International Journal of Engineering & Technology, 7 (1) (2018) 65-69

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i1.8810
Research paper

IT tools for the management of multi - representation geographical information

Ugo Falchi *

Department of Sciences and Technologies, University of Naples "Parthenope" – Italy
**Corresponding author E-mail: ugo.falchi@uniparthenope.it*

Abstract

The goal of this research was the creation of software tools for managing instances of a multi - representation geodatabase, able to define multiple representations and topological constraints, in relation to modeled objects and structures according to the classification of the Italian national technical specifications of the November 10, Italian Ministerial Decree 2011. After the development of a conceptual scheme, encoded in its corresponding logical mode, various computer artifacts were designed and developed from scratch to perform the upload, management and display of data: a Scheme Designer, which allows users to define the logical model and 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; a DB Navigator, which is the web interface to the database multi - representation.

Keywords: *Conceptual Model; Geographical Database; Multi Representation; Topological Relations.*

1. Introduction

Storage and management of geographic information (geometric entities and attributes) can be done through different procedures. The main types of the database Management System (DBMS) used historically are the hierarchical, relational, and object - oriented types.
In this project, after the development of a conceptual scheme, encoded in its corresponding logical mode [1], a suite of software tools has been created to navigate, create and manage instances of the multi-representation geodatabase using the ORACLE RDBMS; they allow to define multiple representations, topological constraints and generic relationships and structure according to Italian national law. The goal is to overcome the limits of classical relational databases by integrating the existing reference model with the concept of multi - representation.
Several research projects (eg MurMur [2] and Cronogeograph [3]) have been developed over the last few years and a conceptual model of multi - representation geodatabase has been conceived to guarantee data processing with spatial and temporal components and with the simultaneous presence of four dimensions: structures, space, time, multiple representations. These dimensions are characterized by being orthogonal to each other. This property allows you to execute individual operations freely and independently of other dimensions.
Constraints, relationships, and metadata tables are transparently handled into the geodatabase, so the user is not required to know all the implementation details.
The final result can be used as a command line on Linux and Windows systems, or a GUI interface on Windows. The product was written in Java and consists of several executables.
The product does not intend to replace the current Relational Database Management System (RDBMS), but is integrated into Oracle with spatial extensions.
Some software products have been developed:
MR 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:

- MR Loader, which allows users to populate the database;
- MR GUI, which is a graphical interface to the tools and Schema Designer Loader;
- MR DB Navigator, which is the web interface to the database multi - representation.
- MR Schema Designer and MR Loader were written in Java (Oracle registered trademark); in order to simplify the operational management of work flows, a virtual machine has been created containing all the server-side software.

From the database point of view, it should be noted that:

- The database is consistent with national law;
- Topological rules are developed at the database level;
- Relations have been defined to link database tables;
- The multi - representation concept was developed.

2. Steady state

The workflow followed for creating a multi - representation database is as follows:
Step 1: Schedule log database schema: Definition of layers, themes, and classes to populate the database;

- Definition of representations for each class;
- Definition of topological constraints;
- Definition of relationships.

Step 2: MR Diagram Designer tool to implement the physical model based on the logic defined in step 1.
Step 3: MR Loader Tool to populate the ORACLE database.
Step 4: Explore the model with the MR DB Navigator tool.
Once step 3 is complete, it is possible to browse the ORACLE database through the ORACLE client or connect the web browser (MR DB Navigator) to its database.

It should be pointed out that a fixed database schema is not proposed, but have been implemented the tools for creating complex databases with the concept of multi-representation. The tools also work if this concept is not implemented, so to create generic geographic databases.
Database experts are asked to propose a database schema based on their experience or functional analysis of software or various requirements and customer requirements. Once defined this is implemented and populated with the help of the tools. So the database structure comparison and scheduling is not deleted, but is one of the most stringent elements since no database schema is valid at all.
It should be noted that the choices must come from a deep knowledge of the state of the art: how to transpose, organize, use the user's data, and the goals that we are aiming for the project.

