Specifying Analysis Patterns for Geographic Databases on the basis of a Conceptual Framework

Jugurta Lisboa Filho[1,2]
[1]Departamento de Informática
Universidade Federal de Viçosa
36571-000,Viçosa, MG, Brazil
Phone/Fax:+55 31 899-2394
E-mail: jugurta@dpi.ufv.br

Cirano Iochpe[2]
[2]Instituto de Informática
Universidade Federal do Rio Grande do Sul
91501-970, Porto Alegre, RS, Brazil
Phone: +55 51 316-6820 Fax: 319-1576
E-mail: iochpe@inf.ufrgs.br

ABSTRACT

Frameworks and Patterns are important instruments that enable the reuse of successful software solutions in recurrent problems. Geographic information systems, on the other hand, are usually designed by people with only little knowledge of database modeling techniques. Therefore, we believe that analysis patterns and conceptual frameworks can both facilitate and improve geographic database design in many organizations.

In order to be used by designers with such a wide variety of backgrounds (e.g., cartographers, biologists, and architects), analysis patterns and conceptual frameworks for geographic database design must rely upon an easy to learn and to use, semantically rich conceptual data model.

This paper introduces the **GeoFrame**, a conceptual framework that serves as a starting point for the conceptual modeling of geographic databases. It relies on an extension of the Unified Modeling Language (UML) that can also be used to define new analysis patterns for geographic applications. In addition, we present an example of an analysis pattern that was extracted out of a database schema which was itself derived from **GeoFrame**. This pattern can be identified in various urban GIS applications.

Keywords

Geographic databases; Conceptual design; Analysis pattern.

1. INTRODUCTION

The nineties have witnessed a great evolution in the geo-processing technology as well as a significant increase in the number of installed Geographic Information Systems (GIS). Such systems have been adopted either in public or in private administration environments.

The increasing expansion of the GIS base installed is related, on the one hand, to the price reduction of data capture systems and of geo-processing software. On the other hand, such popularization is due to the wide spread of the geo-processing culture and the consequent expansion of the set of applications being implemented.

Along with the increase on the number of GIS users, the search for existing geo-spatial digital data to be reused in new applications and/or new types of spatial analysis is also growing.

Although there is a tendency to increase geo-spatial data sharing, mainly with the aid of systems running through the Internet as, for example, FGDC`s Clearinghouse [7], little has been done to facilitate the sharing of modeling solutions of geographic databases. However, the unsystematic use of imported data from several (many times unknown) sources, can result in reliability problems in the system eventually leading to inconsistency and redundancy of data in the local geographic databases.

Another problem that can be observed is that when the data either do not exist, or are not appropriate for reusing, the respective existent geographic database design is in most cases also not reused, and new databases are developed from scratch [11]. For example, an information system for urban territorial tax collection, developed for a certain municipal district, could serve several other municipal districts (perhaps with minor adaptations), even though its related data can by no means be reused. In similar applications, many of the modeled entities tend to be repeated, alongside with the relationships among them. This leads us to research on new methods can support the reuse of geographic database design in an easy way.

While *design patterns* [9] can serve as reusability tools that provide for repeating design solutions for recurrent problems, *analysis patterns* [8,13] are defined to make the reuse of conceptual modeling solutions possible. *Analysis patterns* use terms and jargons from the application domain, while *design patterns* supply software implementation solutions for the modeled problems.

Due to the fact that GIS application designers have different backgrounds (e.g., cartographers, biologists, and architects), and usually have little knowledge of software development methods, geographic applications are strong candidates to the use of *analysis patterns* [21]. The work of these designers can be facilitated if database conceptual design, in several application domains, is documented and registered as *analysis patterns*.

An *analysis pattern* has to be described in such a way that any designer can easily understand the problem being presented and the solution being proposed. The existence of a common conceptual model for pattern specification is a fundamental requirement for the success of the adoption of *analysis patterns* by GIS application designers. Though, there is nowadays no consensus among GIS designers about a conceptual model for geographic applications.

Following the trend of research areas such as software engineering and databases, we chose the Unified Modeling Language (UML) [3] to use in geographic databases design. However it still needs some extensions as those proposed by several specific conceptual data models for GIS (e.g., GeoOOA [18], Geo-ER [12], Geo-OMT [4], MADS [23]).

Even some new CASE tools, like Perceptory [2] and REGIS [15], that have been treated as belonging to a second generation of CASE tools for GIS conceptual modeling [2], still present some limitations as, for instance, the lack of support to modeling geographic fields.

This paper presents **GeoFrame**, an object-oriented conceptual framework, that serves as a basis for class diagram construction to model geographic databases. A proposal of an UML extension for class diagram specification as well as *analysis patterns* description for geographic applications is also presented.

