02-SD

The 6th World Multiconference on Systemics, Cybernetics and Informatics

July 14-18, 2002
Orlando, Florida, USA

PROCEEDINGS

Volume VII

Information Systems Development II

Organized by IIIS
International
Institute of
Informatics and
Systemics

Member of
International Federation of
Systems Research IFSR

EDITED BY
Nagib Callaos
John Porter
Naphtali Rishe

Semantic Design and Oracle Implementation of the Vegetation Database for the Everglades National Park*

Naphtali D. RISHE, Maxim V. CHEKMASOV, Marina V. CHEKMASOVA,
Natalia Y. TEREKHOVA, Anatoliy A. ZHYZHKEVYCH, Gustavo PALACIOS

High Performance Database Research Center
School of Computer Science
Florida International University
Miami, Florida 33199, U.S.A.

ABSTRACT

Several environmental studies have been conducted in South Florida, which are related to vegetation patterns in that area. These studies have accumulated valuable data on plant species with the goal to use this data in further biological analysis and research. To facilitate this, we have designed and implemented the vegetation database. Semantic modeling was chosen as an efficient and flexible approach to the database design. The Oracle relational database management system was chosen as the platform to implement and maintain the system. In the present paper we discuss the vegetation data sets, database requirements, design and implementation. We also describe the tool built in-house to automatically map a conceptual semantic database design into a physical relational database design.

Keywords: Environmental study, semantic schema, relational database.

1. VEGETATION DATASETS

The datasets to be loaded and maintained in the vegetation database vary greatly in their periods of data collection, geography, methods used by the researchers, and the ways the datasets were formally presented for data loading. We will briefly describe the major datasets.

The *herbarium* dataset contains information on the plants that are displayed in the Everglades National Park's (ENP) herbarium. This collection of plants is constantly growing. The idea behind having this dataset in the database is to facilitate fast and easy search of the specimens in the herbarium and to check the availability of a particular plant in the collection. At this time the dataset has no digital photos of the herbarium plants, but the photos may be added at a later stage.

This dataset is closely related to the *plant species classification* dataset. The latter contains taxonomy information on the vegetation: categories, families, genus, and species of the plants to be found in South Florida. As we see below the important feature of this dataset is introduction of a unique 16-character code, derived from the first four letters of the genus, specific epithet, and infraspecific name (if applicable) for each plant species in classification.

The *solution holes* dataset is related to the study of land coverage with plants on several geographical locations (sites) in the ENP. At each site several holes were artificially prepared for the study. The resulting dataset contains physical parameters of the holes like soil depth as well as hole coverage information with live and dead plants.

Length measurements for plants at the Dry Tortugas National Park is yet another dataset to be stored in the database. Several transects were prepared in the Park to conduct measurements. Each transect is divided into seven segments. The dataset records the starting/ending dates of the plant measurements and the heights of the plants.

The study conducted on *the plots in Taylor Slough* resulted in a separate dataset. Unlike the previously described dataset, the plots here were setup using other methodology. The plot records contain data on the physical parameters of the local area like water depth as well as percentage of coverage of the plots with the vegetation.

A special study is conducted to monitor restoration efforts in the ENP. The study produces the so called *'hole-in-the-donut'* dataset. The measurements are conducted on a set of plots, each being precisely geo-located and referenced to the neighboring plots. Record information includes physical parameters and plant

* This research was supported in part by NASA (under grants NAG5-9478, NAGW-4080, NAG5-5095, NAS5-97222, and NAG5-6830), NSF (CDA-9711582, IRI-9409661, HRD-9707076, and ANI-9876409), ONR (N00014-99-1-0952), the Department of Interior, and the FSGC.

conditions. Additionally plant phenology data and tree islands' study results are included. Since the restoration effort in the Park has a complex effect on the whole area, the study also produced data on soil and water depth, as well as wildlife monitoring records.

Finally, exotic plants treatment using herbicides is regularly conducted in the ENP. The data on site visits, plant treatments, personnel, and herbicides used are combined to form *the exotic plants treatment* dataset.

2. USER REQUIREMENTS TO THE DATABASE

From the previous section it is clearly seen that the vegetation studies differ greatly in the data objects collected and the methodology used to conduct the studies. One of the requirements for the database was to combine all the data in a single database. To create a database design where all the data components are logically related between themselves is a challenging task in this case. However we were allowed to choose the methodology to create the design of the vegetation database which proved to be helpful in accomplishing the task.

Another requirement was to implement the system using Oracle RDBMS (relational database management system), [1]. Oracle was chosen because the clients have already used it for their database needs and possess a license for this software package. The database server is installed under a UNIX operating system, whereas most of the database users will use a Microsoft Windows platform for their work.

Certain user requirements are related to the applications to be constructed to facilitate the users' needs in querying the database, saving the query results, importing the results into third-party software tools, and making modifications to the data. It is not expected that an average database user, being a specialist in biology and environmental sciences, will possess skills in database query languages such as SQL (structured query language) or programming. It was decided to create a set of intuitive and easy-to-use forms to assist clients in their work with the data utilizing the software tools provided in the Oracle package.

