A Conceptual Model for the Management of Multi - Representation Geographical Information

Ugo Falchi
Department of Sciences and Technologies, University of Naples “Parthenope”
Centro Direzionale, Isola C4, Naples, 80143, Italy
ugo.falchi@uniparthenope.it

Abstract—The project was aimed at the creation of software tools for managing instances of a multi - representation geodatabase, able to define multiple representations and topological constraints, in relation to modelled objects and structures according to the classification of IntesaGIS first and, later, the Italian national technical specifications of the November 10, Italian Ministerial Decree 2011.

An original conceptual model was created, capable of expressing the different representations (scale, time) of the same geographic entities and relationships between them, directly into the geodatabase structure, by inserting an additional level of abstraction.

In addition, various software tools were developed to create the database schema and its population using command line on Windows and Linux systems, as well as a graphical interface.

The final goal is to be able to meet the needs of the production and use of geographical information with spatial and temporal components, with four dimensions: structure, space, time, and multiple representations.

These dimensions are perpendicular to one another and enable individual operations to be performed independently of each other.

Keyword-Geographical database, Multi representation, Conceptual model, Topological relations, Constraints

I. Introduction

Based on the knowledge of the state of the art of data modeling with spatial and temporal components, gained through study and the previous experiences of the author, the aim of the first phase of the research is to develop a new conceptual and operational model of a relational geodatabase. This is done through the concepts of multi - representation, the use of topological constraints in accordance with the conceptual models based on former IntesaGIS technical requirements and the implementation of explicit relations among geographic data.

The four elements mentioned above are typical of GIS and are now being used in spatial data infrastructure. Their simultaneous presence and their integrated use, however, is hardly find because of the difficulty in codifying this information into the database. The proposed solution included the development of these aspects into the application.

An original conceptual model was created, capable of expressing the different representations (scale, time) of the same geographic entities and relationships between them, directly into the geodatabase structure, by inserting an additional level of abstraction.

In addition, various software tools were developed to create the database schema and its population using command line on Windows and Linux systems, as well as a graphical interface.

II. State Of The Art

This research was initially started in 2008 and was inspired by the results of both the European project Murmur (Multi Representations - Multi Resolutions) and the development of MADS, a conceptual model for the development of traditional applications and space - time, (a project led by Prof. Christine Parent (University of Lausanne) and Stefano Spaccapietra (EPFL), with the cooperation of Prof. Esteban Zimànyi, Université Libre de Bruxelles), both from the ChronoGeoGraph project, developed by Donatella Gubiani and Angelo Montanari, University of Udine.

The storage systems commonly used to manage spatial data, mainly rely on a relational engine (RDBMS). Since the GIS systems were designed according to this logic, the storage of geometric entities in alphanumeric tables, therefore represented a very effective natural evolution.

One of the first relational geodatabases was produced by Oracle and was enriched with GIS capabilities of ESRI SDE. SDE uses a series of meta tables with which to store the information of the space component, hierarchy information and the symbolism of the themes. Later, other commercial solutions were developed, such as MySQL and PostgreSQL, making model the most popular among enterprises.

However, this model shows all its limitations if additional information is introduced, such as the relationship between time and spatial connotation or a vision multi - representation of an entity, for example the same portion of urban territory as seen as by an urban or a transportation engineer.

The project was aimed at the creation of software tools for managing manage instances of a multi - representation geodatabase, able to define multiple representations and topological constraints, in relation to modelled objects and structures according to the classification of IntesaGIS first and, later, the national technical specifications of the November 10, Italian Ministerial Decree 2011.

Constraints and metadata have been managed within the geodatabase in a transparent way, the user is thus not required to know all the implementation details.

By incorporating the needs expressed by the decision makers in the control and management of the local area, it was possible to begin the design of a global scheme, capable of supporting multiple representations, where the copied elements, along with their geometric primitives, scale, and time, contribute to the formation of separate tables.

III. Multi – representation database

The final goal is to be able to meet the needs of the production and use of geographical information with spatial and temporal components, with four dimensions: structure, space, time, and multiple representations. These dimensions are perpendicular to one another and enable individual operations to be performed independently of each other.

The concept of backward compatibility was given special importance. The work was carried out in order to extend the new concepts, information and links to traditional geodatabases, or with the current GIS software.

From a practical point of view this led to the need for methodological and conceptual compromises, as well as inheriting all the limitations of the scalability of software and technological levels typical of the RDBMS (Oracle, PostgreSQL and so on).