The remainder of the paper is organized as follows. Section 2 presents the structure of **GeoFrame**, describing each one of its classes. Section 3 explains how application class diagrams can be created starting from **GeoFrame** and using methods of schema simplification. In Section 4, an example of GIS *analysis pattern* is presented. The example was built based on **GeoFrame** and documented using the notation proposed in Section 3. Finally, Section 5 presents our conclusions.

2. GEOFRAME

GeoFrame is a conceptual framework that supplies a diagram of basic classes in order to assist the designer in the conceptual modeling of geographic phenomena as well as in specifying *analysis patterns* for geographic databases. **GeoFrame** class diagram is an evolution of a basic class library that has been developed to be used as a starting point for the conceptual modeling of geographic applications [19]. In [14], the reader can find a description of how this library has evolved over time. In [20], we discuss its use and evaluation within the development of a real GIS system called SIGPROGB.

Souza [25] defines a framework as being "a generic design in a domain that can be adapted to specific applications, being useful as a mold for constructing applications". This definition provides a more comprehensive view of the potentiality of a framework than the definitions presented by object-oriented programmers. For instance, Ralph Johnson defines a framework as being "a reusable project of a program, or a piece of a program, expressed as a set of classes" [16].

As a reusability instrument, a framework does not need to be implemented in a programming language to supply a partial solution for a family of problems. In this work, the concept of framework is used in a more generic approach expressing the idea of an unfinished conceptual design schema for a family of applications.

GeoFrame relies on various interesting proposals of GIS specific, conceptual data models [5, 12, 18, 22, 24, 26]. Each of the proposed models, though, presents limitations that can make the specification of *analysis patterns* more difficult. For example, the hierarchy of classes proposed in the MGeo model [26], later on extended to the MGeo+ model, mixes elements of different levels of abstraction (e.g, it includes classes like *GeographicDatabase*, *Symbol* and *OrientedLine*). A similar problem happens in the basic data schema proposed in the GISER model [24].

The GMOD model [22] supplies quite a complete class diagram. However, since the model is part of a computational environment for the development of geographic applications, its class diagram includes elements whose aim is to give support to the environment. For example, *Project* and *Document* are classes of the diagram related to the environment. Besides, processes, rules and relationships are all specified as classes, so that they can be kept as objects of the environment. These characteristics make the GMOD model unnecessarily complex for the definition of *analysis patterns* of geographic databases. Furthermore, the approach based on instances specification for user created class diagrams results in quite large final schema, thus reducing their readability. Despite these problems, several other ideas present in GMOD were incorporated to **GeoFrame**.

From the Modul-R [1] and GeoOOA [18] models, we have taken (and extended) the idea of simplifying schemas through the use of graphical symbols (see Section 3).

2.1 GeoFrame Class Diagram

GeoFrame was defined in according to the rules of the Object Orientation formalism. This was done using the graphic notation of the UML class diagram [3]. The choice of UML was based on its wide acceptance by designers and by software industry in general, and on its tendency of to become a standard language for object modeling. One of the main requirement for *analysis patterns* specification is to use a well-known, easy to understand language. UML has demonstrated to be a good answer to that requirement as can be noted through the facilities provided by new GIS CASE tools (e.g., Perceptory [2] and REGIS [15]).

Figure 1 shows **GeoFrame** class diagram. There are four main classes: GEOGRAPHICREGION; THEME; NONGEOGRAPHICOBJECT; and GEOGRAPHICPHENOMENON. They generalize, in a high level of abstraction, the elements of a geographic data schema.

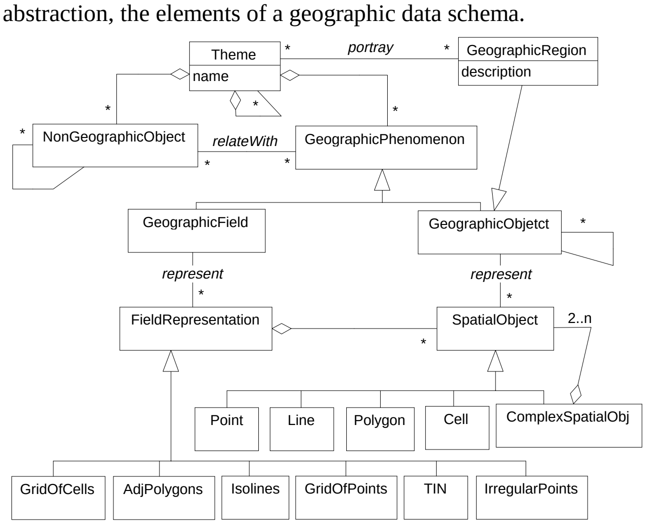

Figure 1. GeoFrame classes diagram

The classes THEME and GEOGRAPHICREGION are the basis of any geographic application. All geographic applications have as their main objective the management and the manipulation of a data set for a certain region of interest, constituting a geographic database. For example, in an urban GIS application the city's area can be specified as being the geographic region of interest. For this geographic region, one can imagine that the following themes might be defined: limits of the urban zone; road network; neighborhoods; buildings (e.g., schools, hospitals); public transportation; and zones of garbage collection. It is possible that some additional themes need to be defined for some sub-areas. This would be the case, for instance, of special safety and emergency zoning for the downtown area.