Some user requirements are related to data maintenance. In particular, new data has to be regularly uploaded to the database, preferably automatically.

3. SEMANTIC SCHEMA FOR THE DATABASE

The semantic modeling approach was chosen for the database design. This approach allows the unification of different database models into one framework. Most of the concepts and languages may be presented in terms of a unifying semantic model. The other models may be technically treated as subsets of the semantic model. Therefore, the concepts and languages of the semantic model may apply to them.

For our project, semantic modeling may be considered as a tool for database design in the relational database model. The top-down relational database methodology may be presented as follows. First, we analyze and specify our project semantically. This produces a concise, flexible, user-oriented specification of the vegetation database, unconstrained by computer-oriented concerns. In the second stage this specification is converted into a relational database schema with all its integrity constraints. The semantic description, which we call the **semantic schema**, remains as a high-level documentation of the database.

Semantic modeling for the database project starts with the identification of the **objects**. An object may be any item in the real world. It can be either a **concrete object** (value) or an **abstract object** (non-value). A **category** is a concept of the database design that is a unary property of objects. In other words, a category describes a set of objects which possess the property. **Abstract categories** are categories whose objects are always abstract, **concrete categories** are the ones whose objects are always concrete. A **binary relation** is a concept of the database design that is a binary property of objects. It has the meaning of relationship or connection between two objects. **Attributes** are binary relations whose range is a concrete category. These concepts are used in the semantic design of the database, which is represented by the semantic schema. We refer to [2] for a thorough discussion of the semantic modeling approach.

For visual presentation and printouts, the semantic schema may be depicted as a set of diagrams called **sub-schemas** of the semantic schema. A category is displayed on the schema as a box with the category name in uppercase bold letters. Attributes appear inside the corresponding category boxes in italic letters. Relations between the categories are denoted by arrows. The names of the relations appear in bold letters. Cardinality of the relations as well as additional constraint information is placed on the semantic schema in italics.

Figure 1 presents a sub-schema for taxonomic classification of plant species from the semantic schema for the vegetation database. Categories **CATEGORY**, **FAMILY**, **GENUS**, and **SPECIES** store the taxonomy structure for plant species. In particular, each species belongs to a certain genus, which in turn is part of the family, and families comprise a category. We also store information on the subspecies in category **SPECIES**, with the relation **is-part-of** linking subspecies to the corresponding species record. The relations, corresponding to the above-mentioned categories, namely **the-genus, the-family, the-category,** are marked as *total*,

indicating mandatory status of the relationships information. Thus each plant

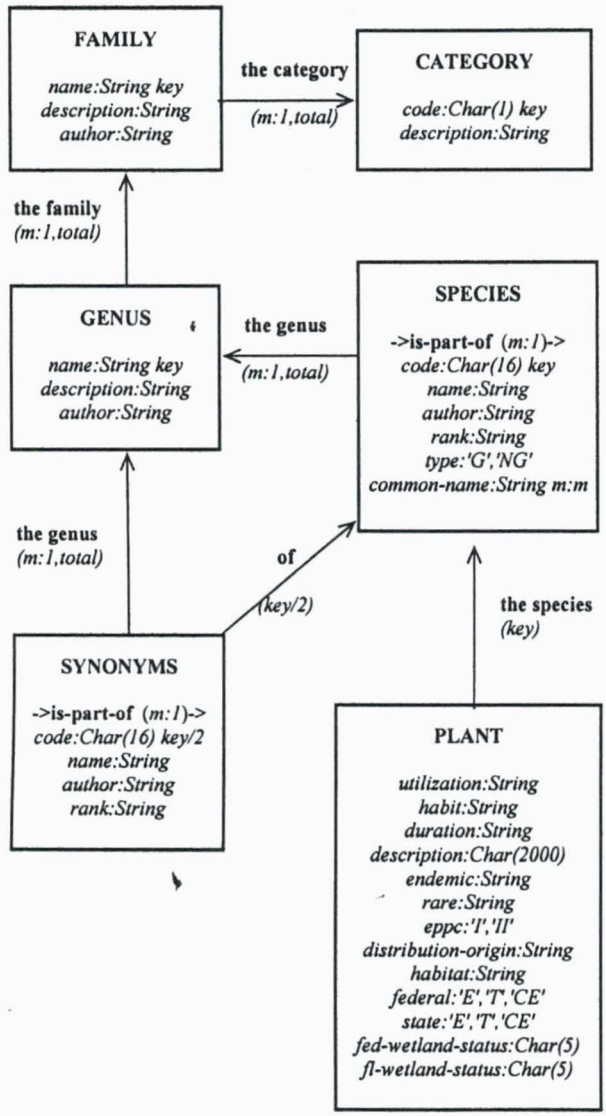

Figure 1. A semantic sub-schema for taxonomic classification of plant species.

species is uniquely identified by 8-, 12- or 16-letter code, derived from the first four letters of the genus name, first four letters of the species name and, if applicable, first four letters of the infraspecific name(s).

