ArgoCASEGEO - an open source CASE tool for Geographic Information Systems modelling using the UML-GeoFrame model

Jugurta Lisboa Filho, Maurício Fidélis Rodrigues Júnior, Jaudete Daltio and Victor de Freitas Sodré

{jugurta,mfrj,jdaltio,vsodre}@dpi.ufv.br

Abstract: This paper describes the development of an open source CASE tool, the ArgoCASEGEO, and its modular architecture. The ArgoCASEGEO tool allows the modelling of geographic database based on the UML-GeoFrame conceptual model that is specific for applications of Geographic Information Systems (GIS). The work also describes the transformation rules used by the mapping of the conceptual schema for the GIS GeoMedia representation. The data dictionary associated to the modeled schema is stored as a document XML/XMI, aiming its use by other programs.

Key Words: GIS, UML, CASE Tools, Reuse, Analysis Patterns.

1 Introduction

Geographic Databases (GeoDB) are collections of geographically referenced data, manipulated by Geographic Information System (GIS). In the Geoprocessing area, normally the user itself is the one who develops the GIS applications. Thus, redundancy and inconsistency are strong characteristics in the majority of the GeoDB, many times compromising the reliability of the system and, consequently, putting great public or private investments into risk. For that matter, the development of methodologies and tools that assist the GeoDB designers are essential to improve the quality of GIS applications.

A GeoDB should be designed following a database project methodology that includes conceptual, logical and physical design phases [4]. To elaborate the data schema in the conceptual phase, a data model must be chosen. Various specific models for GeoDB have been proposed in the past few years as GeoOOA [9], MADS [15], OMT-G [2], UML+SpatialPVL [1] and UML-GeoFrame [12].

Concluded the conceptual modelling, the next step - logical design - consists of the conceptual schema transformation into a data schema compatible with the data model of the GIS software that will be used. This stage of a conceptual schema transformation into a logical-spatial schema, and its settlement in a GIS software, can be made automatically by a CASE (Computer Aided Software Engineering) tool. Some of these conceptual models previously mentioned are supported by CASE tools, for example, Perceptory [1], REGIS [8], AIGLE [10] and Publisher Java MADS [15].

However, a difficulty faced by these tools is that each GIS implements its own data model. The lack of a standardized spatial data representation model causes problems on the data exchange between distinct GIS software, that include lost in data accuracy, consignment to the information quality, lost in attribute definitions and geographically referenced data. So, to make sure that the exchange of spatial data happens without the consignment to the information, a high degree of interoperability between the GIS is necessary. With this purpose, the OpenGIS Consortium is defining a standard model for data exchange that is supported by the Geography Markup Language (GML), to the geographic information storage [16].
This paper describes the ArgoCASEGEO architecture, an open source CASE tool for GeoDB modelling that supports the UML-GeoFrame model [12]. The conceptual schema elaborated by this tool is stored in XML (Extensible Markup Language) format, so can be easily accessed and used. This tool also provides an automatic generation module, able to generate data schemas to the most common formats usually found in commercial GIS. Moreover, the ArgoCASEGEO has a support for reuse based on analysis patterns [5].

This work is organized as following: section 2 presents a summary of the UML-GeoFrame model. Section 3 describes the using advantages of analysis patterns in GeoDB conceptual modelling. Section 4 details the development of the ArgoCASEGEO tool, showing its architecture and describing each implemented module. Final considerations and future works are described in section 5.

2 The UML-GeoFrame Model

The conceptual modelling of GeoDB based on the UML-GeoFrame model [12] produces an easy understanding conceptual schema, improving the communication between designers and/or users. Besides being used in the database schema elaboration, the UML-GeoFrame model is appropriate to the analysis patterns specification.

The GeoFrame is a conceptual framework that supplies a basic class diagram to assist the designer on the first steps of the conceptual data modelling of a new GIS application. The mutual use of the UML class diagram and the GeoFrame allows the solution of the majority requirements of GIS applications modelling. A geographic conceptual schema built based on the UML-GeoFrame model includes, for example, the spatial aspects modelling of the geographic information and the difference between conventional objects and geographic objects/fields. The specification of these elements is made based on the stereotypes set shown in Figure 1.

