Modeling spatial constraints in conceptual database design of network applications

J. Lisboa Filho & S. M. Stempliuć

Federal University of Viçosa, Minas Gerais, Brazil

ABSTRACT: This paper describes the problems found in geographic databases design during the conceptual modeling of real world elements that are related by a network structure. It is believed that is possible to reach a better quality of the geographical data if spatial integrity constraints about the elements on a network are specified in a conceptual level. Hence, the purpose of this paper is to expand the UML-GeoFrame conceptual data model with specific constructors to allow modeling network elements, added by a formal specification of integrity constraints. The extension of the UML-GeoFrame data model includes the definition of stereotypes to allow an easier reading of the schema and the use of the Object Constraint Language (OCL) to a complementary specification of the spatial integrity constraints. An example of an electric power company diagram is used to illustrate the extension proposed.

1 INTRODUCTION

The quality of spatial data has become an issue of great importance because of the great amount of data manipulated by the Geographic Information System (GIS). According to Cockroft (1997), “the users of most spatial data sets have no idea of the accuracy of the data contained within them. They base their subsequent analysis using the datasets on the assumption that the data is error free or that errors are kept to an acceptable level”.

In many applications, the quality of the data source has a great importance to provide accurate results to queries. This problem is specifically significant in the GIS context due to the variety of data, sources and measuring techniques involved in their collection, and due to the high precision levels expected by the users of the applications (Elmasri & Navathe 2005).

Integrity constraints specification in database has therefore the purpose of certifying the quality of the data stored. In any state of the database, all the constraints must be satisfied to make sure that the data adhere to the expected quality level.

The integrity constraints issue is well consolidated in conventional databases areas. However, as the data used in GIS have characteristics related to location and spatial relationships, a new challenge was established to the integrity constraints subject. It’s necessary, hence, to establish rules capable to deal with the peculiarities of the spatial data, especially with the spatial objects relationships.

The modeling of spatial integrity constraints has been concerned by many authors of conceptual data models specific to GIS (Borges et al. 2001, Kösters et al. 1997, Lisboa Filho & Iochpe 2008, Parent et al. 2006). However, an important issue that do not receive the necessary attention regards the integrity constraint modeling that involves network elements, such as elements of roads and waterways, telephony or energy. One of the models that describes this issue is the GeoOOA (Kösters et al. 1997). Other recent models as the OMT-G (Borges et al. 2001), UML-GeoFrame (Lisboa Filho & Iochpe 2008) and MADS (Parent et al. 2006) do not approach this issue at the same level of detail as GeoOOA.

The objective of this work is to extend the UML-GeoFrame data model (Lisboa Filho & Io-
chpe 2008) in order to provide the necessary constructors for development of class diagrams involving network elements and to show how the OCL (Object Constraint Language) can be used as a formal language to the specification of integrity constraints in the network domain.

This paper is organized as follows: Section 2 presents an overview of the GeoFrame and the UML-GeoFrame data model; Section 3 describes the extension of the UML-GeoFrame to the modeling of network; Section 4 presents an example of an electric power company diagram to illustrate the extension proposed; Section 5 presents our conclusions and future work.

2 THE UML-GEOFRAME CONCEPTUAL DATA MODEL

The UML-GeoFrame data model (Lisboa Filho & Iochpe 2008) proposes a set of stereotypes, generating an UML (Unified Modeling Language) profile for conceptual geographic database modeling. The UML-GeoFrame data model constructors are well founded in a class hierarchy defined on the GeoFrame framework, originally proposed in (Lisboa Filho & Iochpe 1999). Section 2.1 presents a general description of the GeoFrame framework with only those elements necessary to the comprehension of the new constructors proposed in this paper. Section 2.2 presents the usage of stereotypes defined in the UML-GeoFrame data model.

2.1 GeoFrame Overview

The GeoFrame framework consists in a hierarchy structure of classes that describes the basic elements common to any geographic database modeling (Fig. 1).

At the planning level, the geographic regions corresponding to the interest areas (GeograficRegion class) are defined, and for each region, its associated themes (Theme class), such as hydrography, transport, relief, soil type etc. A theme can be subdivided in a hierarchy of sub-themes.

The metamodel level is composed by metaclasses that reflect how the real world elements are interpreted, being represented by conventional objects (ConventionalObject) or geographic objects (GeographicPhenomenon). This last metaclass is specialized in field (GeographicField) and object (GeographicObject) metaclasses.

