f=u

99-HE

6106

IEEE TRANSACTIONS ON

KNOWLEDGE AND DATA ENGINEERING

A publication of the IEEE Computer Society

MARCH / APRIL 1999 VOLUME 11 NUMBER 2 ITKEEH (ISSN 1041-4347)

REGULAR PAPERS

CORRESPONDENCE

Dr. Naphtali Rishe
Florida International University
School of Computer Science
Southwest 8th St. and 107th Avenue
University Park
Miami, FL 33199

1

IEEE COMPUTER SOCIETY

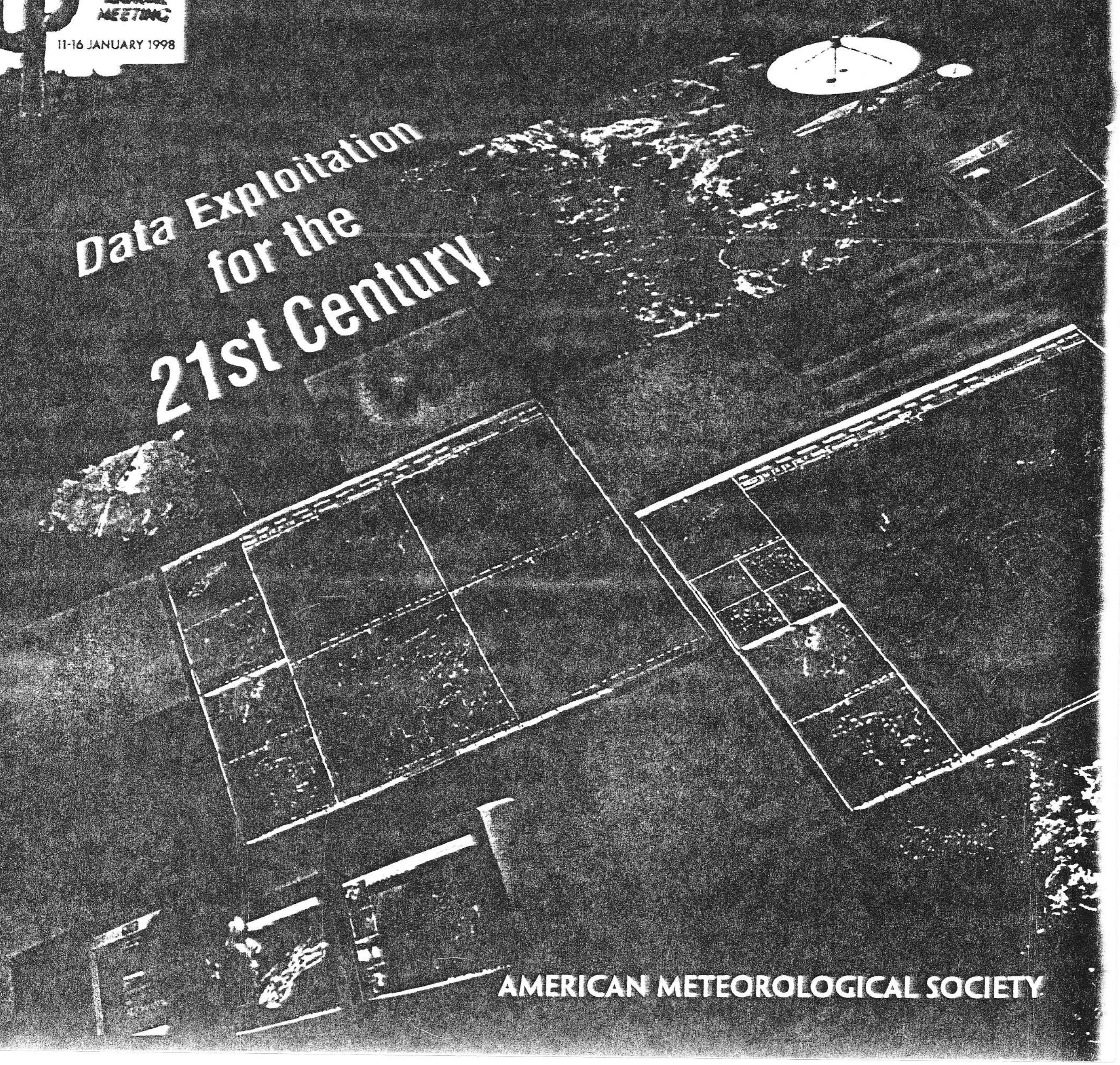
http://computer.org • tkde@computer.org

PREPRINT

14TH INTERNATIONAL CONFERENCE ON INTERACTIVE INFORMATION AND PROCESSING SYSTEMS FOR METEOROLOGY, OCEANOGRAPHY, AND HYDROLOGY