The software tools for the creation of the database schema and its population can be used both from the Linux and Windows or GUIs.

From a technological point of view, the products made in the project were written in Java and consist of several executables (tools) that operate in Oracle.

The product is not intended to replace existing relational database management system (RDBMS), but aims to complete integration.

IV. Main characteristics

In cartography, each geographic entity is a real world object, a multi - representation of this concept is explained by the possibility of forming multiple representations of each element rendered, for example different maps can represent the same object from different views, scales, and times.

In digital mapping, this paradigm is usually treated by separating information and geographical entities into different databases; the most common case is a breakdown by scale or time of acquisition. Often multiple representations are managed by traditional geodatabases as separate classes, which become meaningful only through specific software applications.

However, the various entities, views from different representations, are not connected in the same database and there is only a key relationship. The knowledge of the different representations of the same geographical entity and the relationships between them is not codified in the geodatabase in a standard way. What is proposed is to add this additional level of abstraction to the classical system of geo - relational storage in the management of representations and thus make data linked as a structured level.

Each real element, the object of interest was modeled in the database and constituted by a series of tables containing different representation of shape, scale, and time.

To associate a specific class with various forms (performances), this class is represented by several tables with geometries that can also possess several primitives: simple (polygons, lines, points) or complex (multipolygons, multiline, points).

Similarly, to associate a class with different scales, the resulting representations are made of geometries with different densities that specify the degree of accuracy of the representation.

Finally, classes can be defined that have only one representation and which thus consist of a single mode of analysis.

As the topological constraints are also "weak" mechanisms because of the external database, they are often taken into account only during digitization and the acquisition of data. They are also managed by complex proprietary tools (see ArcGIS). Typically, any constraint or control topological was developed when entering or editing data in the geodatabase.

Developing the conceptual model, which is the aim of this paper, the topological relationships (mutual constraints) were coded as intrinsic to the data itself, and were set at the level of the database. They are managed via ad hoc tools which can simplify the definition of the rules and the population of the geodatabase. It is therefore possible, to guide users in the event that they perform operations that violate these reports.

V. STRUCTURE OF THE GEOGRAPHICAL DATA

The multi - representation model was developed according to Italian guidelines. Consequently, the conceptual model has been enriched with new concepts.

The conceptual model resumes and reinterprets the paradigms exploring the experiences of MADS and ChronoGeograph, relative to the mutual relations between different representations of the same geometric entity. In particular, it draws a strict criterion for the definition and management of these relationships, increasing the level of information available into databases.

Different representations of an object can then be explored, such as its temporal evolution or its representations at different scales, by encoding the information in the database level rather than at the application level.

This innovation results in differences in the way the data are store and processed. It is necessary to add a level of connection between the various representations of each entity, which is achieved by adding an appropriate table which conveys this information. It will be necessary, therefore, to become more aware in the acquisition or production of geographic data. In other words, the traditional GIS data need to be accompanied by additional information that can describe the relationships between the various entities.

The structuring of the data containers of the proposed model, according to the technical norms, has provided a clear hierarchy that was rebuilt in each instance of the multi – representation. The tables have also been instantiated for all classes, even empty ones.

The model includes a meta-catalog that explains all the physical structure and makes it usable and understandable by application. It is important to highlight that the basic GIS information continues to be stored in the same way in the geodatabase (through tables), and the underlying infrastructure continues to operate in the traditional way (through a database engine).

In detail, the articulation in layers, themes and classes has been further implemented with multiple representations.

In addition to the two files model.xml and db.xml, the database also includes geographical and hierarchical information related to the elements represented, according to the following summary:

- MOD_CLASS, used to describe the classes (Table I);
- MOD_THEME, used to describe the themes (Table II);
- MOD_LAYER, used to describe the relationship between the classes and the different representations (Table III);
- MOD_ FIELDS, used to describe the class fields;
- MOD_RELATIONS, used to describe the generic or simple relationships (Table V);
- MOD_TOPOLOGICAL_RELATIONS, used to describe the topological relationships (Table IV).

A. Structure of the model

The file model.xml is the model that contains information on the organization in layers, themes and classes of the database, as well as borrowing from national technical requirements, and can be customized according to the needs of the project.

This file determine the physical structure of the database, since the metatables contain only information on the articulation of the structure.

Loading model.xml on the graphic interface of the tool developed, you can view the file structure and the choices you make.