![Figure 1- Stereotypes of the UML-GeoFrame Model](image)

The first stereotype set (Geographic Phenomenon and Conventional Object) is used to differ the two main object types belonging to a GeoDB. The Geographic Phenomenon class is specialized in Geographic Object (△) and Geographic Field (△) classes, according to two perception ways of the geographic phenomena, described by Goodchild [7]. Non-geographic Objects are modeled on traditional form and are identified through the stereotype (●). The Geographic Object’s Spatial Component and Geographic Field’s Spatial Component stereotypes sets are used to model the phenomena spatial component according to object and field visions, respectively. The existence of multiple representations is modeled through the combination of two or more stereotypes on the same class. For example, a County class can have two abstraction ways of its spatial component, punctual and polygonal, that is specified by the stereotype pair (△△).
Finally, the stereotype \texttt{<<function>>} is used to characterize a special type of association that occurs when modelling categorical fields. According to Chrisman [3], in a structure of categorical covering the spatial is classified in mutually exclusive categories, that is, a variable has a value of category type in all the points inside a region. Figure 2 exemplifies the UML-GeoFrame model use showing a class diagram containing two themes: Education and Environment.

![UML-GeoFrame diagram](image)

Figure 2- UML-GeoFrame schema example

The Education theme, modeled as a UML package, includes three geographic phenomena classes perceived in the object vision (District, City and School), and the Student class that is a non-geographic object. In the Environment theme, three classes of geographic phenomena perceived in the field vision are modeled, Vegetation, Relief and Temperature, each one with its different types of spatial representation. This theme still includes the Vegetation Type class, which is modeled as non-geographic object, being associated to the Vegetation class through the stereotype \texttt{<<function>>}, that is, each polygon is associated to a vegetation type.

3 Analysis Patterns

An analysis pattern is a reuse mechanism that allows less experienced designers to reuse the knowledge of other specialists. According to Gamma and others, "a pattern presents the essence of a solution for a recurrent problem in a specific context" [6]. This generic patterns definition, used either to design patterns or analysis patterns, includes the basic ideas of a pattern. The expression "a solution to a problem" means that each pattern identifies a problem and presents its solution. The term "essence of a solution" means that only the essential elements are described, leaving the specific aspects to be detailed by the designer, once specific aspects are not normally reused. The term "recurrent problem" means that the patterns must be described for problems that had already occurred several times and will probably occur again. Finally, "in a specific context" means that the complete solution is valid for a particular context.

An analysis pattern describes a set of classes and the existing associations between them [5]. Analysis patterns can be seen, therefore, as a way to describe sub schemas of more complex projects, which occur frequently during the modelling process of many applications. The use of patterns improves, significantly, the development time of new applications, once the reuse occurs through the sub schemas and not through isolated classes. Generally, an analysis pattern presents the solution to the problem from a more suggestive than prescriptive form, supplying a model and the argument of why the solution is proposed this way, its advantages and disadvantages. According to Fowler, the pattern contribution that really matters is not the model supplied as solution, but the reasoning behind it [5].
The majority of the analysis patterns published up to now was designed, mainly, to solve commercial applications problems, for example, the patterns described in [5]. However, the analysis patterns idea can be used to increase the quality and the productivity in the development of non-conventional applications like GIS applications. The suitability of the analysis patterns use in the GeoDB modelling is shown in [11]. Examples of analysis patterns for GIS applications can be found in [13].

4 The ArgoCASEGEO Tool

ArgoCASEGEO is a CASE tool whose goal is to give support to the GeoDB modelling based on the UML-GeoFrame model. The data schemas elaborated using this tool are stored in XMI (XML Metadata Interchange) format, a syntax for conceptual schema storage, in XML documents [14].