Thus, a collection of themes can be specified for each geographic area. The organization of the geographic information in themes is directly related to the way the user uses it to both organize and process his/her geographic information. We have taken into account that before the advent of computerized GIS, the user of this kind of information (normally in the form of analog maps) already classified data according to some specific criteria in order to reduce the complexity of the system.

The conceptual specification of a theme does not necessarily entail the implementation of a data layer in a GIS. A theme defined during conceptual design can lead to the implementation of several layers. This issue depends on the software used as well as on the user's needs. Due to the limitations of some current GIS, a simple theme as *Rivers* can generate more than one data layer, for example, one containing only linear spatial objects and another containing polygonal spatial objects. These details should be hidden from the user during conceptual design. Therefore they will not be discussed in this paper.

Another advantage of using the concept of theme, in the conceptual schema, is that it works as a complexity reduction mechanism for large database schemas. Geographic applications with hundreds of modeled entities (e.g., SIGPROGB [20]) are quite common. The use of themes allows the designer to divide the schema into cohesive sub-schemas, in which classes that are strongly related to each other are put together. In large projects, groups of related themes can be grouped in a more generic theme, thus forming a hierarchy of themes.

In a geographic database, as in any information system, there exist some objects with no geo-referencing, i.e., conventional objects. In **GeoFrame**, they are considered instances of subclasses of NONGEOGRAPHICOBJECT.

The abstract class GEOGRAPHICPHENOMENON generalizes any phenomenon whose location in relation to the earth's surface is being considered. For example, a land parcel could be considered as an instance of GEOGRAPHICPHENOMENON if its spatial attributes were to be represented in the database.

Geographic phenomena and non-geographic objects can be related to each other (Figure 1, *relateWith* association), as in the case that "every parcel belongs to some owner". The modeling of that kind of relationship allows the data stored in a GIS to be integrated with other information systems.

Geographic phenomena are perceived by users according to the view dichotomy of fields and objects [10]. These two views cause geographic phenomena to be modeled in different ways. The classes GEOGRAPHICFIELD and GEOGRAPHICOBJECT specialize the class GEOGRAPHICPHENOMENON, allowing to the designer to specify in a distinct (yet integrated) way, the geographic fields and objects, respectively.

The abstract class GEOGRAPHICOBJECT is a generalization of all classes of the domain that are perceived in the object view. Those classes representing geographic phenomena that can be individualized (possessing an identity and characteristics that can be described through a set of attributes) are included in this case. Classes such as *Parcel*, *River*, *Highway*, and *City* should be modeled as a specialization of the GEOGRAPHICOBJECT class.

It is important to notice the difference between the concept of geographic object, with is a geographic phenomena perceived in the object view (in opposition to the field view) and the concept of object, the basic constructor of object-oriented models.

The *relateWith* association (Figure 1), inherited from the GEOGRAPHICPHENOMENON class, can be specialized to represent the possible associations between geographic objects and non-geographic objects. Figure 2 shows a sub-schema of an urban GIS in which the *Parcel* class, a subclass of GEOGRAPHICOBJECT, is associated with both the *Owner* and the *TaxType* classes, subclasses of NONGEOGRAPHICOBJECT, as well as with the *Block* and *Street* classes, subclasses of GEOGRAPHICOBJECT.

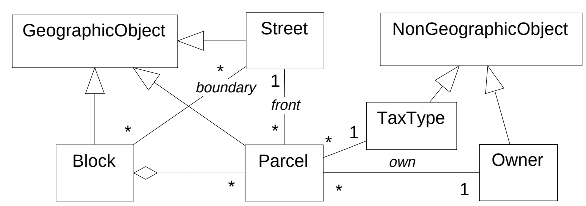

Figure 2. Example of *relateWith* relationships

In its turn, the abstract class GEOGRAPHICFIELD generalizes phenomena that fit in the field view. Geographic fields are modeled as functions on a variable. Some fields refer to variables that are distributed on the surface, in a continuous way (e.g., *Relief*, *Temperature* and *Soil Covering*), and other fields refer to variables which are distributed in a discrete way (e.g., *Population* and *Epidemic Occurrences*) [22].

