Proceedings of the 6th AGILE
April 24th-26th, 2003 – Lyon, France

OBJECT-ORIENTED MODELLING OF COMMUNICATING GIS-BASED APPLICATIONS

François Pinet (1), Ahmed Lbath (2)

(1) Laboratory of Computer Science, Modelling and System Optimization, ISIMA / Clermont Ferrand University, 63177 Aubière, FRANCE.

(2) Laboratory for Information Systems Engineering (LISI), INSA Lyon, Campus de la Doua, 69621 Villeurbanne, FRANCE.

(2) CIRIL SA, 20 rue Guérin, 69100 Villeurbanne

1. INTRODUCTION

Recent communication technologies (Internet, GSM, GPRS, PDA, GPS) have implied new GIS-based applications called telegeoprocessing applications [2]. This new type of systems concerns fields such as :

- Internet access to geographic data using Web Map Server,
- Mobile geographic databases,
- Traffic monitoring.

These new applications have naturally produced new needs in terms of conceptual methods. In order to facilitate the specification of telegeoprocessing applications, this paper proposes an extension of the Unified Modelling Language (UML) [4].

Indeed, UML began the new standard for the graphical object-oriented specifications of systems [4]. The works of [3] underline that UML extensions can be proposed in order to facilitate the GIS-based application design at a high level of abstraction. The purpose of these extensions is to encapsulate the complexity of geographic data during the first steps of modelling.

This paper aimed at describing the method T-OMEGA for telegeoprocessing design. This method integrates:

- a new extension of UML class diagrams for GIS-based application modelling (section 2),
- an extension of the Object Constraint Language (OCL) associated to UML [4] so that topologic constraints can be expressed between geographical objects (section 3),
- a new extension of UML for communication modelling ; this extension is intended for GIS designers with low skills in distributed systems field (section 4),
- an implementation of proposed UML extensions (section 5).

In introduction, the method T-OMEGA is evoked. This new method is dedicated to telegeoprocessing application design and was defined in the context of an industrial project with the French company CIRIL SA. The method T-OMEGA reuses some characteristics of the previous proposed method named OMEGA originally defined for GIS-based application design [3].

T-OMEGA considers two different types of actors participating to telegeoprocessing development project: GIS specialists and communicating system specialists. GIS specialists specify the global system to model and communicating system specialists define protocols of interaction modules used between applications. The methodology of T-OMEGA recommends that communicating system specialists use a dedicated language (such as

SDL, ROOM, SADT, etc...) in order to design communication protocols. For GIS specialists, T-OMEGA proposes a specific UML extension that will be described in this paper.

2. UML EXTENSION FOR GIS-MODELLING

The extension of UML proposed in this section focuses on the expression of geographic objects. In the T-OMEGA method, geographic entities of the real world are grouped in geographic classes. A geographic class includes an attribute called geometry and is drawn in class diagram with a stereotype <<geographic>>. The geographic type associated to objects of the class can be expressed with a tagged value {geoTypeConstraint=...}. This type can be simple (point, polyline, polygon) or combined (for example, the representation of a building object can be either point either a polygon). In order to combine geographic types, the proposed extension uses three operators (applied on geographic types by [1]):

· **AND**, each geographic class object is associated with several types of geometries; for example if a geographic class has the tagged value {geoTypeConstraint=point AND polygon}, then each object of the class is associated with a point and a polygon,

· **XOR**, each geographic class object is associated with different types of geometries; for {geoTypeConstraint=point XOR polygon}, each object is associated with a point or a polygon (but not both),

· **MULT**, each geographic class object is associated with several geometries belonging to a geographic type ; for {geoTypeConstraint=polygon MULT (0..*)}, each object is associated with from 0 to an infinite number of polygons.

An example concerning school sectors is going to underline the introduced notations. The map of each town is divided into school sectors. Each child living in a sector must be registered at the school corresponding to the sector. It exists several sector divisions depending on the study level. Sectors of the fig. 1 considers for example two types of study levels (primary school sectors and secondary school sectors).

Sectors concerning primary schools

Sectors concerning secondary schools

Fig. 1 School Sectors Exemple

The fig. 2 presents the class diagram corresponding to the school sector example. A sector is a polygon and a school can be associated to a set of polygons (i.e. a set of buildings) or to a point (depending on the available representation).

