (An analysis for land or marine)
- ⇒ mode – attribute of ANALYSES (An analysis can be done at research or operational mode)
- ⇒ for – relation between OBSERVATION and ATMOSPHERIC-EVENT (m:m) (Many observations for many tropical systems)
- ⇒ for – relation between OBSERVATION and ATMOSPHERIC-EVENT (m:m) (Many stormfixes for many tropical systems)
- ⇒ participate-into – relation between OBSERVATION and ANALYSES (m:m) (Many observations participate into an analysis)
- ⇒ participate-into – relation between STORMFIX and ANALYSES (m:m) (Many stormfixes participate into an analysis)

6. PHYSICAL DESIGN - IMPLEMENTATION

In order to achieve functional equivalence between an aggregate schema database and its set of underlying databases, four basic tasks must be performed:

1. Mapping of aggregate schema names to underlying databases.
2. Maintenance of inter-database connections.
3. Maintenance of currency for the aggregate database.
4. Protecting the consistency of the aggregate databases.

We implemented the above in ORACLE 8, an object-relational database. We were able to map, in totality, WANDA semantic schema. This paper reports only part of the Observation subschema; Full presentation will be presented at the conference.

```
CREATE TYPE Atmospheric-event-t as OBJECT(
eventno             Number(10),
when-utc             Date,
eventname           V archar2 (30),
yr-event-no          Number (6),
landfall             Number,
type                Varchar2 (30),
MAP MEMBER FUNCTION
  event_no RETURN NUMBER,
  PRAGMA RESTRICT_REFERENCES (
    event_no, WNDS, WNPS, RNPS, RNDS));
CREATE TABLE Atmospheric-event OFAtmospheric-event-t( PRIMARY KEY (EVENTNO));;
CREATE or REPLACE TYPE BODY atmosevent_t AS
MAP MEMBER FUNCTION event_no RETURN NUMBER IS
BEGIN
 RETURN EVENTNO;
END;
END;
CREATE TYPE  last_event_no_t as object(
lasteventno      NUMBER(10),
MAP MEMBER FUNCTION
  last_event_no RETURN NUMBER,
  PRAGMA RESTRICT_REFERENCES (
    last_event_no, WNDS, WNPS, RNPS, RNDS),
MEMBER FUNCTION
  new_event_no RETURN NUMBER,
  PRAGMA RESTRICT_REFERENCES (
    new_event_no, WNDS, WNPS) );
CREATE or REPLACE TYPE BODY last_event_no_t AS
MEMBER FUNCTION new_event_no RETURN NUMBER IS
tmp_no  NUMBER(10) := 0;
BEGIN
 tmp_no := lasteventno + 1;
 RETURN tmp_no;
END;
CREATE TYPE observed-date-t as OBJECt(
when-utc              date,
eventref            REFAtmospheric-event-t);
CREATE TABLE Observed-date of observed-date-t(
SCOPE FOR (eventref) IS atmospheric-event);
CREATE TYPE observed-point-t as OBJECT(
latitude-degree             Number(8,5),
longitude-degree             Number(8,5),
when-utc                    REF observed-date-t  );
CREATE TYPE analysis-t as Object(
when-utc                      date,
at-hour-minute               Number(4),
exposure                    Varchar2(15)  );
CREATE TABLE analysis of analysis-t;

CREATE TYPE observation-t as OBJECT(
zonal-wind-mps          Number(8,5),
meridionall-wind-mps     Number(8,5),
temperature-c           Number (8,2),
relative-humidity-%     Number(8,2),
geopotential-ht-mps      Number (8,2),
points                 observed-point-t,
produced-by            REF sensor-t,
quality-control        Varchar2(10),
qc-when-utc             date,
participate-into        REF analysis-t,
MEMBER FUNCTION raw RETURN NUMBER
PRAGMA RESTRICT_REFERENCES (raw, WNDS, WNPS);
CREATE or REPLACE TYPE BODY AS
MEMBER FUNCTION edited RETURN NUMBER
PRAGMA RESTRICT_REFERENCES (edited, WNDS, WNPS),
MEMBER FUNCTION raw RETURN NUMBER IS
BEGIN
update observation
  set quality-control = 'Raw';
 RETURN 1;
END;
CREATE TABLE observation of observation-t( SCOPE FOR(produced-by )  IS SENSOR,
SCOPE FOR (participate-into) IS analysis);
```


1. Smith and Smith, Database Abstraction: Aggregation and Generalization, ACM Transactions on Database Systems Vol. 2, No. 2, June 1977.
2. Chen P., The Entity-Relationship Model toward a Unified View of Data, ACM Transactions on Data Base Systems, Vol. 1, No. 1, March 1976, pp. 9-36.
3. Aschim, Frode, Mostue Bernt M., IFIP WG 8.1 Case Solved Using SYSDOC and SYSTEMATOR, Information Systems Design Methodologies: A Comparative Review, edited by T.W. Olle, H.G. Sol, and A.A. Verrijn-Stuart,North_Holland Publishing Company, 1982, pp. 15-40.
4. Date J.C., An Introduction to Database Systems, Sixth Edition, Addison-Wesley Publishing Company, 1996.
5. Peckham Joan, Maryanski Fred, Semantic Data Model, ACM Computing Surveys, Vol. 20, No. 3, September 1988, pp. 153-189.
6. Brodie Michael L., Active and Passive Component Modeling, Information Systems Design Methodologies: A Comparative Review, edited by T.W. Olle, H.G. Sol, and A.A. Verrijn-Stuart, North_Holland Publishing Company, 1982, pp. 41-99.
7. Brodie Michael L., Data Abstraction for Designing Database-Intensive Application, Proceedings Workshop on Data Abstractions, Databases and Conceptual Modeling, SIGPLAN Notices, 16, 1, January 1981.
8. Gustafsson Mats R., Karlsson Terttu, Bubenko Janis A., A Declarative Approach To Conceptual Information Modeling, Information Systems Design Methodologies: A Comparative Review, edited by T.W. Olle, H.G. Sol, and A.A. Verrijn-Stuart,North_Holland Publishing Company, 1982, pp. 93-142.
9. Rishe Naphtali, Database Design: The Semantic Modeling Approach, McGraw-Hill, Inc., 1992.
10. Morisseau-Leroy Nirva, Atmospheric Observations, Analyses, and The World Wide Web Using A Semantic Database, Master Thesis, School of Computer of Sciences, Florida International Universit, Miiami, Florida, April 1997.