Domain & Model Driven Geographic Database Design

Jugurta Lisboa-Filho, Filipe Ribeiro Nalon, Douglas Alves Peixoto, Gustavo Breder Sampaio and Karla Albuquerque de Vasconcelos Borges

Abstract After many years of research in the field of conceptual modeling of geographic databases for Geographic Information Systems, experts have produced many different alternatives of conceptual data models from extensions of the Entity-Relationship model or of Unified Modeling Language (UML). However, the lack of consensus on which is the most suitable one for modeling applications in the geographical domain, brings up a number of problems for field advancement, mainly problems of interoperability of database design and CASE tools. The Model Driven Architecture (MDA) approach allows the development of systems from an abstract view until the corresponding implementation code that can be automatized by means of models transformation. A UML Profile is an extension mechanism of UML which allows a structured and precise extension of its constructors, being a good solution to standardize domain-specific modeling, as it uses the entire UML infrastructure. This chapter describes the use of MDA approach in the design of databases in geographical domain; using a UML Profile called GeoProfile aligned with international standards of ISO 191xx series. The chapter also shows that with the automatic transformation of models it is possible to achieve the generation of scripts for spatial databases from a conceptual data schema in a high level of abstraction.

Jugurta Lisboa-Filho
Universidade Federal de Viçosa, Departamento de Informática, e-mail: jugurta@ufv.br

Filipe Ribeiro Nalon
Prodabel - Empresa de Informática e Informação do Município de Belo Horizonte e-mail: fnalon@gmail.com

Douglas Alves Peixoto
Universidade Federal de Viçosa, Departamento de Informática e-mail: douglasalves.ufv@gmail.com

Gustavo Breder Sampaio
Universidade Federal de Viçosa, Departamento de Informática e-mail: gustavobreder@gmail.com

Karla Albuquerque de Vasconcelos Borges
Prodabel - Empresa de Informática e Informação do Município de Belo Horizonte e-mail: karla@pbh.gov.br
1 Introduction

The activity of software development is a task that requires increasing use of standardized methodologies and techniques that are widely known. Currently, the main concern of the designer is a good understanding of the problem domain in order to generate solutions that suit the real necessities of the users.

In order to assist in this task of understanding the problem and reducing the system complexity to be developed, the main technique that is used is modeling. A model is a reality simplification [9]. In database design, the construction of models in steps helps to design the database structure without having to worry about implementation details.

In the last 20 years, research has aimed to create or adapt conceptual data models for geographic applications. The existence of several models has brought a problem to the area, which is the lack of a modeling standard. Tools have been created for different models and it is difficult to obtain interoperability among the created solutions.

For the standardization of these models, a UML profile called GeoProfile [21] was proposed. A profile is an extension mechanism of the Unified Modeling Language (UML), which allows customizing the UML to a specific domain. The GeoProfile was proposed for conceptual data modeling in the geographic domain, which puts together the characteristics of the main existing spatial data conceptual models. The construction process of the GeoProfile can be compared with a step of the domain engineering, which is the domain analysis, in which the domain knowledge is studied and analyzed. According to Falbo et al [10], one of the domain analysis goals is to make possible the reuse of the domain model generated for a group of applications. One of the UML profile construction outcomes is a domain metamodel. This metamodel contains reusable requirements, of the intended domain, to build applications in that domain.

Furthermore, as an effort for the geographic information standardization, some organizations, such as the International Organization for Standardization (ISO) and the Open Geospatial Consortium (OGC), have published international standards to help in the construction of standardized geographic applications.

The objective of this chapter is to describe the use of the Model Driven Architecture (MDA) approach in the design of databases in geographical domain, using the GeoProfile aligned with international standards of ISO 191xx series. The chapter also shows that with the automatic transformation of models it is possible to achieve the generation of scripts for spatial databases from a conceptual data schema in a high level of abstraction.