3. Multi – representation database schema

Before using the developed software, it is necessary to implement the logic database schema compliant with national regulations [4].
This is an extremely delicate and expensive phase in which the geographic objects are defined and structured, specifying if we want to associate different representations changing shape, scale or both [5]. For each representation, a table containing the geometry and a set of attributes will be created, which can be imported for example from a shape file containing the object information.
An object (feature) will then consist of a series of tables in the different representations [8]:

- Shape
- Scale
- Time

A fundamental feature of the conceptual model [1] will be the respect of national norms and the architecture: layer, theme, class.
If you want to associate different forms (representations) in a class, then the classes that are part of it will be made up of geometries in which the topology differs (simple: polygons, lines, points; complexes: multipolygons, multilines, points).
Similarly, associating several scales to a theme, the classes that will be part of it will be geometries with different granularity that will specify the level of detail of the representation. Finally, it is possible to define classes that have only one representation.
Each class will have its own attributes.
During the conceptual scheme implementation, a series of meta - tables will be populated. For example, for each geographic table created, the representation, time, form, and scale will be stored in the database, as shown in table 1:

Table 1: XML File Example

Class	Representation	Form	Time
St01te01cl1	St01te01cl1_Plg_A_1k	Plg	.....
St01te01cl1	St01te01cl1_Arc_A_1k	Arc	.....
St01te01cl1	St01te01cl1_Pts_A_1k	Pts	.....
St01te01cl1	St01te01cl1_Arc_A_5k	Arc	.....

3.1. Topological rules

After defining the structure of the database, the topological relationships are structured between the geographic tables.
The main purpose of the topology is to define a series of spatial relationships between features present in one or more feature class. The definition and implementation of such relationships within the geodatabase allows for obtaining a data numerical model that is more and more close to objects present in the real world [6]. Among the main geometric properties and the spatial relationships that exist between the various elements present in the real world, we remember the relationships of touch, equal, disjoint and overlap. However topological rules may also be very complex in relation to the objectives concerned.
Here are some examples of topological rules:

- DISJOINT: The boundaries and interiors do not intersect.
- TOUCH: The boundaries intersect but the interiors do not intersect.
- OVERLAPBDYDISJOINT: The interior of one object intersects the boundary and interior of the other object, but the two boundaries do not intersect. This relationship occurs, for example, when a line originates outside a polygon and ends inside that polygon.
- OVERLAPBDYINTERSECT: The boundaries and interiors of the two objects intersect.
- EQUAL: The two objects have the same boundary and interior.
- CONTAINS: The interior and boundary of one object is completely contained in the interior of the other object.
- COVERS: The interior of one object is completely contained in the interior or the boundary of the other object and their boundaries intersect.
- INSIDE: The opposite of CONTAINS. A INSIDE B implies B CONTAINS A.
- COVEREDBY: The opposite of COVERS. A COVEREDBY B implies B COVERS A.
- ON: The interior and boundary of one object is on the boundary of the other object (and the second object covers the first object). This relationship occurs, for example, when a line is on the boundary of a polygon.
- ANYINTERACT: The objects are non-disjoint.

Each report is translated with a table and triggers are created to prevent topological rules being violated. This level of control ensures that the data entered on the server is in compliance with the specifications.
Relationships bind the classes so to respect the semantics. For example, within a theme, a class with a smaller granularity will have to fall within a class with greater granularity. Or two classes in which the shape changes will have to intersect, as shown in figure 1:

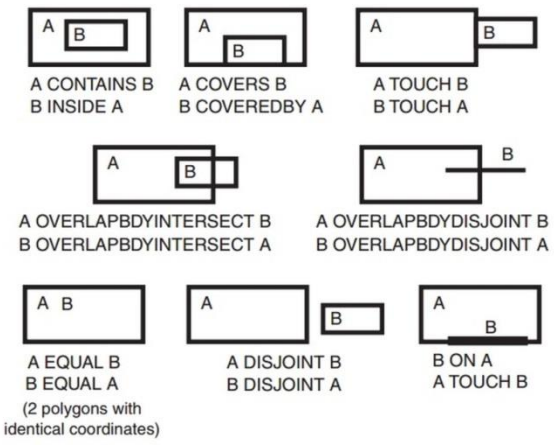

Fig. 1: Topological Relationships (Oracle.Com).