The spatial representation level has the objective to reflect the way real world elements are represented by designers and users in a more abstract level regarding its presentation in a database. The SpatialObject class generalizes the spatial representation classes observed in an object view, which are Point, Line, Polygon and ComplexSpatialObj. This last one represents a spatial object composed by two or more spatial objects. The FieldRepresentation class generalizes the classes of the field view, which are GridOfCells, AdjPolygon, Isolines, GridOfPoints, TIN (Triangular Irregular Network) and IrregularPoints. Multiple representation is supported in the two views. The conceptual modeling of spatial representation of geographic phenomena noticed in the field view is the major difference of GeoFrame as compared with other models (Lisboa Filho & Iochpe 2008).

2.2 UML-GeoFrame Overview

The UML package constructor corresponds to the first concept incorporated to the UML-GeoFrame data model. It is used in the planning level to identify the themes related to a geographic region and allows a top-down approach to the problem.

The UML-GeoFrame data model defines a set of UML stereotypes. In the model, these constructors are important for the application classes do not be represented as sub-classes of the metamodel level. The goal is to avoid overloading the application diagram and making it easier to understand. Figure 2 illustrates examples of application classes using the stereotypes defined for ConventionalObject, GeographicObject and GeographicField, respectively.

The objective of this work is to extend the UML-GeoFrame data model (Lisboa Filho & Iochpe 2008) in order to provide the necessary constructors for development of class diagrams involving network elements and to show how the OCL (Object Constraint Language) can be used as a formal language to the specification of integrity constraints in the network domain.
The spatial characteristic of geographic phenomena is not abstracted in the form of spatial attributes, but by means of associations between the classes of geographic phenomena and the classes of spatial representation. This is specified by the `represent` association in GeoFrame (Fig. 1). Again, in order to avoid overloading the class diagram and making it easier to understand, stereotypes are defined to replace these associations (Fig. 3). Therefore, in an UML-GeoFrame data schema, each geographic phenomenon class of the application domain will have at least two stereotypes, one for specialization and another for spatial representation.

3 EXTENDING UML-GEOFRAME FOR NETWORK MODELING

This section describes the extension of the UML-GeoFrame data model so that it will provide specific constructors for modeling elements involved in a network. The proposal also includes new stereotypes and the relationship with the integrity constraints specification.

3.1 GeoFrame with Network Modeling Classes

The inclusion of new classes in GeoFrame is done on the metamodel and spatial representation levels, as illustrated in Figure 4.

A new specialization was added from the GeographicPhenomenon class at the metamodel level, making it possible to interpret reality elements as network elements through the NetworkObject class. At the spatial representation level, the NetworkRepresentation class generalizes the classes of network representation Node and Arc.

Although similar with the field and object views, the hierarchy used in the solution for network has two particularities. First one refers to the Network class at the metamodel level. As only alphanumeric information is added to aggregate network objects, the interpretation of a network became the same of a conventional class, but maintaining the aggregation of network objects.
This aggregation between Network and NetworkObject implies that in the application diagram an instance of Network will be an aggregation of NetworkObject instances, allowing network handling as it was a composed object. It is possible through this aggregation that an instance of NetworkObject belonged to more than one network.

The second particularity refers to the multiplicity in the association between the NetworkObject and NetworkRepresentation classes. Since this association has multiplicity one to one (1:1), an instance of the NetworkObject class should be represented by only one instance of the NetworkRepresentation class, i.e., there are no multiple representations.

A last modification was made in order to specify bidirectional and unidirectional arcs. Although data structure for nodes and arcs are different in these two types of networks, the representation, on the other hand, only differs by the arc shape. For the GeoFrame, the need to specify the type of a network and to maintain simplicity of model is based on representation, therefore the idea consisted in specializing the Arc class into the Bidirectional and Unidirectional classes.

### 3.2 New Network-based Stereotypes

With new classes added to the framework, new stereotypes were also included. They follow the same UML-GeoFrame principles, with one stereotype defined for the specialization of the geographic phenomenon and another for spatial representation. The Network class uses the conventional object stereotype. Figure 5 shows three examples of classes with the generalization stereotype for the ObjectNetwork metaclass, and representation stereotypes for the Node (a), Bidirectional arc (b) and Unidirectional arc (c) metaclasses, respectively.
3.3 Definition of Network Representation Elements

Simplicity and expressivity are characteristics of the UML-GeoFrame data model that make it easy to understand the modeled reality. On the other hand, they imply in additional textual information to build a complete diagram concerning the problem domain. UML-GeoFrame models, which provide simple constructors for developing simple diagrams, also have textual information to facilitate concept understanding.

There are some aspects that cannot be described through the use of diagrams, such as unicity, derivation and value limits of an attribute, and integrity constraints to be used on data entry or data modification. To complement and provide an accurate description of the diagrams, a formal expression language must be used.