Section 2 presents the main concepts related to modeling of geographic databases, describes the main international standards for geographic information and summa-
rizes the GeoProfile. Section 3 describes the steps and types of models used in the MDA approach. Section 4 describes the process of designing geographic databases based on the MDA approach. Section 5 illustrates the entire process of designing a geographic database, based on a case study of a system in the field of sugarcane crop for ethanol production. Some conclusions are presented in Section 6.

2 Geographic Database Modeling

2.1 Basic Requirements

One of the main components of a Geographic Information System (GIS) is the storage component denominated geographic database, whose function is to structure and store the data in order to allow analysis operations involving spatial and alphanumeric data [36].

Due to the complexity of GIS applications, a major challenge in developing these systems has been designing the database, since this type of project requires the use of different tools, as the activities required for their preparation vary according to the complexity of the system, the type of personnel involved, the database management system (DBMS) used, etc.

The database design is traditionally done in three stages: conceptual, logical and physical [8]. According to Borges et al. [4], for applications in the geographical domain, the level of conceptual representation provides a set of concepts with which the geographic phenomena, such as rivers, buildings, roads and vegetation, can be modeled at a high level of abstraction, as perceived by the user. Classes to be created in the database are defined at this level, which are possibly associated with some kind of spatial representation.

Parent et al. [27] highlight some advantages of using conceptual modeling in applications that manipulate geospatial data. First, users can express their knowledge of the system using concepts that are close to their reality and independent of computing concepts. Moreover, as conceptual modeling is independent of system implementation, the result of modeling remains valid in case of technological changes. That is, the developed scheme can be reused regardless of the GIS chosen for the system implementation. Finally, due to its readability, conceptual modeling promotes the exchange of semantic information referring to the project.

Because of the particularities of the geographic information, several specific solutions for the modeling of geographic data have emerged in recent years. Lisboa-Filho and Iochpe [19] proposed a list of key requirements for modeling geographic data as follows:

- Geographic phenomena and conventional objects: In a geographic database, in addition to the data on geospatial phenomena, there are generally conventional data, such as those contained in any information system. A farm, for example, can be a geographical phenomenon if its spatial information is stored in the database,
such as its boundaries. On the other hand, the information about the owner of the farm can be an example of a conventional object if it does not present spatial features;

- **Field and object views**: The classification of geographic phenomena in these two views is intended to represent properly the geographical reality observed. While in the object view the real world consists of entities and individual well-defined spatial boundaries (e.g., rivers, plots, and streets), in the field view, the real world is understood as a set of attributes that vary continuously in space (e.g., relief, soil type, temperature);

- **Spatial Aspects**: This requirement refers to the need of connecting fields and geographic objects to an abstract spatial form for their representation. In the object view the phenomena are represented by points, lines, polygons, or their combinations. In the field view, a continuous surface can be represented by numerical models, sets of isolines or grid of cells, for example. The type of representation to be used depends on the purpose of the application, the representation scale and the shape of the phenomenon;

- **Thematic aspects**: In a GIS, geographic entities are not treated in isolation. They are grouped according to the characteristics and relationships they have in common. The division of a system by themes (e.g., hydrography, vegetation, urban planning) allows modeling simplification and facilitates understanding of the area by the designer;

- **Multiple representations**: Users can have different views of the same phenomenon, which are probably represented in different scales or projections. For example, a city can be represented as a point or a polygon depending of the data scale;

- **Spatial relationships**: The identification of the types of relationships that must be kept in the database is a complex problem, since the number of possible relationships is large in the geographical area, because of the spatial interactions that may occur among the phenomena. For example, a road can cross a river, but this relationship can be or not be held in the database;

- **Temporal aspects**: The storage of the changes that occur in geographic features is important for a better understanding of the phenomena and making predictions. For example, the limits of a parcel can change over the years.

Other similar classifications were also proposed, for example, Friis-Christensen et al. [11] divide the requirements for modeling geographic data into five groups as follows: spatial-temporal properties, roles, associations, constraints and data quality. In a deeper analysis, it is possible to confirm that both classifications are equivalent in many aspects.