It is interesting to learn that the botanists are still debating on affiliation of some plants with particular genus and species. Thus we decided to introduce a separate category **PLANT** to store the information on plant species. When the taxonomy name of a particular plant is changed by the botanists (which in terms of the database is equivalent to relating plant information with another species code), only relation **the-species** between the categories **PLANT** and **SPECIES** has to be changed avoiding reload of the whole record for this plant. Finally, the category **SYNONYMS** was introduced to store the alternative names for the plant species used in the works of some researchers.

The concept of a plant's common name is an important one related to plant species. Several plant species may have the same common name, and one plant species may have several common names. Since many historical studies describe the plants using common names, it is crucial to carefully maintain the relationship between species and common names. An attribute *common-name* of cardinality many-to-many is introduced into the category **SPECIES** to store this information.

4. RELATIONAL SCHEMA FOR THE DATABASE

In this section we describe the **schema-conversion** process. This process is a widely used means of database design: a schema is first designed in a higher-level database model and then translated into a lower-level model which is supported by the available DBMS (database management system). In our case, the semantic schema of the vegetation database represents a high-level conceptual database design, and a set of SQL data definition instructions for the Oracle RDBMS represent a low-level physical database design.

The schema conversion process consists of several major steps:

1. Choosing a key for every abstract category, excluding subcategories.
2. Converting the intersecting abstract categories into disjoint categories.
3. Converting every proper one-to-many or many-to-many relation whose range is a concrete category into a new abstract category with its two functional relations through a relation-split.
4. Converting every one-to-many relation into a many-to-one relation by changing its direction and its name.
5. Converting every proper many-to-many relation into a category and two functional relations through a relation-split.
6. Choosing a key for every category produced through a relation-split.
7. Replacing every many-to-one relation *f* whose range is an abstract category by the composition of *f* on the chosen key of its range.
8. Removing redundant non-key attributes.
9. Translating the integrity constraints into the terms of the new schema: the constraints of the original semantic schema and the additional constraints accumulated during the conversion process.

A special tool called **sem2rel** has been developed in-house which implements all the steps of the schema-conversion process. The tool takes the database semantic design as input (semantic design is stored in a plain text file as a set of SDL (schema definition language) instructions) and produces a set of SQL data definition instructions for the Oracle RDBMS.

As an illustration of the conversion process, let us consider an attribute *common-name* of the category **SPECIES** having cardinality many-to-many (see Figure 1). On the high-level design this attribute is represented by the instruction:

```
.aT common-name SPECIES String m:m
```

On the low-level design this attribute is converted to the following SQL instructions creating separate table and constraints:

```
/** CATEGORY SPECIES__common_name **/

CREATE TABLE SPECIES__common_name
   ( species__code_K VARCHAR2(16)
          NOT NULL
     , common_name_K VARCHAR(50)
          NOT NULL
) STORAGE (INITIAL 3M NEXT 500K);

CREATE UNIQUE INDEX
IND_SPECIES__com20 ON
SPECIES__common_name(
species__code_K,common_name_K)
TABLESPACE vegindx
STORAGE (INITIAL 3M NEXT 500K);

ALTER TABLE SPECIES__common_name
ADD PRIMARY KEY
(species__code_K, common_name_K);

ALTER TABLE SPECIES__common_name
ADD (
FOREIGN KEY (species__code_K)
REFERENCES SPECIES (code_key));
```

The difference is striking in volume and complexity of the instructions representing conceptual and physical designs of the database. It is assumed in this example that the tablespaces corresponding to the vegetation database and the database user are properly created under Oracle RDBMS prior to executing the SQL instructions provided above.

5. IMPLEMENTATION AND RESULTS

The SQL instructions for creation of the vegetation database can be executed under Oracle environment thus completing physical implementation of the database. The next step is the loading of actual data into the database. We planned this to be a semi-automatic process. A set of formats for the data files was worked out. The data to be loaded into the database has to conform to one of these formats. A tool was created which automatically takes the data file in a predefined format and loads the data into the vegetation database. We are now nearing completion of the initial data loading.

In the future, we plan to design and implement an Internet-based application for querying the vegetation database and receiving the results. The application will be based on Oracle PL/SQL procedures, and applets written in Java, JavaScript, and HTML. A separate set of forms created for this application will facilitate manual data updates by the researchers. The vegetation web application will work in two modes of access. An internal access system for querying the whole database and making data updates will only be granted to the researchers. Public access will probably be limited to some particular datasets and no updates will be allowed.

6. REFERENCES

[1] Oracle Corporation, http://www.oracle.com

[2] N. Rishe, Database Design: The Semantic Modeling Approach, McGraw-Hill, 528 p., 1992.