The OCL - Object Constraint Language (OMG 2006, Warmer & Kleppe 2003) is used to describe expressions associated to the UML diagrams. Through its utilization it is possible to specify queries to define derivation rules, initial values, new attributes and operations. Besides, each expression written in OCL depends on the types defined in the UML diagrams.

Textual information incorporated to models is useful to define each of the classes or even to establish restrictions common to several domains. The Line class, for example, is defined in UML-GeoFrame as a spatial representation in object view used when the interpretation of the reality by the designer can be represented as a linear geometric form, such as, a line segment, a polyline or even a curve.

Similarly, the Node class is a spatial representation in network object view that the designer uses when the end or cross between lines is represented by a point with connectivity properties, independent of real position or transformation applied to the reality element. Unlike the graph theory, the definition of this class in GeoFrame establishes that every node should be connected to at least one line, based on the hypothesis that in the reality there are no network applications formed by nodes without an associated line.

The Arc class is a spatial representation in the network object view used by the designer to express the connectivity between one or two nodes through a line. It is also independent of the real position or transformation applied to the element of the portrayed reality.

Therefore, in any diagram, Node class instances must obligatorily connect to instances of Arc class and vice-versa. In addition, as networks are formed by at least one node and one arc, the aggregated minimum cardinality between the Network and NetworkObject classes is two. Thus, every instance of the Network class in the diagram will be an aggregation of at least two instances of the NetworkObject class.

3.4 Integrity Constraints for Graphs Data Structures

There are two basic graphs types: non-directed and directed. Hence, it is necessary two relationship types between the instances of the Node and Arc classes. Adapting the structures presented in (Laurini & Thompson 1992) to the needs of the UML-GeoFrame data model, two data structure diagrams are proposed to represent these relationships (Fig. 6).

Figure 6-a presents a diagram for non-directed graphs where each arc is associated with exactly two nodes, not necessarily different, and each node is associated with one or more arcs. The recursive association in Arc class can be used to create arc chains or arc ways. An arc can be associated to many others or be the only one in the graph. A network must be an aggregation of one or more nodes and arcs. Those can be part of more than one network.
According to the directed graph diagram presented on Figure 6-b, each arc must be associated to exactly two nodes, not necessarily different, from the beginning to end, and each node must be at least the beginning or end of at least one arc. In this kind of graph, each node must also have two extra properties, known as in and out degree, representing the number of arcs that use the node as end or beginning, respectively. The recursive association of the Arc class makes possible the creation of an arc sequence. An arc can be associated sequentially with many other arcs or be the only one on the graph. A network must be an aggregation of one or more nodes and arcs. Those can be part of more than one network.

3.4.1 Integrity Constraints for non-Directed Graphs

Though the association between the Node, Arc and Network classes can be represented by the previous diagrams, not all the necessary information is transmitted completely and accurately. For this reason, the OCL is used to complement formally the relationship specification between these classes, as shown in Figure 7. The constraint expressions (Fig. 7: i-iv) are based on the non-directed graph diagram. The constraint i describe the Network relationship with Arc and Node, where all the nodes must belong to the same network of their associated arcs. The opposite can not be assumed. Due to the existence of networks and sub-networks, an arc not necessarily belongs to every network of its associated arcs. The constraints (ii-iv) concerns the recursive relationship of the Arc class, once it is optional and no one can deduce easily when an arc must relate with another in order to create a path. The constraint ii certifies that an arc won’t be connected with itself. This rule avoids an ambiguity when an arc has a common node with both of its ends. The constraint iii imposes that every arc with node(s) in common must form a path. Nevertheless, it is important to notice that due to the aggregation with many networks, nothing can be affirmed about the connectivity between the arcs and the networks to which they belong. An arc can be connected to another one, but they do not necessarily belong to the same network.

3.4.2 Integrity Constraints for Directed Graphs

The integrity constraints (Fig. 7: v-ix) are based on the directed graph diagram. The constraint v imposes that all the nodes must be the origin or the destiny of at least one arc. The constraint vi certifies that all the nodes must belong to the same networks of their associated arcs. As the non-directed graph diagram, due to the existence of networks and sub-networks, an arc not necessarily belongs to every network of its associated nodes. The constraint vii certifies that an arc won’t connect to itself. The constraint viii imposes that the arcs that have a node(s) in common and that alternates between origin and destination must form a sequence. The constraint ix imposes that all the connected arcs must have at least one node in common so that it alternates between origin and destination.
(i) context Network
inv: self.arcs.nodes -> forAll( n | n.networks -> includes( self ))

(ii) context Arc
inv: self.linkedTo -> forAll( a | a <> self)

(iii) context Node
inv: self.arcs -> forAll( a | a.linkedTo -> includesAll(self.arcs -> excluding(a)))