Pinet [28] lists a series of conceptual models of specific data for the geographical domain. Among them we can highlight some models based on objects, such as GeoOOA [17], OMT-G [4], MADS [27], UML-GeoFrame [20] and the PVL model of the Perceptory tool [1]. Each model has particular characteristics and seeks to meet the requirements for geographic application modeling. Based on these models and according with International Standards for Geographic Information shown in the next section, the UML GeoProfile was proposed as described in the Section 2.3.
2.2 International Standards for Geographic Information

Standards are used in many fields of society and some of them are better known as, for example, the series of standards ISO 9000, which is quite common in organizations, because it defines a set of rules and guidelines for quality management in an organization. With respect to geographic information, some efforts to standardize work have been undertaken by organizations like the International Organization for Standardization (ISO) and the Open Geospatial Consortium (OGC).

These organizations are examples of the two types of groups that exist for international standardization, which are the international organizations and international consortia. International organizations base their decisions on consensus and are independent of the interests of individual industries or governments. Their standards are known as \textit{de jure} standards. An example of international organization is the ISO. On the other hand, the international consortia are made up primarily of members from industry, government agencies and universities. The standards developed by consortia are called industry standards or \textit{de facto} standards. The OGC can be considered the most important consortium in the geographic information community [18]. According to Brodeur and Badard [5], the development of standards for geographic information aims to reduce the inconsistency between \textit{de jure} and \textit{de facto} standards.

2.2.1 The ISO 191xx series of the ISO/TC 211 Technical Committee

The Technical Committee ISO/TC 211 is the one responsible for the preparation of the ISO 191xx series, which defines the international standards regarding the geographic information field. These standards aim to promote the usage of geographic information in an efficient, effective and economical way, thus contributing to the solution of global problems, such as the humanitarian and ecological problems [35].

The ISO 191xx series standards are divided into specific groups. As listed in the ISO/TC 211 [35], there is the group of standards that specify the infrastructure for the geospatial standardization, standards that describes data models for geographic information, standards for geographic information management, standards for geographic information services, standards for encoding of geographic information and standards for specific thematic areas. These standards can contribute in several levels of abstraction, since modeling up to the consideration of implementation aspects. In this chapter some standards related to data models for geographic information, more specifically the ISO 19107 Spatial Schema [33], ISO 19108 Temporal Schema [32] and ISO 19123 Schema for Coverage Geometry and Functions [34] standards, are analyzed.

The ISO 19107 Spatial Schema standard specifies a schema to describe and manipulate the spatial characteristics of the geographic features. A feature is an abstraction of a real world phenomenon. This abstraction is a geographic feature if it is associated to a relative localization on the Earth [33]. The standard consists of classes’ diagrams that can be used in application schema, profiles and implementa-
tion specifications. It also defines spatial operations, standards for use in the access, query, management, processing, and data exchange of geographic objects. The ISO 19107 standard defines in details the geometric and topological characteristics that are necessary to describe the geographic features.

The ISO 19108 Temporal Schema standard defines the concepts regarding the temporal characteristics of geographic information, showing how these characteristics are abstracted from the real world. Jensen [14] considers two types of time: the valid time and the transaction time. The first one is the time when a fact is true in the observed reality and it is generated by the user. The second one is the time when a fact is stored in a database and it can be recovered. This international standard emphasizes the valid time instead of the transaction time. The standard consists of a class hierarchy that considers the geometric and topological aspects of the temporal characteristics [32].

The ISO 19123 Schema for Coverage Geometry and Function standard, on the other hand, defines a schema for the spatial characteristics of coverage. Coverage is a feature that has multiple values for each type of attribute and can represent a simple feature or a set of features. They integrate discrete and continuous geographic phenomena [34]. Examples of coverage include raster, TIN, point coverage and polygon coverage. They are used in several specific areas such as, for instance, remote sensing, meteorology, soils and vegetation.