JANUARY 11–16, 1998
PHOENIX, ARIZONA

78th AMS ANNUAL MEETING
11-16 JANUARY 1998



AMERICAN METEOROLOGICAL SOCIETY

14th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology

Sponsored by

AMERICAN METEOROLOGICAL SOCIETY

Front Cover: The cover connotes the rapidly emerging capability for the hydrometeorological science community to access and to exploit the data produced by new sensor systems and data generators deployed as elements of the Modernization of the National Weather Service. As the Modernization Program moves toward completion in 1999, members of the community will have ever increasing access to rich sets of imagery, observations, and high resolution numerical weather predication model output. The commercial satellite data broadcast system, NOAAPORT, is the principal resource that will distribute these data.

As discussed in papers 8.22 (page 392), entitled *"Plans for the NOAAPORT Data Stream"* (by Glenn K. Rutledge of the NWS, Office of Meteorology, Silver Spring, MD, et al) and paper 8.1 (manuscript not available) entitled, *"Advanced Weather Interactive Processing System (AWIPS) Communications Network (CAN) Improvements for Deployment"* (by Ed Moore and L. Johnson, PRC, Inc. McLean, VA), the data accessible through NOAAPORT will be a major contributor in driving applied hydrometeorological research in the next century .

This cover is sponsored by PRC Inc. of McLean, Virginia. PRC is the prime contractor for the Advanced Weather Interactive Processing System (AWIPS) and, in association with GTE, for the related NOAAPORT satellite communications data broadcast system.

AMERICAN METEOROLOGICAL SOCIETY

45 BEACON STREET, BOSTON, MASSACHUSETTS USA 02108-3693

TABLE OF CONTENTS

14TH INTERNATIONAL CONFERENCE ON INTERACTIVE INFORMATION AND PROCESSING SYSTEMS (IIPS) FOR METEOROLOGY, OCEANOGRAPHY, AND HYDROLOGY

** Manuscript not available*

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

The Wind Analysis Distributed Application (WANDA) enables scientists at NOAA/AOML/HRD to access, quality control and visualize atmospheric observations and stormfixes on UNIX workstations. The development of WANDA started in 1992 and the third release became available to internal HRD scientists in June 1996.

WANDA is a critical software application, a real-time wind analysis system that provides the most current diagnostic information on the actual areas impacted by hurricane wind fields. The system is equally suitable for use in a research and operational environment. WANDA's final product is a graphical output, presented in a geographical context, that conveys the destructive potential of the wind field.

The user interface is object based (in Objective-C) running on the NEXTSTEP platform and the back-end engine is procedural based (in FORTRAN) running on digital VMS. The object-oriented approach and the modular design ensures that WANDA can be extended to fulfil new requirements. The system is based on a database of ASCII files for data access, in a specific format (Lord Franklin file format).

At NOAA/AOML/HRD information needs fall under two major groups: 1) Data set information sharing between NOAA laboratory scientists on site and 2) Data set information sharing between outside scientists and the general public. The current flat file database makes it difficult to fulfill these needs. The goal of this paper is to design and implement a database system using a Semantic Data Model.

Semantic modeling is a 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 the user relationships between data objects that support the manner in which the user perceives the real-world. It provides the designer the mechanisms and the tools to design a higher level data model which enables the database designer to naturally and directly incorporate in the conceptual model or schema a large portion of the semantics of the database.

This paper details the logical or conceptual model, describes a particular semantic data model, called the Semantic Binary Model (SBM) [9], and summarizes the mapping of the logical design, the implementation of such design, into a physical design using an object-relational database. Here we describe the WANDA database, a database that keeps track of atmospheric observations, stormtrack data, and wind analyses.

1. SEMANTIC MODELING

The main objective of semantic data models [1][2][3][5] 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.

Some of the advantages of 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 [9] and uses this model to capture the structure of the WANDA database.