The proposed formalism offers the capability of graphically bringing geographic types to the fore. As presented in [6], stereotypes can also be directly used to express simple and combined geographic types (without the use of tagged value) and iconic representations can also be drawn in classes.

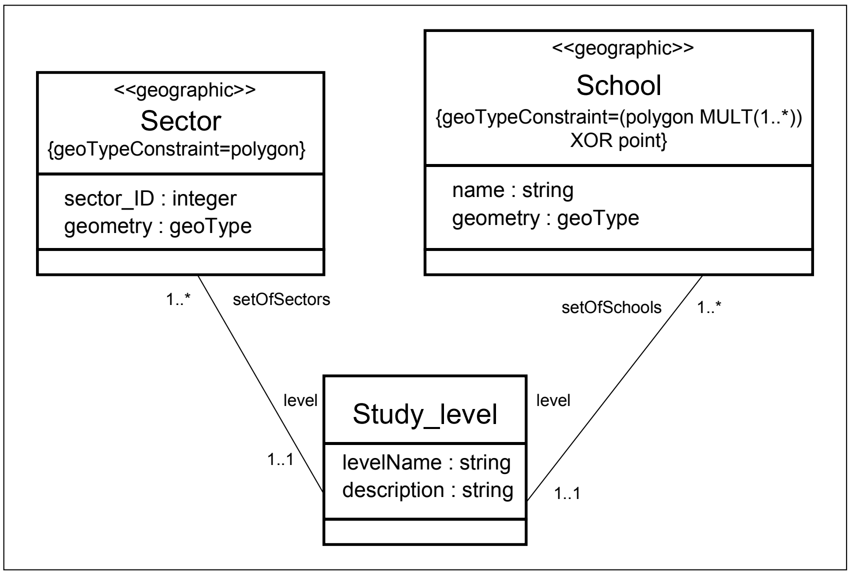

Fig. 2 Class diagram for modelling the school sector example

3. OCL EXTENSION

This section presents an OCL extension for expressing topologic and distance constraints between geographic objects. This extension consists in integrating predefined OCL operations checking topologic constraints between geographic entities having a simple geometry (point, polyline, polygon). These operations are: `isDisjoint`, `isAdjacent`, `intersects`, `isInside`, `contains`, `covers`, `isCoveredBy`, `distance`. Moreover, a new basic OCL collection type named geoType is defined. Each value of an attribute having a geoType can store a collection of simple geometries (point, polyline, polygon). New proposed OCL operations can be applied on simple geometries (point, polyline, polygon) or on OCL collection containing zero or one element.

For example, the following constraints express that sectors corresponding to the same study level are disjoint or adjacent.

```
context Sector inv:
Sector.allInstances->forAll(i|i.Study_Level=self.Study_Level and
                              i<>self implies
                              (i.geometry->isAdjacent(self.geometry)
                          or  i.geometry->isDisjoint(self.geometry))
```

In other words, let `self` and `i` be two sectors. If `self` and `i` correspond to the same study level and if `self` and `i` are different then the two sectors are disjoint or adjacent.

In the proposed OCL extension, the topological constraint can only be checked between two simple geometries. For example, let `a->isAdjacent(b)` be an OCL expression. If `a` or `b` are collections containing more than one geometry then `isAdjacent` returns systemically false else the constraint is checked.

The figure 3 shows an example of monitoring system. Diverse groups of industrial installations are displayed on a map. A group of installations is geographically represented by a set of points ; each point corresponds to the location of one installation belonging to the group. Each object of the class Installation_Group is a group of installations. Some installations can be located in a town.

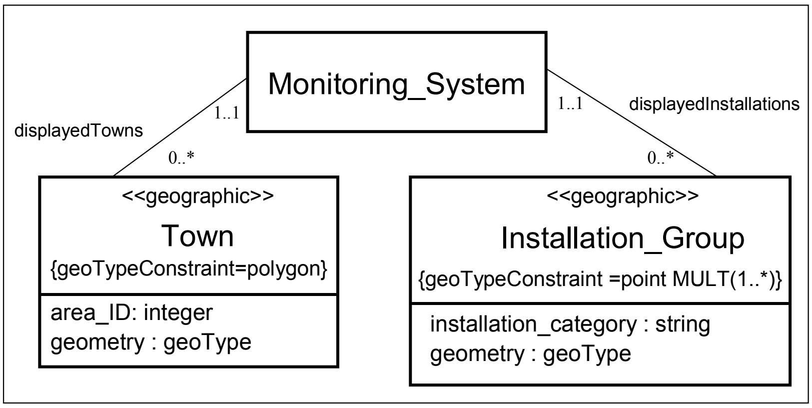

Fig. 3 A monitoring system example

The following constraints express that for each installation of a group, the distance between each installation located in a town must be greater than 3 kilometres.

```
context Installation_Group inv:
Town.allInstances
  ->forAll(i|self.geometry
    ->select(g1|g1.isInside(i.geometry))
      ->forAll(g2,g3|g2.distance(">3km",g3)) )
```