Some related works have analyzed conceptual data models and their integration with geographical standards. Belussi et al. [3] describe the conceptual data model GEOUML in which a geographic database schema can be designed from the specialization of ISO TC211 standards. However, this model does not use graphic symbols for representing phenomena’s spatial representation, which is a feature presented in various models proposed in the literature [1]. A study where the model elements of the Perceptory tool are related to the ISO standards is presented in [6].

2.2.2 Open Geospatial Consortium Standards

The Open Geospatial Consortium (OGC) is currently the main consortium responsible for developing industry standards for Geographic Information Systems. Its development process is different from the ISO approach. The OGC develops specifications mainly focused on implementation, while ISO develops more abstract specifications.

Despite these minor differences, both ISO and OGC have developed cooperative agreements to harmonize their work and to develop future work [6]. For example, the document OpenGIS - Simple Feature Access is also recognized by the ISO under the name ISO 19125. The document is divided into two parts. The first (Common Architecture) describes a common architecture for simple geographical features and the second part (SQL option) describes an SQL implementation of the model described in the first part. Both parts deal with simple features, namely features whose geometry is restricted to two dimensions. The OGC standard used is the Simple
Feature Access, as well as GeoProfile and projects using the MDA approach, both described in this chapter; this will be handled as its corresponding ISO 19125.

### 2.3 GeoProfile - a UML Profile for GIS Databases

The Unified Modeling Language (UML) is a visual modeling language used for documenting artifacts of software systems. Despite of being a general purpose language that can be used in various application domains, there are situations in which the elements of the UML are not able to effectively express some concepts of particular domains. Thus, the language provides extension mechanisms that allow customizing it to suit a specific application domain as follows:

- **Stereotypes**: A stereotype defines how an existing metaclass may be extended and enables the use of specific terminology for a domain or different platform in place of or in addition to the terminology used for the extended metaclass. Stereotypes can also change the appearance of the elements of the extended model using graphic icons;

- **Tagged values**: They are additional meta-attributes associated with a metaclass of the metamodel extended by a profile and add information to elements of the model;

- **Constraints**: These are restrictions associated with the corresponding elements of the metamodel. They can be written using natural languages or in Object Constraint Language (OCL), which is also standardized by the Object Management Group (OMG);

- **Profile**: A UML profile is a set of extension mechanisms grouped in an UML package stereotyped as profile.

A well-specified UML profile will have direct support of Computer Aided Software Engineering (CASE) tools. In other words, once the profile is defined, there is no need to implement new CASE tools. Enterprise Architect [31] and Rational Software Modeler [13] are examples of CASE tools with support for UML profiles.

The UML profile denominated GeoProfile [21, 23] was proposed to integrate the features of the major geographic data conceptual models. Thus, the GeoProfile is not a new model, but rather a compilation and integration from the builders of specific GIS applications present in the main models in the literature. With this, it is possible to use all the elements and advantages of UML 2.0. In addition, a designer familiarized with a particular model can customize GeoProfile, making it look like that model. The GeoProfile comprises a metamodel and a set of stereotypes, described in the following subsections.
2.3.1 The GeoProfile Metamodel

The GeoProfile metamodel, as defined in [21], is showed in Fig. 1. A geographic database comprises a number of themes, each of presented as a Theme metaclass. A theme can be formed by the aggregation of other themes or objects with or without spatial representation, characterized by the classes GeoPhenomenon and ConventionalObj, respectively.

When one chooses to associate a spatial representation with objects of a class, it is possible that the phenomenon is perceived in the geographic field view (GeoField) or object view (GeoObject). Depending on the technique used in geographic information acquisition in the field, its representation is selected from six options as described in [12]: AdjPolygons, Isolines, TIN, GridOfPoints, GridOfCells or IrregularPoints. Representation of geographic objects can be of the types Point, Line, Polygon or ComplexSpatialObj (the object geometry consists of other geometries). To specify multiple representations, it is possible to use more than one stereotype in the same class of the conceptual schema.