2. 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

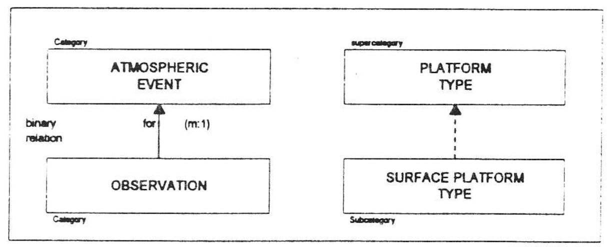
* Corresponding author address: Nirva Morisseau-Leroy, NOAA/AOML/HRD, 4301 Rickenbacker Causeway, Miami, Florida, 33149; email: moriss@aoml.noaa.gov

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.

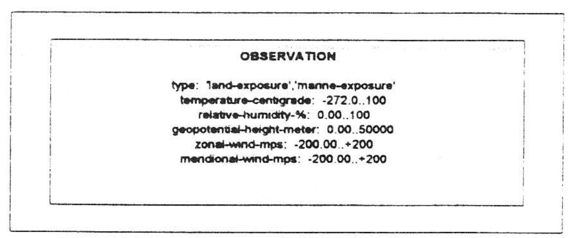
3. SEMANTIC BINARY MODEL NOTATION

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 categorization of objects is as follows [9]:

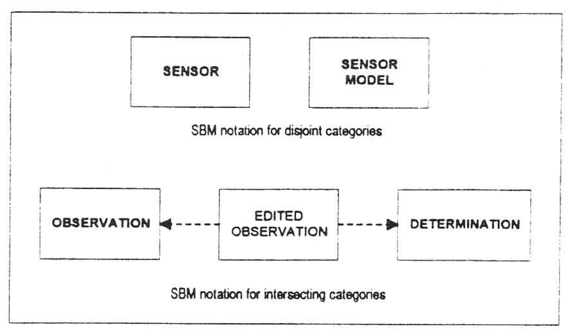
- Value of concrete object: A printable object such as a number, a string. For example, latitude is a concrete object. In this example, latitude is also an attribute. An attribute is a functional relation (i.e., many-to-one, or one-to-one) whose range is a concrete category. Figure 2 presents a concrete category named OBSERVATION that keeps track of atmospheric observations and a set of concrete objects (attributes) such as *type, temperature-centigrade, relative-humidity, geopotential-height-meter, zonal-wind-mps,* and *meridional-wind-mps,* that describes the category. OBSERVATION is the domain for these attributes. The values that constitute this domain are specified via a data type such as Integer, Number, String, and so on. SBM provides also the specification of constraints. A constraint is the physical and operational interpretation of the semantic. SBM allows the specification of domain constraint on an object. Domain constraints are constraints on the type of values an attribute can have. For example, the values -90.0000 ...+ 90.0000 are the constraints imposed on latitude objects. Any value outside of this range will be rejected by the database.
- Abstract object: An abstract object is a non-printable object. Such an object can be an item (hurricane Andrew), or an event (wind analysis of a tropical system), or an idea (WANDA application). For example, ATMOSPHERIC EVENT (Figure 1) is an abstract category that keeps track of abstract objects (storms, tropical depression, etc.). ATMOSPHERIC EVENT category reports all tropical systems such as hurricanes, tropical depression, invest, etc.
- Category: A category (Figure3) 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. A category is represented by a rectangle and is identified by a unique name written in upper case. SENSOR is a category that reports all sensors located on an observation platform; SENSOR MODEL a catalog of sensor models; OBSERVATION an aggregation of observations; EDITED OBSERVATION a generalization of observations that undergo quality control; DETERMINATION a catalog that reports on the observation status after they undergo quality control. Categories may be disjoint or they may intersect: SENSOR and SENSOR MODEL are disjoint categories whereas OBSERVATION , DETERMINATION and EDITED OBSERVATION are intersecting categories. The broken arrow between EDITED OBSERVATION and OBSERVATION is used in the SBM to express inheritance between two categories. Inheritance is a powerful abstraction for sharing similarities among categories while preserving their differences. For example, an edited observation object is an observation object that has been edited.
- Subcategory: All objects of a subcategory (Figure 1) belong to another category (the supercategory). PLATFORM-TYPE is an example of a supercategory and SURFACE-PLATFORM-TYPE is a subcategory. SURFACE-PLATFORM-TYPE is a specialized platform type. 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. SURFACE-PLATFORM-TYPE inherits the properties of its supercategory PLATFORM-TYPE.