Based on the file model.xml, the metatables layers, themes and classes will be populated with the data given and will be connected to each other through their respective layer, theme, class fields.

The field of the class metatable MOD_CLASS (Table I) is the sum of fields of MOD_THEME (Table II) and MOD_LAYER (Table III). For example, if the class field is MOD_CLASSI ST01TE01CL01, it means that the field of theme MOD_TEMI is 0101 and the field layer MOD_STRATI is 01.

The file db.xml, an example of which is shown below, is the file endorsements. It generally contains:

TABLE I. Metatable with Class Descriptions and Representations

MOD_CLASS		
Field	**Type**	**Description**
class	VARCHAR2 (30)	The name of the class. Example: ST01TE01CL01
representation	VARCHAR2 (3)	The name of the representation (the same as the table name). Example: ST01TE01CL01_PLG_A_1k
shape	VARCHAR2 (3)	The geometry of the object. Example: PLG
time	VARCHAR2 (30)	The string that indicates the temporal dimension of the representation. Example: 20110601
ladder	VARCHAR2 (5)	The scale of representation. Example: 1k
codiceTempo	VARCHAR2 (5)	The code representing the temporal dimension. Example: A
alias	VARCHAR2 (150)	The alias class. Example: AREA TRAFFIC VEHICULAR
Description	VARCHAR2 (3000)	The class description.
view	VARCHAR2 (35)	The name of the view of the representation. Example: MT_AREADICIRCOLAZIONE_PLG_A_1k

TABLE II. Metatables with Theme Descriptions

MOD_THEME		
Field	**Type**	**Description**
theme	VARCHAR2 (4)	The ID of the theme. The first two digits indicate the layer which it belongs to..Example: 0101
alias	VARCHAR2 (150)	The alias theme. Example ROADS
Description	VARCHAR2 (3000)	The description of the topic

TABLE III. Metatables with Layer Descriptions

MOD_LAYER		
Field	Type	Description
layer	VARCHAR2 (2)	The ID layer. Example: 01
alias	VARCHAR2 (150)	The aliases layer. Example TRAFFIC, MOBILITY AND TRANSPORT
Description	VARCHAR2 (3000)	The description of the layer

The file db.xml, an example of which is below, is the endorsements file. It generally contains:

- default representations to associate with each class - they are defined within the Multirep block with the allClass construct. In this case, the specified representation is applied to all the classes defined in model.xml. For example:

<allClass scale = "1k" time = "A" />

<AllClass scale = "3k" time = "D" />

In this example we created two representations for each class: a scale 1k to the time "A" and thesecond of scale 3k to the time "D".

- specific representations to associate with a group of classes were defined within the Multirep block. This block is where users specify the classes to which they want to associate the representation. Field representation of metatable MOD_CLASS (Table I) contains the name of the table which the geographical representation refers to. Thus, within the database there is a table for each geographical representation. When the loading tool used, these tables are populated if the shape folder specified on the command line when running the tool, contains the corresponding file. For example:

<Representation scale = "1k" time = "A">

<Class st = "09" you = "01" cl = "01" shape = "pts" />

<Class st = "09" you = "01" cl = "02" shape = "plg" />

</ Representation>

The scale and time attributes of the tag representation represent the scale and the time defined by the representation. The blocks class indicates which classes should be associated with this representation. In this case the attribute st is the layer of the class, the attribute te represents the theme, cl represents the class, and shape represents the geometry to use (pl, arc, or the pts).

This construct is the only one that allows the default geometry of a class to be changed.

It is possible to use both constructs (allClass and rapresentation), or alternatively just one of them.

- the topological relationships that bind the classes - the topological relationships are defined in the following form:

<Relation name = "touch">

<Class st = "01" te = "01" cl = "07" shape = "plg" scale = "1k" time = "A" />

<Class st = "01" te = "01" cl = "08" shape = "plg" scale = "1k" time = "A" />

</ Relation>

Where touch is the kind of relationship (adjacent). The second class is dependent on the first declared (this means that the data of the first class must be entered into the database before the data of the second class). In practice, the second class is the one that must be controlled when inserting or updating data.