The metaclass Network is used to modeling a whole network structure and contains only alphanumeric attributes which describes the general features of the network. Since this metaclass does not have spatial information, it was defined as a specialization of ConventionalObj. The networks are formed by NetObject objects, which can be nodes (Node), unidirectional arcs (Unidirectional) or bidirectional arcs (Bidirectional).

GeoProfile also indicates whether a class is considered temporary or not. In this case, it is implied that both the attributes and spatial data of an object can vary, and these changes must be maintained in the database. In this way, the metaclass TemporalObject was added to the metamodel. This metaclass has two attributes that characterize temporal information. One of these attributes indicates the temporal type (validity time, transaction time or bitemporal time), whereas the other defines the used temporal primitive type (instant or interval). There are two enumerations (TemporalType and TemporalPrimitive) for the possible values these attributes can assume.

2.3.2 GeoProfile's stereotypes

After creating the domain metamodel, the next step is to extend the UML metaclasses to create the profile itself. Fig. 2 illustrates the stereotypes that have been defined for GeoProfile, which extend the metaclasses Class and Association of the UML. The black arrows refer to an extension relation between a stereotype and an UML element. This is the mechanism proposed by OMG to extend the UML in the following way: a basic UML element can be used to represent an element of a generic domain; in this case the UML elements Class and Relationship are used to represent the GeoProfile elements.

The white arrows refer to a specialization relation between UML stereotypes. This kind of relation is called Is-A, where a stereotype subclass inherits all the
features and information from another stereotype superclass. New information and features can be added to the stereotype subclass, differentiating it from the others. The stereotypes Node and Arc, for example, are specializations of the stereotype NetObject. Both stereotypes, Node and Arc, are subclasses of NetObject and have common features inherited from NetObject, but each one of them can hold its own distinct features and information.

The GeoObject and GeoField stereotypes represent the geographic phenomena perceived in the objects and fields views, respectively. Since these stereotypes were defined as abstracts, as well as the NetworkObj and Arc stereotypes, they will not be included in the data schema during the modeling using the GeoProfile, but their corresponding subclasses will.

To deal with temporal aspects, the TemporalObject stereotype, that also extends the metaclass Class, was included. The two enumerations that were included (TemporalPrimitive and TemporalType) are used to list the possible values that the meta-attributes (tagged values) temporalPrimitive and temporalType may assume, which are: instant and interval.

Besides the extensions to the metaclass Class, extensions to the metaclass Association were included. These extensions are aimed to creating stereotypes to serve the topological relationships [7], which are: Touch, In, Cross, Overlap and Disjoint. In addition, designers are allowed to indicate that an association between two objects is only valid for one period and this history should be kept in the database. This is done by simply assigning the stereotype Temporal.

An important requirement for the project of UML profiles described in [26] is the definition of a graphical notation for the stereotypes. In the modeling of geographic databases, the use of this feature to represent the spatial characteristics of geographic objects is used in many models, for example, pictograms, initially developed by Bédard and Paquette [2], which influenced many models that emerged.
later. They help improve clarity and make the modeling more intuitive for the designer and easily understood by users. Fig. 3 shows a set of icons that can be added to GeoProfile stereotypes. These icons were also based on the models mentioned above (UML-GeoFrame, OMT-G, MADS, and GeoOOA Perceptory’s model), but designers accustomed to using a particular model can customize these icons as they wish.

Figures 4 and 5 illustrate some examples of classes modeled with GeoProfile stereotypes in graphic and textual forms. Fig. 4 illustrates an example of spatial relationship between two classes (District and AdmRegion) with polygon spatial representations, specified by the stereotype Polygon. The stereotype Overlap shows the spatial relationship that occurs between the classes. Fig. 5 illustrates three examples of classes with spatial representation in the field view.
The SatImage class with the stereotype GridOfCells, the Humidity class with the stereotype IrregularPoints and the Relief class with the stereotype GridOfPoints.