SBM Notations for category, binary relation, supercategory, and subcategory.
Figure 1



SBM notations for a category with attributes
Figure 2



SBM Notation for disjoint and intersecting categories
Figure 3

- Relation: 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 binary or non-binary relations. For example, *for* is a binary relation between ATMOSPHERIC EVENT and OBSERVATION categories. The many-to-one (m:1) cardinality indicates that there exists many observations for one atmospheric event (many observations for one specific hurricane, for example, hurricane Andrew). The functional relation can be one-to-one (1:1); a one-to-one relationship uniquely identifies an object, for example, the name of sensor. The binary relation *for* (Figure 5) between QUALITY CONTROL EVENT and ATMOSPHERIC EVENT is a many-to-many (m:m) relation indicating that many quality control events are performed on many tropical systems. Appendix A presents the specification for all the concepts used in WANDA Semantic subschemas.

4. CONCEPTUAL MODEL - WANDA SEMANTIC 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. A schema can be subdivided into several subschemas. WANDA Semantic Schema is composed of two subschemas (Figure 4 and 5): Observation Subschema and Atmospheric Observation Platforms Subschema. 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. A SBM schema presents specific user views. In the SBM, a user view is represented in terms of categories, associations, attributes and relations in a view diagram (Figure 4, 5).

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 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.). Additionally, SBM provides mechanisms for the specification of all concepts used in the data model.

4.1 Observation Subschema

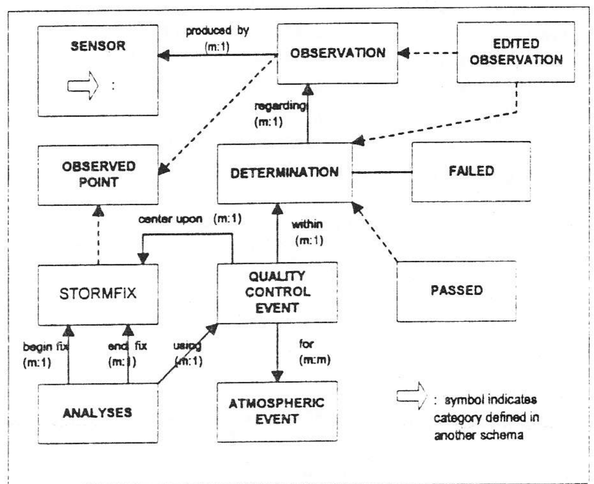


Figure 5 Observation Subschema

An "arrival" transaction for a new observation generates the following events:

- An observation is *produced by* a sensor. Therefore, we store a new sensor object or we establish a new link between the observation object and the SENSOR category (Figure 5).
- A sensor is located *on* an atmospheric observation platform. The database relates SENSOR to ATMOSPHERIC OBSERVATION PLATFORM (Figure 4).
- A sensor is *of* a specific model type.
- An atmospheric observation platform is *of* a specific platform type. This platform type can be a surface platform type (Buoy), an aircraft (NOAA P3), other platform types, a single sensor platform (Profiler, Model), etc.
- An observation has not yet undergone quality control. Scientists have yet to take determining actions on the observation. However, the system will traverse the binary

Master's thesis, School of Computer Sciences, Florida International University, Miami, Florida, April 1997.

APPENDIX A

1. 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)

2. Atmospheric Observation Platforms 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)

APPENDIX B

```
CREATE TYPE atmosevent_t AS OBJECT(
eventno             NUMBER (10),
eventname           VARCHAR2 (30),
landfall            NUMBER,
category            NUMBER,
type                NUMBER,
MAP MEMBER FUNCTION
 event_no RETURN NUMBER
 PRAGMA RESTRICT_REFERENCES (
 event_no, WNDS, WRNS ) );

CREATE TABLE event_tab of atmosevent_t
(PRIMARY KEY (eventno));

CREATE TYPE fixdate_t AS OBJECT(
fixdate             DATE,
for                 REF atmospheric_t );

CREATE TYPE observed_point_t AS OBJECT(
latitude-degree     NUMBER (8,5),
longitude-degree    NUMBER (8,5),
pressure-mb         NUMBER (4),
...,
..,
);
```