Formalizing existing relationships between objects, identifying those that meet certain spatial conditions, and providing insights into the formulation of hypotheses about the meaning of the reported relationships is an important aspect in the territorial information systems [7].
This aspect has been formalized into the workflow and inserted into procedures. From the analysis of existing RDBMS, the most complete is ORACLE which unlike other database engines, implements several triggers that cover a great case histories of topology. PostGIS approaches ORACLE in terms of the capabilities of spatial functions implemented but still does not have the same level of ORACLE space constraints.

In the table 2 there are some operators used by Oracle and PostGIS to check if two geometries have a certain topological relationship.

Table 2: Some Operators

	Oracle	Postgis
Disjoint	Sdo_Filter(G1, G2, 'Querytype=Join') = 'False'	Disjoint(Geom1,Geom2) = True
Contains	Sdo_Contains(Geom1, Geomy2) = 'True'	Contains(Geom1,Geom2) = True
Touches	Sdo_Touch(G1, G2) = 'True'	Touches(Geo1,Geo2) = True
Intersects	Sdo_Relate(Geo1, Geo2, 'Mask=Overlapbdyintersect') = 'True'	Intersects(Geo1,Geo2) = True
Equals	Sdo_Equal(Geom1, Geom2) = True	Equals(Geo1,Geo2) = True
....	.....	....

The user can choose the type of topological report (from a list) and then choose which tables (those defined above) will be subject to the constraint.

3.2. Relationship

In addition to topological relationships, it is possible to define generic relationships that can also be associated with tables that do not contain geometric references. Dbf files can import to create new alphanumeric tables; they can be linked by user-selected fields of geographic tables or generic tables.
In the context of database design, the entity - relationship model (also called entity - relationship model, entity – association model or model E-R) was used as a model for conceptual representation of data at a high level of abstraction.
The E-R model has long been one of the most robust approaches to modeling application domains in the IT; for this reason, it has often been used outside of the context of database design.
Associations (also referred to as relationships) are a link between two or more entities. The number of linked entities is indicated by the level of association; a good E/R scheme is characterized by a prevalence of associations with grade 2. It is possible to bind an entity with itself (through a ring association), to bind the same entities with multiple associations.
The user will be presented in a form where he will be asked to specify the operations to be performed during the data modeling, choosing which behavior to take over the handling of the tables.

3.3. General rules for data entry

Operations to manipulate information within the tables must be governed by structures that allow to comply with the semantics defined in the previous steps. Triggers are used to implement these controls.
Inserting new records within a table should only be allowed if the data you want to enter does not violate the (topological or generic) associated constraints. If not, the system will need to provide an appropriate error message.
Modifying records in a table must be subject to a check-up control of the existing relational constraints, so there is no inconsistent data. For example, giving two geometric tables A and B, where B elements must be included within the elements of A, and if R is the table that correlates the elements of A and B, modifying a B record should only be allowed if the new geometry continues to be within the A-record geometry attached via table R. Otherwise, the operation must be prevented and the result must be signaled by a message.
The deletion of records belonging to a table linked by a relationship must also produce an automatic deletion of the records that relate to them in the relational table and, in the case that it includes the semantics, also delete the records of the other tables to which are binded.

4. Server and clients requirements

In order to simplify the operation a virtual machine was created with all the necessary software package.
Modern computing is increasingly looking at new forms of distribution of workloads within the networks, both to allow the maximum use of each resource and to economize the latent management and costs of available resources. For this reason, the use of technologies such as cloud computing and virtualization is becoming more and more important.
"Virtualization" consists primarily to generate a simulated version of an existing resource. It is easy to understand that whatever the resource is available, this can be theoretically virtualized, both software and hardware. It is often said that virtualization mainly concerns the server environment.
The main advantages of virtualization are:

- Optimize the efficiency and availability of IT applications and resources;
- Provide greater longevity to applications;
- Delete the old one-server / single application-based model and run multiple virtual machines on each physical machine;
- Relieve IT administrators from expensive server management tasks.