Besides the stereotypes, some constraints were also added, which are useful for the conceptual schema validation. Those constraints basically prevent the occurrence of three error types: addition of incompatible stereotypes with a same element, poor network construction and addition of impossible topological relationships between two elements (e.g. Cross relationship between two geographic objects with point representation). These three constraints groups were analyzed and a set of OCL expressions was specified. OCL has been frequently used to specify additional integrity constraints to the UML diagrams [29].

The code below shows one of the GeProfile’s OCL constraints, which is applied to the stereotype GeoField. This constraint defines that each class stereotyped as a geofield (context GeoField) must capture all stereotypes applied to this class (getAppliedStereotypes). If the output is stereotyped as a geographic object (Point, Line, Polygon, or ComplexSpatialObj) through the method select, the result set must be empty (isEmpty). This constraint checks for incompatible stereotypes in a class which has been assigned the stereotype GeoField in the schema. More details regarding all OCL constraints and how to implement them in a UML profile can be accessed in the GeoProfile’s Web page at www.dpi.ufv.br/projetos/geoprofile.

context GeoField
inv: self.getAppliedStereotypes() ->
Finally, although the GeoProfile is most commonly used to static data schema designs, the class behavioral modeling, that is, the specification of the applicable operations to instances of a class, can be done naturally within this class, using methods specified in the UML.

3 Model-Driven Architecture

To improve software development OMG has adopted the MDA approach, which emphasizes the use of models. In this approach, the software development process is directed by the modeling activity of the system. A system model is a description using a specific notation. The artifacts produced in MDA are formal models, that is, models that can be understood by computers [25].

In MDA, the system requirements are modeled using a Computation Independent Model (CIM). This model is called domain model or business model and it uses a familiar vocabulary to the domain experts. A CIM does not show details of the systems structure, but of the environment in which the system will operate. This kind of model provides a useful way to understand the problem itself [9, 25].

In the second level of abstraction we find the Platform Independent Model (PIM). This is a model with an abstraction level relatively high and independent from any implementation technology [9, 16, 25].

Later, the PIM is transformed into a Platform Specific Model (PSM). A PSM is customized in order to specify the system in terms of implementation constructors which are available in a specific implementation technology. For instance, a PSM relational database include terms such as table, column, foreign key, among others. A PIM can be transformed into one or more PSMs. For each specific technology platform, a separate PSM is generated. The following step is the transformation of each PSM to source code. This transformation is relatively direct since the PSM is adjusted to the selected technology. Fig. 6 illustrates the different levels of abstraction of MDA approach, showing the CIM as the highest level of abstraction model and the others, PIM and PSM, as inferior levels.

The CIM, PIM and PSM are shown as artifacts in different steps in the system development life cycle, and they represent different abstraction levels in its specification as well. The ability of transforming a high level CIM into a PIM and later, transforming a PIM into a PSM increases the abstraction level in which a designer can work. This allows a designer to face more complex systems with fewer struggles [9, 25].

The development process using MDA approach may be compared with the process of domain engineering, which highlights three main steps: domain analysis,
domain design, and domain implementation [10]. At the domain analysis step, the domain requirements are defined. These requirements are expected to be reusable, such as the CIM in the MDA approach, which is used to understand the problem itself. In the domain design step a generic and independent of platform architecture is established, such as the PIM level. Finally, at the domain implementation step the identification of reusable assets is done, as well as the architecture and components implementation, such as in the PSM level of the MDA, which transforms the considered PIM into a specific platform.

One of the aims of the MDA approach is to reduce the system development time. For this purpose, models in different abstraction levels are used, starting with models in high abstraction levels. Therefore, one of the challenges is transform high level models into lower level models. The transformation of models is the process of converting a model into another model that represent the same system [25].