According to Câmara [5], a geographic field can be mathematically defined by a mapping that abstracts the space distribution of a variable *V* on an area *R* of the earth's surface. Thus, a geographic field can be modeled as a class of functions, with the following attributes: *domain*, whose value is a sub-area of *R*; *image*, whose value belongs to the domain *V*; and *mapping*, whose value is the mapping *f* of *R* in *V*. Thereby, the set of values assumed by *V* in the sub-areas of *R* determine the class to which the geographic space *R* belongs. For example, a geographic field *SoilCovering* can be characterized by the set of values *V* = {*NativeForest*, *Culture*, *Reforestation*, *Pastures*, and *UrbanArea*}. For any point, inside the area *R*, there is a corresponding value element in the set *V*.

Thereby, a geographic field can be conceptually modeled as subclass of GEOGRAPHICFIELD. Some geographic field types, the so-called categorical fields [6], have an association with the class that represents the image of the field mapping function, a subclass of NONGEOGRAPHICOBJECT. This association is a specialization of the *relateWith* association for geographic fields. Fields whose the function *image* is composed by a set of numeric values do not specialize this association. For example, the *Relief* geographic field has the set of the real numbers as its image.

To Kemp [17], there are many philosophical questions in understanding of geographic fields. Some of these open questions include: "Do the fields only exist when they are measured? Do the fields possess their own characteristics? Is the attribution of a certain value to an area real, or is a conceptual mechanism addressed to a specific objective?". Despite the fact that these questions still do not have a generally agreed answer, the definition of a field as a function in a domain which is a subset of the space-time domain is enough for computational purposes.

In a GIS, the implementation of the spatial representation and relationships among the geographic objects is based on spatial data structures. For example, the location of a pole can be stored through a pair of geographic coordinates (x,y). Yet, the location of a street intersection should be stored in a network node data structure. Besides the coordinates (x,y) it is necessary to store topological information about the connectivity of this node with the arcs of the network that represent the streets.

The choice of the best data structure to implement the spatial representation of each geographic phenomenon is a task to be performed after the conceptual project, and is not to be considered at this level.

One of the fundamental principles of conceptual modeling is that a conceptual schema should only contain the elements of the domain, discarding implementation aspects [23]. The goal of including information on spatial objects related to the geographic phenomena in the conceptual schema is to allow the modeling of the spatial component of each phenomenon.

The information about geographic phenomena has three components: attribute, space and time [6]. These three components must be abstracted from the reality during the conceptual modeling process. Usually, in non geographic applications, only the component attribute (and in some cases the time component) is considered.

Therefore, when one considers points, polygons or grids of cells in **GeoFrame**, he/she is considering the forms of abstraction of the geographic phenomena spatial component, but not the way in which they will be structured in the database. Nevertheless, the majority of the currently available GIS supplies very similar data structures for the storage of these abstract constructors.

For the purpose of modeling, the main concern is to determine, for example, if a *pole* will have punctual dimensions or if the *pipe* will have linear dimension. It is not necessary, however, to consider that the pipe will be stored by means of an arc directed in a vector structure with topology.

Some geographic phenomena can present complex spatial dimensions, that is, dimensions composed by other spatial objects (e.g., an archipelago). The SPATIALOBJECT class generalizes classes that are used for specification of the spatial component representation of the geographic phenomena in object view. They are POINT, LINE, POLYGON, CELL and COMPLEXSPATIALOBJ. The SPATIALOBJECT class was defined based on the *Composite* design pattern [9].

Notice, in Figure 1, that not even the possible associations among the classes POINT, LINE and POLYGON are represented in the **GeoFrame** class diagram, because any of these associations might not be true according to a specific GIS implementation.

The spatial aspects of a geographic field are abstracted in a different way from the spatial aspects of a geographic object. Chrisman [6] describes some geographic models through which the spatial component of the geographic information can be abstracted. The models presented by Chrisman, which are appropriate to modeling phenomena in the field view, can be summarized in the six spatial models described by Goodchild: grid of cells; adjacent polygons; isolines; grid of points; irregular triangular network; and irregularly sampled points. In **GeoFrame**, these models correspond to the subclasses of the FIELD-REPRESENTATION class, namely GRIDOFCELLS, ADJPOLYGON, ISOLINES, GRIDOFPOINTS, TIN and IRREGULARPOINTS.

In a GIS, those models will be, at a later stage, implemented through the raster and vector representation models [10]. Each one of the six geographic field representation models can be implemented either in the raster or in the vector model, although some mappings are more natural. For example, the mapping of a field whose component has been abstracted by means of a grid of cells should be implemented in the raster model.