In summary, XMI combines the definition, validation and sharing document formats benefits of XML with the specification, visualization, distributed objects and business-oriented models documentation and construction benefits of the UML visual modelling language.

A CASE tool is primarily a graphical drawing software. To avoid a great programming effort in developing a new graphical drawing tool, some existing graphical softwares were chosen to be used as starting point. This software must support the UML class diagram drawing and be extensible to support the stereotypes defined in the UML-GeoFrame model.

After analyzing some alternatives, the ArgoUML publisher was chosen as the base tool. Thus, the ArgoCASEGEO was developed as an ArgoUML software extension, a modelling tool found over an use license and open source distribution, developed in the Java language. Figure 3 illustrates the ArgoCASEGEO tool architecture, composed of four modules.

The Graphical Module allows the design of the conceptual schema, providing a set of constructors of the UML-GeoFrame model. The Data Dictionary Module stores the description of the diagram elements created by the designer. The Automatic Generation Module allows the transformation of the conceptual schema stored in the data dictionary in logical schema corresponding to some models used in commercial GIS. The Reverse Engineering Module, not yet implemented, will enable the designer to get conceptual schemas from existing GIS applications. The following sections describe each module at great length.
4.1 Graphical Module

The ArgoCASEGEO tool enables creation of diagrams that contain the constructors and stereotypes suggested by the UML-GeoFrame model. From this diagram the user can create its conceptual schema. An UML-GeoFrame conceptual schema supports three distinct class types: Geographic Object, Non-geographic Object and Geographic Field. The existing fields in the implemented class have name, attributes (and its respective domain), operations and symbols corresponding to the spatial representation type.

These classes can be related by relationships as generalization & specialization, aggregation, composition or association. In an association, the relationship name and the multiplicity of each class can be specified. The classes can be grouped to form a definitive theme, which is modeled by the UML’s Package constructor.

Figure 4 illustrates the ArgoCASEGEO environment having the analysis patterns Urban Street Mesh (Malha Viária Urbana), composed by the CrossRoad (NôLogradouro), Road Strech (TrechoLogradouro) and Street Mesh (MalhaViária) classes, which are subclasses of the Geographic Object (△) class and the Road (Logradouro) class, which is subclass of Non-geographic Object (4), that is, does not have a spatial representation.

Figure 4 - The ArgoCASEGEO graphical environment representing the analysis pattern Urban Street Mesh in the UML-GeoFrame model.

An UML-GeoFrame data schema can be saved as a new Analysis Pattern, which can be (re) used in new data schema, composing thus, the Analysis Patterns Catalogue. On the other hand, if the designer is starting a new project is interesting to look up in the catalogue in order to find existing analysis patterns. An analysis pattern can also contain an additional documentation, in which context, forces and examples of that pattern use can be described.
4.2 Data Dictionary Module

The dictionary stores the data schema created by the user. A schema has two data types, the graphical data (drawing) and the semantic data (classes’ names, attributes, associations’ multiplicities, etc). The semantic data are stored in the data dictionary, while the graphical data are stored in an ArgoCASEGEO file.

The data dictionary stores the conceptual schema in XMI format. Every class is delimited by a tag that contains the class name, its spatial representations and its features. The feature tag has two sub levels corresponding to the attributes and operations storage.

Figure 5 exemplifies the data dictionary to the Road Strech (TrechoLogradouro) class that has spatial representation of the line type, the Direction (SentidoCirc) attribute and the getDirection (getSentidoCirc) operation. The types used in this definition, including the attribute type, parameters and operations’ returned values are defined by the ArgoUML itself.

![Figure 5 - A UML-GeoFrame class in XMI representation](image)

The relationships between the classes modeled in the schema are stored in a specific tag that contains its name, related properties (that vary according to the type, association, aggregation or composition), its multiplicity and the classes’ references that participate in the relationship. From the generalizations definition vision, the internal tag is responsible for storing references to subclasses and super classes. Multiple inheritances are allowed. Finally, the package definitions are kept in a more external tag that includes everything previously described and has only its name as attribute.