Multiple relationships can be tied with the symbol "+", for example:

<Relation name = "+ cover contains">

<Class st = "01" te = "01" cl = "07" shape = "plg" scale = "1k" time = "A" />

<Class st = "01" te = "01" cl = "08" shape = "plg" scale = "1k" time = "A" />

</ Relation>

This means that the relationship is satisfied if the class ST01TE01CL07_PLG_A_1k is completely within the class ST01TE01CL08_PLG_A_1k (without intersecting edges), or if the class ST01TE01CL07_PLG_A_1k lies completely within ST01TE01CL08_PLG_A_1k and their edges have points of intersection.

Below is an example of the file db.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<db>
 <mandatoryInfo
  idField="ID_FIELD"
  geometryName="ORA_GEOMETRY"
  extentX="426609.226100000, 570197.675700000, 0.000000050"
  extentY="4301310.240000000, 4573593.134500000, 0.000000050"
  extentZ=""
  srid=""
 />
  <multiRep>
  <time>
```

```
     <timeRep code="A" value="20110101" />
     <timeRep code="B" value="20110102" />
     <timeRep code="C" value="20120103" />
     <timeRep code="D" value="20120104" />
   </time>
  <representations>
     <allClass scale="1k" time="A"/>
     <allClass scale="3k" time="D"/>
     <!-- representation scale="1k" time="A" >
       <class st="09" te="01" cl="01" shape="plg" />
     </representation>
     <representation scale="3k" time="A" >
       <class st="09" te="01" cl="04" shape="plg" />
     </representation -->
   </representations>
  </multiRep>
  <topologicalRelation>
   <!-- test -->
   <relation name="touch">
     <class st="09" te="01" cl="04" shape="plg" scale="3k" time="A" />
     <class st="09" te="01" cl="04" shape="plg" scale="1k" time="A"  />
   </relation>
  </topologicalRelation>
  <simpleRelations>
    <simpleRelation>
      <class st="09" te="01" cl="04" shape="plg" scale="3k" time="A" field="A09010401" />
      <class st="09" te="01" cl="04" shape="plg" scale="1k" time="A" field="A09010401" />
    </simpleRelation>
    <simpleRelation>
      <class st="09" te="01" cl="05" shape="plg" scale="3k" time="A" field="A09010501" />
      <class st="09" te="01" cl="05" shape="plg" scale="1k" time="A" field="A09010501" />
    </simpleRelation>
  </simpleRelations>
</db>
```

B. Topological Relations

The topological relations implemented at the database level are:

- disjoint;
- contains;
- touches;
- overlaps;
- equals.

A mechanism was introduced to maintain the knowledge of the relationships between the data in the geodatabase, and the relational model has been extended to make those links explicit.

The topological relationships are contained in a separate metatable (Mod_topological_relations), and are accompanied by a constraint that guarantees the consistency of the data in function of the reports. The maximum benefit in relations was reached in terms of navigation and simplification of query operations, which for the application make good use of the implementation of these meta tables.

TABLE IV. Metatables with Topological Relations

MOD_TOPOLOGICAL_RELATIONS		
Field	**Type**	**Description**
nome	VARCHAR2(30)	The name of the trigger. TOPOLOGICAL_TRIGGER[id progressivo]. Example: TOPOLOGICAL_TRIGGER_0
classe_a	VARCHAR2(30)	The name of the master representation Example: ST09TE01CL04_PLG_A_3k
casse_b	VARCHAR2(30)	The name of the slave representation Example: ST09TE01CL04_PLG_A_1k
relazione	VARCHAR2(25)	Relation type. Example: touch

C. Simple relations

Another level of relations, implemented in model multi - representation, consists of simple relationships between tables. Thanks to these connecting elements it is possible to navigate database tables orthogonally at the application level. The simple relationships are also stored in special meta-tables.

TABLE V. Metatables with Simple Relations

MOD_RELATIONS		
Field	**Type**	**Description**
classe_a	VARCHAR2(30)	The name of the first relation table Example: ST09TE01CL04_PLG_A_3k
id_field_a	VARCHAR2(30)	The field of the first table show in the relation Example A09010401
campo_tabella_rel_a	VARCHAR2(10)	The name of the relation table field that identifies the first table Example: IDA
classe_b	VARCHAR2(30)	The name of the relation second Example: ST09TE01CL04_PLG_A_1k
id_field_b	VARCHAR2(30)	The field ot the second table used into the relation Example: A09010401
campo_tabella_rel_b	VARCHAR2(10)	The name of the table field that identifies the second table. Example: IDB
tabella_rel	VARCHAR2(15)	The name of the table that realize the relation

VI. Analysis of functionality

In this section we focus on the main features that represent the characteristics of the project. As regards the geodatabase multi - representation, note that:

- it was developed at the level of the conceptual model and the physical model;

- it is not a software product and runs on an existing engine (Oracle) exploiting all the functionality of the engine;
- it does not replace the traditional geodatabase but adds new functionality by implementing standard cataloguing data;