A geographic field can have its spatial component abstracted in different ways, i.e., through more than one of those field representation models. For example, the field *Temperature* can be abstracted by means of irregularly sampled points or by means of isolines. A similar situation happens with the geographic objects whose spatial components can be perceived, sometimes by alternative geometry, other times by double geometry, depending on aspects like the scale with which one intends to capture the spatial dimension of each phenomenon. The possibility of having multiple representations is shown in Figure 1, through the (1:n) *represent* association.

2.2 Relationships in GeoFrame

Figure 2 shows that geographic phenomena and non-geographic objects can be related to each other. The relationships among classes that are represented as associations in the diagram, allow for the specification of integrity constraints that should be hold in order for the database to remain consistent.

The relationships can be divided into three categories: *semantic*, *spatial* and *temporal*. Semantic relationships complement the description of knowledge regarding the descriptive aspects of the phenomena (e.g., the streets building blocks and a block can contain several parcels). Spatial relationships establish associations among the locations of the phenomena (e.g., the spatial limit of a block contains the limits of its parcels). The modeling of the temporal aspects is under investigation, and will be treated in a separate paper.

Some GIS conceptual models propose new modeling constructors for spatial relationships. For example, the GeoOOA model [18] defines three new kinds of constructors for spatial aggregation: *covering*, *containment* and *partition*. The Geo-ER model [12] defines only one constructor for spatial aggregations. Even though these constructors allow the specification of spatial integrity constraints, they present two disadvantages. First, they make it more difficult for users/designers to understand the model and, second, they are incomplete, i.e., they do not allow several other relationships to be modeled through similar mechanisms. For example, in the GeoOOA model it is possible to specify that the location of an industry should be inside (*containment* structures) of an industrial district area, but at the same time there is not a constructor, as detailed as this, to specify that this district can be crossed by an avenue or that it can be adjacent to a river.

No matter how hard the designer tries to specify the restrictions of spatial integrity, he/she will always have a much bigger volume of relationships that will not have been specified. So, the relationship modeling approach in **GeoFrame**, that has been adopted in the development of our systems [19, 20], is just modeling the semantic relationships, leaving the spatial relationships to be considered in the data acquisition and data quality assessment processes. Thus, we only use the relationship modeling constructors from the UML, i.e., association, specialization, aggregation and composition [3].

3. BUILDING CONCEPTUAL SCHEMA FROM THE GEOFRAME

GeoFrame provides a basic class diagram for the elaboration of conceptual schemas of geographic databases. In this section some modeling guidelines are described, as well as a proposal for simplification of schemas based on stereotypes, a UML mechanism of extensibility [3].

A conceptual schema is built starting from the specialization of **GeoFrame** classes. The geographic database conceptual modeling, using **GeoFrame**, is accomplished according to a top-down approach composed of three stages. Initially, for each geographic area considered, several themes (and sub-themes) are identified. In the second stage a class diagram is specified for each identified theme. The specification of the associations among classes of different themes is made still in that stage. Finally, the analysis and modeling of the spatial component of each geographic phenomenon is done. Each one of these phases is described in more detail in the following sections.

3.1 Theme Diagram

As described in Section 2, each geographic region is portrayed by zero or more themes. However, a theme can also be associated with more than one geographic region. A theme is specified as an aggregation of geographic phenomena classes, of non-geographic classes and of other more specific themes.

To increase the readability of the resulting schemas, themes are not modeled as subclasses of the THEME class, but as an UML constructor called *package*. A package is composed of a set of UML elements that could be of any type as, for example, classes, associations and other packages [3].

Figure 3 shows an example of theme hierarchical diagram defined for an application of environmental control (SIGERCO System) [19], developed for the coast of the Rio Grande do Sul state, Brazil. As it can be seen in Figure 3, themes are associated with an object, an instance of GEOGRAPHICREGION class. For each region of a project the important themes should be specified. The theme diagram was based on the Geo-OMT model [4].

3.2 Specifying Geographic Phenomena and Non-Geographic Objects

After the theme identification stage, each theme should be refined. In the object-oriented approach, a geographic database schema is represented by the class diagram, which describes the geographic phenomena, the non-geographic objects and the possible relationships among them.

Each class identified in the domain should be modeled as a subclass of one of the following **GeoFrame** classes: NONGEOGRAPHICOBJECT, GEOGRAPHICFIELD or GEOGRAPHIC-OBJECT. To avoid visually overloading the diagram, stereotypes are used as a schema simplification mechanism. A stereotype extends the UML vocabulary (i.e., the metamodel) allowing the user to create new constructor types that can be used like any other element of the language [3].