The suite of developed products need also the implementation of an optimized client environment. Specifically, it need the implementation of a Java Virtual Machine.
The Java virtual machine, also known as Java Virtual Machine or JVM, is the virtual machine that runs programs written in bytecode.
Bytecode is a more abstract intermediate language of machine language, used to describe the operations that make up a program. It is called in this way because the operations have a code that occupies only one byte, although the length of the entire statement may vary because each operation has a specific number of parameters to operate. The parameters of these operations may consist of logs or memory addresses, just like the machine language.
An intermediate language such as bytecode is very useful to those who create programming languages because it reduces dependence on hardware and help to create the interpreters of the language itself.
Bytecode is generally produced by compiling sources written in Java language. However, it is also possible to produce bytecode from other languages; in fact, there are some partial or complete implementations of compilers that work in this way.
The most popular language among those who use the bytecode is Java. Java has both a virtual machine (Java Virtual Machine) that interprets the bytecode code, and is a just-in-time compiler that translates the bytecode into machine language.
JVM is defined by a specification, maintained by Sun Microsystems. Any system that behaves in a consistent manner with this specification should be considered as a particular implementation of JVM. There are software implementations for virtually all modern operating systems, both free and commercial. In addition, there are special implementations for particular hardware / software environments (such as cell phones and handhelds), and even hardware implementations.
The availability of Java virtual machine deployments for different operating environments is the key to Java portability, proclaimed in the slogan "write once, run everywhere". The virtual machine actually creates a homogeneous execution environment that hides to Java (and hence to the programmer) any specificity of the underlying operating system.

5. MR schema designer

The tool allows to create the multi - representation database structure based on two xml files: one containing the template and the other the specialization.
The template contains information about layer organization, themes, database classes, based on national guidelines. For each class, you also specify the list of attributes, type, and default geometry, as shown in table 3.
Specialization may include:

- default representations to be associated with each class;
- the specific representations to associate with a group of classes;
- topological relationships that link the classes.

Table 3: Simple relations

```
<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>
```

In addition, the association between the temporal dimension values [6] and the codes associated with them must be defined. It is also necessary to provide some mandatory information concerning the name of the geometric field, the field name used to relate different representations belonging to the same class and the limits of the bounding box used to contain the represented geometric objects.

6. General user interface: MR GUI

During the project, an interface (GUI) was developed to monitor the features and embedded in MR designers and MR Loader tools. The interface also allows you to navigate and interactively interact with the layers, themes and classes that will populate the multi - representation database.
The interface allows to create the ORACLE database based on an XML file that defines the multi - representation logic scheme. The same interface can be called to populate the ORACLE database. These operations can be solved by using the above tools directly. The goal of this product is to serve a non-specialist system user by simplifying the database creation and population operation.

7. MR DB navigator

Using a proprietary framework, an interactive navigator has been developed that allows interactive database browsing based on the multi - representation database schema.
The WEB interface to the multi - representation database created by the MR schema designer tool, suitably populated by the MR Loader, can be displayed on the WEB from the MR DB Navigator application.
MR DB Navigator is basically a navigable and interactive web catalog with a GIS viewer. MR DB Navigator connects directly to the ORACLE database and based on the data in the meta - tables and into tables (classes, representations, etc.) exposes its content in a user friendly environment, ie: layers, themes, classes, relationships and topological constraints.

8. Software testing

Along with the development activity, an intense testing was carried out.
The activity is aimed to achieving a good product quality standard. Especially when developing complex information systems, such as the one that is the subject of this work, test activity is crucial due to periodic reviews and quality check before being commissioned.
Testing was really important in the development cycle of this project and has involved valuable human resources. The tests were carried out by staff not involved in the development: this allowed to operate aseptically and ensures that the software abnormalities are detected more efficiently.
There are many test methods used to measure the robustness and reliability of the software, each with its features, its pros and cons.
Scott Barber has tried to classify them and tell us what they fit best in certain contexts:

1) Finding information and functional analysis by discussing with users,
2) Design and test planning,
3) Implementation and execution,
4) The analysis of the results and finally the report.