- it does not allow any degradation for query performance in traditional applications but provides software support aimed at performing orthogonal searches able to return results including the various dimensions of the model;
- due to the implementations, it occupies more disk space than a traditional geodatabase, using a series of metadata and metatables;
- it does not work on all database engines, but was developed in this phase of the research for use with Oracle Spatial 10 (oracle binary geometry);
- each performance is viewed and treated as a normal class of a geodatabase Oracle Spatial, if it was opened by a traditional application or viewer;
- software tools were developed to instantiate and populate the geodatabase multi – representation, and additional features will be accessible from applications developed specifically for the new paradigms;
- it will include a metacatalog of classes and different representations of the same class, developed according to Italian guidelines, for cataloging content in layers - themes - classes;
- the various representations will be a combination of: form, scale, time.

With regard to the implementation topological constraints, inherent in the model, note that:

- topological constraints have been set in the software modeller for the definition of classes and performances;
- there are clear relationships between explicit representations, classes and data tables;
- there are relationships between the entities of the same class in the different representations in order to use cross queries to find all representations of the same entity (scale / time / etc).

VII. CONCLUSIONS

The conceptual scheme implemented was encoded in its corresponding logical model. In the first test phase the model was populated with a subset of geographicalal information, certainly not exhaustive, but nonetheless able to ensure the smooth functioning of the entire design process.

To perform the upload, management and display of data, various computer artifacts were designed and developed from scratch:

• a Scheme Designer, which allows users to define the logical model and on the basis of this, implement the physical model of an instance of Oracle;

• a Loader, which allows users to populate the database;

• a GUI, which is a graphical interface to the tools and Schema Designer Loader;

• DB Navigator, which is the web interface to the database multi representation.

In the continuing research, a massive loading with a full set of geographic data is expected, in order to create a physical model complex. This will enable it to be used not only on the web interface developed in this phase, but also on commercial GIS environments, in particular, by exploiting open source products, based on Oracle.

Another aspect that will be investigated, is the product code recompilation, to be used, regardless of the database engine used.

ACKNOWLEDGMENT

This work is framed within a R&D project financed by the Sardinia Region, called PIA, and conducted by the Company ConsulMedia Srl located in Viale Mattei n. 72, Oristano, Italy.

REFERENCES

[1] U. Falchi, Spatial database (DBT) IntesaGIS: a project proposal for the transition from DB10k to minor scales, proceedings of the 12a National Conference ASITA, L'Aquila 2008.

[2] S. Balley, C. Parent, S. Spaccapietra., Modeling geographic data with multiple representations, International Journal on GIS (IJGIS). v18 June. 329-354, 2006.

[3] C. Parent, S. Spaccapietra, E. Zimányi, The MurMur project: Modeling and querying multi-representation spatio-temporal databases, Direct Science, Elsevier, Information Systems 31 733–769, 2006.

[4] Regione Autonoma della Sardegna 2005, Implementazione e realizzazione di data-base topografici comunali

[5] C. Vangenot., Multi-representation in spatial databases using the MADS conceptual model, ICA Workshop on Generalisation and Multiple representation, Leicester 20-21 August, 2004.

[6] MurMur, Multi-representations and multiple resolutions in geographic databases, "Project 10723 - 1.1.2000 to 31.12.2002"

[7] Italian Ministerial Decree, November 10, 2011

AUTHOR PROFILE

Ugo Falchi He is tenured Assistant Professor at the Department of Sciences and Technologies, University of Naples "Parthenope" for Scientific and Disciplinary Group ICAR/06 - Topography and Cartography. Graduated with full marks in Civil Engineering at University of Cagliari, he obtained his Research Fellowship at University of Cagliari with a research titled: "The automatic cartographic generalization in GIS enviroment". He has participated to research projects financed by MIUR, UE, taking care himself of Geographical database, Cartography, GIS and Photogrammetry. He is a PhD Doctoral College Professor and Tutor in Geomatica, Navigation and Geodesy. He is author or co-author of more than 20 papers published in scientific journals or proceedings of National or International Conferences.