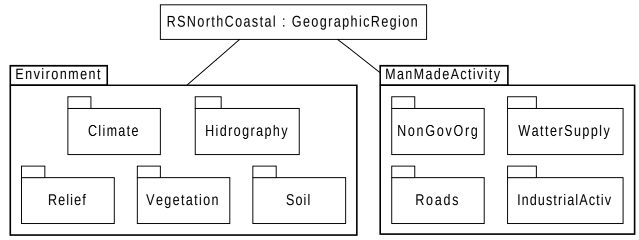

Figure 3. An example of theme diagram

To substitute the generalization relationships between the domain's classes and the **GeoFrame** classes, three stereotypes were defined (Figure 4). Figure 4 also shows examples of classes using these stereotypes, namely *NonGovOrg* (Non-governmental Organization), subclass of the NONGEOGRAPHICOBJECT class, *LandUse*, subclass of the GEOGRAPHICFIELD class and *River*, subclass of the GEOGRAPHICOBJECT class.

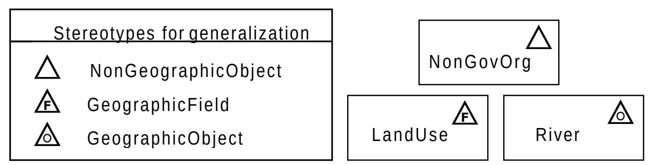

Figure 4. Stereotypes[1] for generalization

The specification of these stereotypes will make the stage of mapping between the conceptual and logical design simpler.

3.3 Modeling the Spatial Component of Geographic Phenomena

In **GeoFrame** every geographic field or object can be represented by multiple instances of the FIELDREPRESENTATION and SPATIALOBJECT classes, respectively (Figure 1). A geographic phenomenon can have multiple representations for several reasons. Among them, we can mention the need of multiple scales, users with different views of a same phenomenon and temporary versions.

The possibility of having multiple representations for a geographic phenomenon is modeled through different associations between geographic phenomena and possible forms of abstraction of their spatial components (*represent* association in Figure 1). Bédard [2] proposes a set of possible variations for the modeling of the spatial component as, for example, optional geometry, simple and complex aggregates, alternative geometry and derived geometry. In **GeoFrame** these variations can be specified by the designer through the free combination of different stereotypes in the same class. However, this needs additional documentation in the data dictionary.

A second set of stereotypes (Figure 5) replace the great quantity of associations that results from modeling spatial components of the geographic phenomena. The semantic of this type of stereotype is the replacement of explicit associations between geographic

[1] These stereotypes are defined as characters of a special MS-Windows font, called SIGMODA, available for download at http://www.inf.ufrgs.br/gpesquisa/sigmoda. Some stereotypes were reused from the CASE tool Perceptory [2].

phenomena and their spatial representations. Stereotypes also indicate geometric forms of spatial representations.

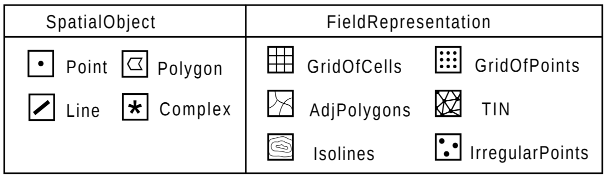

Figure 5. Stereotypes for *represent* association

Figure 6 shows part of a conceptual database schema for an urban application developed from **GeoFrame.** In this example the mechanisms of schema simplification are showed. Some classes do not present stereotypes (e.g., *BuiltParcel*) because the respective relationships are inherited from its super-class.

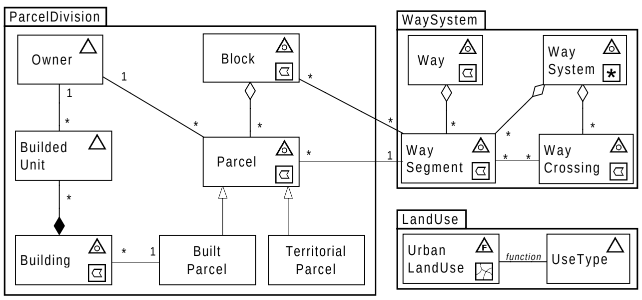

Figure 6. Example of class diagram using GeoFrame

The use of graphical symbols as a schema simplification mechanism was presented, originally, by Bédard and Paquette [1]. This technique has been used in several GIS conceptual models as, for example, GeoOOA [18], Geo-OMT [4] and MADS [23], as well as in the CASE tools Perceptory [2] and REGIS [15]. However, these models present some limitations that make them inadequate for adoption as models for GIS *analysis patterns* description. Among these limitations, we can cite the following: (1) the models Modul-R and MADS do not support geographic fields modeling, while the GeoOOA model proposes a mistaken solution, in which the concepts of geographic field and raster structures are used in an ambiguous way; (2) the over use of graphical symbols reduces the readability of the resulting schemas in the models Geo-OMT and MADS; (3) the GeoOOA model (including the CASE tool REGIS) does not support the specification of multiple representations of the spatial component.