The author provides a classification of the test modes based on how the steps are followed.
A typical testing mode is called “waterfall model”. Thinking at development methodologies, this would be analogous to the "cascading method”, in which the phases flowing steadily downwards sequentially. The approach is suitable for situations that are easily manageable or where the context, times and modes are known. Small implementations like patches or bug fixes are situations where the approach work to the best.
The second category is an iterative path and the analogy with development methodologies goes on. The approach involves some cyclicity in the path and is necessary that the steps be taken up again in case of criticality. The concept of iteration introduces the necessary dynamism to handle more complex situations. Unlike the development method, testing iteration can stop at any time and bring the testers back to the design stage.
The third and final method, which is also the one adopted during the project, is the agile one, in which the sequential phase is dumped. Each activity, or transition to another, takes place only when it detects the need, without any formalism or obstacle.
Like the same development methodology, the approach presupposes the availability of almost constant users, it is costly in terms of commitment, but is best suited to highly critical situations whose context is unknown.

9. Results

The software test phase has been carried out throughout the project lifetime. In particular, a number of control operations have been carried out throughout the development phase with a continuous comparison with possible end users. The product (or semi-finished product, ie the product being tested) has been checked to detect any malfunctions and resolve them before the final release of the software.
An important part of the job was the functional analysis and the search for information useful to release a product that can meets the needs of the users. Several potential users have been involved in this phase, which have highlighted critical issues and situations to be considered. Based on these interactions and functional requirements, the test plan has been programmed.
Software tests can be performed either manually or automatically. Both systems have been used in the project.
In automated tests, we have tried the software options using predefined patterns. These tests involved controlling a series of normal use situations. Their job is to exempt manual testers from repetitive and "boring" jobs.
However, it was impossible to automate the processing methods and therefore manual tests aimed to find bugs that repetitive tests

did not highlight. Automatic input is to lighten the manual testers to get more time to run complex tests. Research has shown that it is not advisable (for time-resource loss) to automate a test if it is not repeated at least six times and that even the best automated test developers can not cover more than 30% of the total tests.
The aim of the software test is to make a product as free as possible from bugs. The main obstacle to the test phase is that the test may indicate errors, but cannot guarantee its absence. In principle, we point out that error-free software does not exist or is not certified.

10. Conclusions

This project laid the basis for an organic approach to the problem of implementing a geodatabase that complies with national technical regulations. An original approach has been proposed, capable of enhancing the features already present in some proprietary software with the simultaneous management of time and space parameters.
In addition, the concept of multi-representation within the structure of the data and the architecture of the tables has been introduced.
In the future, attention will be paid to similar tools that have been developed in Italy following the issuance of a specific law on the production and management of geographic information.

Acknowledgement

This research was part of the “New Geodetic Reference System and GNSS data quality” project supported by the University of Naples “Parthenope”. In loving memory of Prof. Raffaele Santamaria, the former Director of the Department of Sciences and Technologies, University of Naples “Parthenope”, for his fatherly help and scientific support of our research activities.

References

[1] Falchi U, “A Conceptual Model for the Management of Multi - Representation Geographical Information”, *International Journal of Engineering and Technology (IJET)*, (2016), vol 7 No 6, 2060-2068, ISSN: 0975-4024.
[2] Parent, C.; Spaccapietra, S.; Zimányi, E, The MurMur project: Modeling and querying multi-representation spatio-temporal databases, Direct Science, Elsevier, Information Systems, (2006), 31, 733–769. https://doi.org/10.1016/j.is.2005.01.004.
[3] De Fent, D; Gubiani; Montanari, A, “Granular GeoGraph: a Multigranular Conceptual Model for Spatial Data”, *Proceedings of the 13th Italian Symposium on Advanced Database Systems (SEBD)*, (2005), Bressanone (BL), Italy, 368-379.
[4] Italian Ministerial Decree, November 10, 2011, (G.U. n. 48, 27/02/2012 - Supplemento ordinario n. 37).
[5] Balley, S, Parent, C, Spaccapietra, S, “Modeling geographic data with multiple representations”, *International Journal on GIS (IJGIS)*, (2006), v18, 329-354.
[6] Vangenot. C, “Multi-representation in spatial databases using the MADS conceptual model”, *ICA Workshop on Generalisation and Multiple representation*, (2006), Leicester.
[7] Falchi, U, “Spatial data: from cartography to geodatabase*”, Geodesy and Cartography*, Volume 43, Issue 4, 142 – 146, (2017), https://doi.org/10.3846/20296991.2017.1412613.
[8] MurMur, “Multi-representations and multiple resolutions in geographic databases”, *Project 10723 - 1.1.2000 to 31.12.2002.*