4.3 Automatic Generation Module

After the conceptual modelling, the user needs to transform the elaborated schema into an effective implementation, characterizing a GIS application. As each GIS has its own data logical model, it is not possible to establish a single set of transformation rules to make the automatic generation of the logical-spatial schema. Thus, for each GIS the ArgoCASEGEO tool needs a specific Automatic Generation Module (AGM).

Two AGM have already been implemented in the ArgoCASEGEO tool. The first module transforms UML-GeoFrame schema to Shape format, used in the GIS ArcView 3.2 (ESRI). A second AGM, described in the section below, transforms conceptual UML-GeoFrame schema into logical-spatial schema of the GIS GeoMedia (Intergraph).

4.3.1 GeoMedia Automatic Generation Module.

The AGM-GeoMedia has as input the data dictionary identification that contains the conceptual schema to be transformed. To create a work environment in this software a connection with an existing database must be established. This connection will store the layers and the associated tables. For that matter, the AGM-GeoMedia creates a database Access
For each element of the conceptual schema a specific transformation rule is applied. These rules are described as following.

**Rule 1 - Packages:** A package is formed by a set of interrelated classes. Therefore, a database is defined for each package in the conceptual schema that will store all the themes generated by the mapping related to it. The file name and its location are supplied by the user.

**Rule 2 - Geographic Object Classes** (\(\Delta\)): Each class mapped as Geographic Object generates at least one layer inside the corresponding database, whose spatial representation is defined according to the representation’s stereotype. The attributes defined in the class are mapped as fields of a relational table.

**Rule 3 - Geographic Field Class** (\(\Delta\)): The GeoMedia is a vector software, therefore, it is not possible to carry through an automatic mapping of the phenomena that are modeled as field. However, according to Goodchild [7], the geographic fields’ representations are simply aggregations of points, lines and polygons, connected to spatial characteristics. A field with spatial representation of Isolines type, for example, can be mapped in a layer of line type, having a value associated to each line.

According to this analysis, the program considers some mapping options to the designer. Beyond the suggested attributes, the attributes of each class are also added to the table. Whether these approaches are not useful, the designer can choose not applying them and the geographic fields are mapped similarly to non-geographic objects.

**Rule 4 - Non-Geographic Object Classes** (\(\Delta\)): Each class modeled as Non-geographic Object generates directly one relational table. Objects are classified as Non-geographic exactly for having no spatial representation.

**Rule 5 - Relationships:** Relationships as association, aggregation and composition are made in accordance to the specified multiplicity. There are basically three types of multiplicities: one-to-one (1..1); one-to-many (1..*); and many-to-many (*..*). This mapping follows the same rules used to generate relational DBMS, which are well known [4]. Relationship of generalization-specialization type can also be translated using the same solutions applied in relational DBMS. Indeed, the spatial representation must be considered accordingly.

To exemplify the automatic generation process, the data dictionary shown in Figure 4 can be taken as input. The AGM-GeoMedia will create the logical-spatial schema shown in Figure 6.

**5 Conclusions**

The CASE tool use during the GIS applications development makes creation time smaller, which, consequently, reduces cost. Moreover, the geographic databases quality increases.

The ArgoCASEGEO tool was implemented to assist the designer to develop its GIS applications with higher quality, following a design methodology based on a conceptual model specific for geographic applications and on a reusable collection of analysis patterns. The documentation produced during the project (e.g.: conceptual schema and data dictionary) permits further references and visualization, which makes future system maintenance easier and the immediate generation of new versions of the application with the updates. The data dictionary storage in XML/XMI format allows the schema exchange and can be used by other applications, for example, analysis patterns discovery and search automatic tools.

As future works we can point the implementation of new Automatic Generation Modules (AGM-GML and AGM-TerraLib) and of the Reverse Engineering Module.
This work has been partially supported by CNPq (Brazilian National Research Council), the Brazilian governmental agency for scientific and technological development.

References