GeoFrame approach allows the construction of easily readable schemas either by designers or by users, besides supporting the main requirements for the conceptual modeling of geographic information (e.g., geographic field modeling). An important contribution of this work is the stereotype use as a replacement for generalization-specialization relationships. This idea can be adopted as a simplification mechanism for schema containing complex class hierarchies.

4. EXTRACTING GIS ANALYSIS PATTERNS FROM GEOFRAME DERIVED SCHEMAS

From a careful analysis of existing geographic database designs, we can extract parts of class diagrams with potential to be transformed into *analysis patterns* to be reused in other geographic applications.

An *analysis pattern* presents, apart from the classes diagram proposed as the problem solution, a set of information (e.g., context, example) that helps the designer to understand the pattern and to evaluate its adequacy as a solution to a given problem. Thus, the description of a new pattern requires an additional effort, mainly from experienced designers, in creating a library of GIS conceptual modeling solutions.

An *analysis pattern* should just present the essence of the problem, discarding the specific particularities of each application. The following sub-section presents, as an example, a simple *analysis pattern* extracted from the schema presented in Figure 6.

4.1 An Example of Analysis Pattern

Pattern name: Urban Way System

Problem: How to model an urban way system?

Context: Almost all the cities in the world present a similar organization pattern. A city is subdivided in ways (e.g., streets, avenues, paths) forming a network of ways. Each way should have an identification code (e.g., ZIP code) and a name, apart from being divided into several segments. A way's segment corresponds to the space between two other way's segments that intercept or cross it. Each way's segment can be identified by a sequential number (associated to the way's identification) and has as attribute an interval of parcel numbers. The set of way segments and connections constitute a way system.

Solution: Figure 7 shows the pattern`s class diagram.

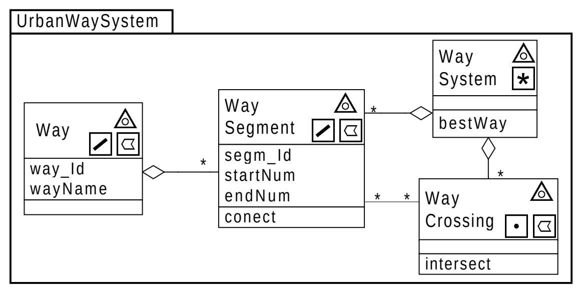

Figure 7. Urban Way System analysis pattern

For each geographic phenomenon the pattern specifies only the more generic attributes and operations, which are extended in accordance with each specific application. On the other hand, the possible abstractions of the spatial components are specified in the pattern. For example, the spatial component of the *Way* class is specified in Figure 7 as being linear or polygon, even though the same class is specified with distinct dimensions in the examples of Figure 6 and Figure 8. Due to space limitations, the specification of the attributes and of the operations were omitted in the examples.

Example: Figure 8 exemplifies the *analysis pattern* use in a public transportation application. The example was extracted from a conceptual schema presented in [4].

5. CONCLUSIONS

Frameworks and patterns have been used as tools for solutions reuse in recurrent problems, by the object-oriented system designers' community. The authors believe that the use of *analysis patterns* can facilitate and improve the geographic applications design, mainly due to the wide domain of GIS database and also to the great number of applications designed by users who are not familiar with modern techniques of modeling.

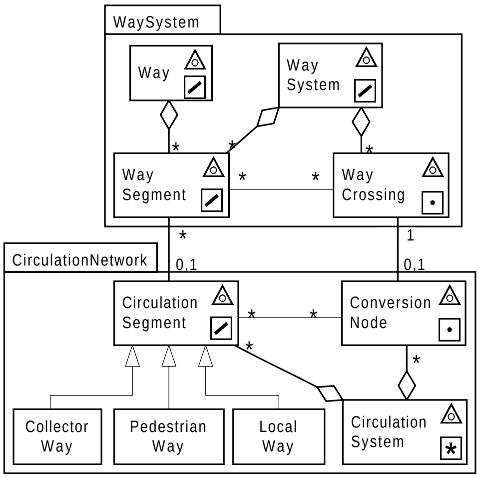

Figure 8. Example of use of the analysis pattern

One of the requirements for the success of the use of *analysis patterns* in geographic applications is that the patterns should be described in a language that is semantically rich but easy to understand. This paper presented **GeoFrame**, a conceptual framework that serves as a base for the conceptual modeling of geographic databases. Furthermore, an extension of the UML graphical notation for specification of geographic databases conceptual schemas, as well as for the description of GIS *analysis patterns* based on **GeoFrame** was presented.