The OCL operations “select” and “forAll” can be applied on the geometry attribute because this attribute is an OCL collection.

The extension of OCL provides a powerful and practical language for expressing complex geographical constraints.

4. UML EXTENSION FOR COMMUNICATING GIS MODELLING AND IMPLEMENTATION

The purpose of this extension is to give the capability for GIS designers (non specialists in communicating systems) to model communications between applications. In order to offer understandable abstraction, T-OMEGA proposes the following concepts:

- **application classes**; this new abstraction provides the capability to group class diagrams by application.
- **connection associations**; An instance of a connection association exists between two application objects if and only if these two objects are connected in a network (by Internet for example), in order to communicate.
- **imported classes**; Imported classes define which class instances belonging to a first application are accessible by a second application.

The fig. 4 underlines these new concepts in describing a client-server architecture. In this system, mobile clients request routes to servers. A server can communicate with another server in order to access additional data. Thus, two application classes are drawn in the diagram: a server class and a client class. Class diagrams associated with each application are presented inside application classes.

The server class and the client class are linked by a connection association named 'COMGSMDATA' in order to model that clients and servers can communicate in using the GSM-DATA protocol. The relation multiplicity shows that a client can be connected to 0 or 1 server and a server can be connected to 0..50 clients. In the same way, the connection association 'COMInternet' shows that a server can communicate with another in using Internet.