(iv) context Arc
inv: self.linkedTo -> forAll( a | a <> self and a.nodes -> forAll( nl1,nl2 | nl1 <> nl2 and self.nodes -> forAll( ns1,ns2 | ns1 <> ns2 and (ns1 = nl1 or ns1 = nl2 or ns2 = nl1 or ns2 = nl2))))

(v) context Node
inv: self.in -> size() > 0 or self.out -> size() > 0

(vi) context Network
inv: self.arcs.origin.networks -> includes(self) and self.arcs.destination.networks -> includes(self)

(vii) context Arc
inv: self.next -> forAll( a | a <> self) and self.previous -> forAll( a | a <> self)

(viii) context Node
inv: self.out -> forAll( a | a.previous -> includes( self.in -> excluding(a))) and self.in -> forAll( a | a.next -> includes( self.out -> excluding(a)))

(ix) context Arc
inv: self.previous -> forAll( a | a <> self and a.destination = self.origin) and self.next -> forAll( a | a <> self and a.origin = self.destination)

Figure 7. OCL constraints for non-directed and directed graphs

4 USING THE NEW UML-GEOFRAME NETWORK CONSTRUCTORS

In order to exemplify the semantic expression power of the UML-GeoFrame data model extended to network modeling, Figure 8 illustrates the class diagram of an electric power distribution system. This example was originally presented by Kösters et al. (1997) as showed in Figure 9.

When nodes and arcs represent spatial objects, a network is called spatially embedded. Examples of this type of network include: road, electric power, water and gas networks. A counterexample is the construction of a network over adjacency relationships between neighborhoods of a city.

The concept of multiple inheritances in diagram classes was chosen to elucidate the relationship between the elements of a network and the representation of spatial objects. In order to this to happen, a class should be a specialization of the GeographicObject and NetworkObject classes, and it should associate the possible spatial representations of both views. The replacement of these specializations and associations are carried out by using generalization and representation stereotypes. It is still possible the use of multiple spatial representations for object view.

Figure 8 shows that most classes have four stereotypes. The PowerPlant class, for example, plays the role of node in the main network and at the same time has polygonal representation as geographic object. The use of four spatial stereotypes, in spite of simplifying diagram visualization, did not avoid its overloading.

To prevent this situation, it stands out as characteristic of UML-GeoFrame the choice of not representing generalization stereotypes, since stereotypes of spatial representation for field, objects and network views are disjoint groups. A class can therefore use only stereotypes of spatial representation. Superclasses also assist designers and users, as the classes sharing common properties are identified, including generalization and spatial representation.
Another characteristic about the usage of stereotypes is the representation of a network. Exemplified by the class HighVoltage, a network is composed by nodes and arcs without new geographic information incorporated to it. This way, it is not necessary to create a new stereotype to the network, the conventional class stereotype is used.

Figure 8 also shows that constructors for aggregation and composition make it possible to define to which network(s) the classes of type node and are belong. Furthermore, the relationships between arcs and nodes reduce the ambiguity during diagram interpretation. The Transformer class, for example, aggregates to the HighVoltage and LowVoltage networks, and relates to the TransmissionLine and PowerLine classes, connecting these two networks.

Formal specifications can be made through the OCL usage, and superclasses make possible the identification of common information of the diagram classes, reducing significantly the complexity of its presentation.
5 CONCLUSIONS AND FUTURE WORKS

This work presented the extension of the UML-GeoFrame data model for conceptual modeling of network elements. The proposed solution was based on a set of specific requirements for these applications identified by (Kösters et al. 1997), which include: (a) show which arcs are incident with which nodes; (b) show which networks the arcs and nodes belong to; (c) make clear which classes are arc, node and network types, without identifying this types by class names; (d) identify classes with common properties (superclass); (e) make clear which class represents the link between two or more networks.

Comparing the solution proposed in this article with the solution presented by the GeoOOA model, the UML-GeoFrame, besides complying with all these requirements, it still has as additional advantage the possibility of representing specific associations that may happen between elements of a network represented as nodes and arcs, as well as facilitating the aggregation of these elements with the composed object, which is the own network. We believe that, by this means, the conceptual scheme is read and interpreted without ambiguities by non-specialist users of the software field, and at the same time provide the designer larger freedom to accurately specify requirements of the application.

This paper also concerned the integrity constraints modeling issue involving network elements. It was presented the relationship between the new proposed classes and the way OCL can be used as a formal integrity constraint specification language in this type of application.

The need to add constructors to the UML-GeoFrame model for drawing up conceptual schemes in network applications involving non-planar graph is proposed to be included as future work.

ACKNOWLEDGMENT

This project was partially supported by CAPES, Fapemig and CNPq/MCT/CT-INFO.

6 REFERENCES