Another contribution of this paper is the proposition of a method for the conceptual modeling of geographic phenomena that are perceived in the field view. Most of the models proposed in the literature deal only with phenomena in the object view. The method stresses the existence of different abstraction levels, in which the conceptual constructors should not be confused with implementation aspects. Thus, it must be clear that the inclusion of classes such as *Point*, *Polygon*, *Grid of Cells* in the class diagram, is related to the form of perception of the spatial component of each geographic phenomenon and not to the data structures used to store those data in the GIS.

6. ACKNOWLEDGMENTS

The authors thank Willian Wazlawik for the confection of the graphical symbols for SIGMODA font and Karla Borges for supplying a series of conceptual design examples in urban GIS applications. This work was developed in part with SIGMODA project funds, a research project supported by CNPq.

7. REFERENCES

[1] Bédard, Y.; Paquette, F. Extending entity/relationship formalism for spatial information systems. in Proceedings of AUTOCARTO, 1989.

[2] Bédard, Y. Visual modeling of spatial databases: towards spatial PVL and UML. Geomatica, 53, 1999.

[3] Booch, G.; Jacobson, I.; Rumbaugh, J. The unified modeling language user guide. Addison-Wesley, 1998.

[4] Borges, K. A. V. Geographic data modeling: an extension of the OMT model for geographic applications. Belo Horizonte: Fundação João Pinheiro, 1997. Master Thesis. (in Portuguese)

[5] Câmara, G. et al. Anatomy of geographic information systems. Campinas: UNICAMP, 1996. (in Portuguese)

[6] Chrisman, N. Exploring geographic information systems. New York: John Wiley & Sons, 1997.

[7] Federal Geographic Data Committee. Content standard for digital geospatial metadata. Washington, 1997.

[8] Fowler, M. Analysis patterns: reusable object models. Menlo Park: Addison Wesley Longman, 1997.

[9] Gamma, E. et al. Design patterns**:** elements of reusable object-oriented software. Reading: Addison Wesley, 1994.

[10] Goodchild, M. F. Geographic data modeling. Computers & Geosciences, 18, 4, 1992.

[11] Gordillo, S.; Balaguer, F.; Neves, F. Generating the architecture of GIS applications with design patterns. in Proceedings of ACM-GIS, Las Vegas, 1997.

[12] Hadzilacos, T.; Tryfona, N. An extended entity-relationship model for geographic applications. SIGMOD Record, 26, 3, 1997.

[13] Hay, D. C. Data model patterns: conventions of though*t*. New York: Dorset House Publishing, 1995.

[14] Iochpe, C.; Lisboa F., J. BCH/GIS: a basic class hierarchy to support GIS conceptual design. in Proceedings of International Conference on Modeling Geographic and Environmental Systems with GIS, Hong Kong, 1998.

[15] Isoware. CASE-Toll REGIS. 1999. http://www.isoware.de.

[16] Johnson, R.E. Documenting frameworks using patterns. in Proceedings of OOPSLA, Vancouver, 1992.

[17] Kemp, K.; Vckovsky, A. Towards an ontology of fields. in Proc. Geocomputation, Bristol, U.K., 1998.

[18] Kösters, G.; Pagel, B.; Six, H. GIS-Application development with GeoOOA. IJGIS, 11, 4, 1997.

[19] Lisboa F., J.; Iochpe, C. Adapting the OMT object model for GIS application conceptual modeling. in Proceedings of SEGEO, Rio de Janeiro, 1996. (in Portuguese)

[20] Lisboa F., J.; Garaffa, I. M.; Iochpe, C. Conceptual data models for geographic applications: an experience with an inter-institutional GIS. in Proceedings of Brazilian Symposium of Geo-processing, São Paulo, 1997. (in Portuguese)

[21] Lisboa F.,J.; Iochpe,C.; Beard,K. Applying Analysis Patterns in the GIS Domain. in Proceedings of SIRC`98, Dunedin, NZ, 1998.

[22] Oliveira, J. L.; Pires, F.; Medeiros, C. B. An environment for integrated modeling and analysis of geographic information. GeoInformatica, 1, 1, 1997.

[23] Parent, C. et al. Modeling spatial data in the MADS conceptual model. in Proceedings of SDH`98, Vancouver, 1998.

[24] Shekhar, S. et al. Data models in geographic information systems. Commun. ACM, 40, 4, 1997.

[25] Souza, C.R.B. A framework for cooperating diagrams editor based in annotation. Unicamp, 1998. Master Thesis. (in Portuguese)

[26] Times, V. C.; Salgado, A. C. An object-oriented modeling for geographic applications. in Proceedings of Brazilian Database Symposium, São Carlos-SP, 1994. (in Portuguese)