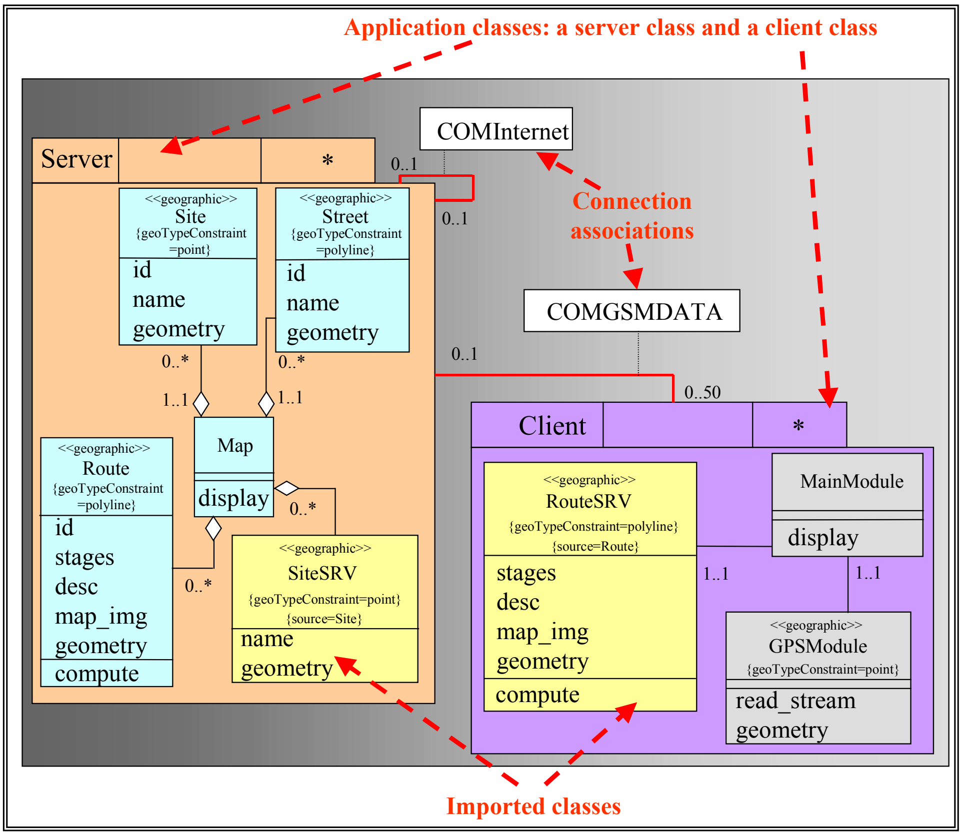

Fig. 4 A client-server example

Servers display diverse geographic data (site, street, route) and client applications integrate Global Positioning System (GPS) modules. An imported class is produced by a duplication of a second class ; this second class is called source class. An imported class *C1* (defined in an application *A1*) indicates that instances of a source class *C2* (defined in an application *A2*) are accessible by *A1*. A projection operation can also be applied on the list of attributes and operations in order to hide some features. In the example of the fig. 4, the imported class named *RouteSRV* shows that the instances of the source class *Route* are accessible by the application *Client*. Only the technical attribute *id* is not accessible by the client application. In the same way, the class *SiteSRV* indicates that an instance of the class site stored an application A1 can be acceded by a remote application A2.

In general, it is possible that a source class is a geographic class but not the corresponding imported class (because the attribute geometry is not declared in the imported class). Moreover, in order to model data conversion, the tagged value geoTypeConstraint of a source class and the tagged value of the corresponding imported class can be different.

The T-OMEGA formalism has been implemented in a tool named T-AIGLE. The next section focuses on a short introduction of T-AIGLE.

5. T-AIGLE: A TOOL FOR GENERATION OF TELEGEOPROCESSING APPLICATIONS

The method T-OMEGA is supported by the tool T-AIGLE which is an extension of the CASE Tool AIGLE dedicated to GIS-application generation. The new tool T-AIGLE has been especially developed for telegeoprocessing application design. It is composed of four parts:

- **Diagram editor**; in order to draw T-OMEGA diagrams and other UML graphical components.
- **Code generator**; the generator can automatically produce code and documentation from T-OMEGA specifications. The target language of the generator is Java.
- **Code import module**; this module allows the import of existing Java code into T-OMEGA diagrams.
- **Repository**.

The generator module provides Java code using different communication protocols: Java DataBase Connectivity, Remote Method Invocation, Servlet. T-AIGLE is an extension of the UML model of Microsoft Visio 2000. Specific components have been coded and integrated into Visio 2000, in order to support T-OMEGA.

Thanks to T-AIGLE, several telegeoprocessing applications have been produced in diverse fields (school sector management, street cleaning monitoring, urban network,…). These applications are based on an Intranet/Internet portable architecture [5].

6. CONCLUSION AND PERSPECTIVES

This paper presents a new method for telegeoprocessing applications (communicating GIS-based applications). Different abstractions have been defined, in an extension of UML, so that GIS designers can easily model all aspects of telegeoprocessing systems. The tool T-AIGLE has been implemented in order to support the method T-OMEGA.

We underline different perspectives:

- definition of analysis to check the consistency of diagrams,
- study of the formalism visual complexity,
- automatic generation for other target programming languages,
- development of a mobile architecture with the tool T-AIGLE.

7. BIBLIOGRAPHICAL REFERENCES

[1] Brodeur J., Bédard Y., Proulx M. J. “Modelling geospatial application databases using UML-based repositories aligned with international standards in geomatics”, *Proceedings of the 8th ACM Symposium on Advances in Geographic Information Systems*, Washington D.C., USA, November 10-11, 2000, p. 39-46.

[2] Laurini R., “A Short Introduction to TeleGeoProcessing and TeleGeoMonitoring”, *Proceedings of the 2nd Symposium on TeleGeoProcessing*. Nice, France, May 10-12, 2000, p.1-12.

[3] Lbath A., “From Visual Customization of GIS Software to Visual Design and Automatic Generation of "TeleGeoProcessing Applications”, *Proceedings of the 1st Workshop on TeleGeoProcessing*, Lyon, France, May 6-7, 1999, p.200-204.

[4] OMG, *Unified Modelling Language Version 1.4*, OMG Technical Report, 2001.

[5] Pinet F., Lbath A. “An Extension of the CASE Tool AIGLE for Web-based GIS Applications”, *Proceedings of the 2nd Symposium on TeleGeoProcessing*. Nice, France, May 10-12, 2000, p.41-51.

[6] Pinet F., Lbath A. “Semantics of Stereotypes for Type Specification in UML: Theory and Practice”, *Proceedings of the 20th Conference on Conceptual Modelling (ER’01),* Yokohama, Japan, November 27-30, 2001, p. 339-353.