An important characteristic of the MDA is that the transformations are automatically executed. Traditionally, the transformations from model to model or from model to code are manual. In the MDA approach, on the other hand, transformations are executed preferably by tools [16].

An automatic mapping is specified using a language to describe the transformation of a model into another. A desirable quality of a transformation language is portability; this enables the use of a mapping with different tools [25].

Some tools, available in the market, for supporting the MDA approach, have mechanisms for transforming predefined templates, but the ideal is to offer support to a language that enables users to customize the transformation of models as needed. An example of such language is the Atlas Transformation Language (ATL) developed by the research group ATLAS INRIA & LINA [24].

This language allows the definition of transformation rules, in which, given a schema created in a model of entry, along with transformation rules, generates a new schema in the output model, according to those rules. Thus, this language allows the transformation of a schema made from GeoProfile, for example, into another
specific conceptual model, allowing the exchange of information between models and giving the designer flexibility in creating a schema. This approach can also be used in the transformation of a schema in each of the three levels of MDA; in this case, using the ATL language, it is necessary to define the transformation rules for each level.

4 Modeling Geographic Databases using MDA

The use of MDA is not specific to the geographical domain, but it was used in this work to exemplify a domain engineering on the geographical field of study.

The development of GeoProfile was mainly motivated by the fact that UML can be used, along with all its available resources, for example, CASE tools, to model a geographic database.

The development of GeoProfile based on international standards is in accordance with the abstraction levels of the MDA approach. A major benefit of this approach is the productivity gain through the emphasis on modeling and the transformation of high-level models to lower-level models in an automated way [16]. Thus, the design of geographic databases can also take advantage of these benefits. For example, using tools that support the transformations will make it possible to generate, from the GeoProfile, lower-level models and, later, the database scripts for specific technologies, such as Oracle Spatial and PostGIS.

In related studies, Miralles and Libourel [22] propose a framework to design and implement spatial-temporal databases following the MDA approach, but this framework does not consider the CIM level, but suggests the use of the Perceptory tool to specify the business model. Bérard and Larrivée [1] also mention the MDA approach as a key application for the use of the Perceptory model and consider the three levels of abstraction, namely, CIM, PIM and PSM.

4.1 CIM level

At this level of abstraction, only aspects related to the problem’s domain are addressed, without dealing with implementation details. For the conceptual model of the database, the GeoProfile is used at this level, because it is designed to help designers in the first steps of a database project. The concern is to represent which are the spatial features of a particular geographic element and not how these features will be implemented. The use of stereotypes helps in this direction, since they make the model more intuitive for the user to understand the spatial features that are being represented.

Fig. 7 illustrates an example of a schema modeled with GeoProfile at this level of abstraction. The schema shows four classes, three of them with spatial features and thus are stereotyped considering the notation proposed in Fig. 3.
4.2 PIM level

After constructing the initial model of the database using GeoProfile, this model is transformed into a PIM model. At this level of abstraction, the elements of international standards are taken into account. To make the transformation, the GeoProfile stereotypes were mapped to specific classes of international standards. Table 1 illustrates the mapping carried out with the standards ISO 19107, ISO 19108 and ISO 19123. However, similar mapping can also be made for OGC standards. Due to lack of space, these will not be seen in this chapter.

Fig. 8 shows the PIM model resulting from performing transformation on the model shown in Fig. 7. The spatial features were transformed into attributes whose types are in accordance with the elements of ISO 191xx standards shown in Table 1. For example, the City class, which was modeled with the stereotype <<Polygon>>, takes on a geometry attribute, denominated geometry, of the type GM_Surface. The same was done with the other classes that have spatial features.

4.3 PSM level

The next step is to transform the PIM model into a PSM model, which can be, for example, an object-relational data model extended to manage spatial objects (e.g., Spatial or PostGIS). To illustrate this transformation, Fig. 9 shows an example of the PSM model that corresponds to the platform Oracle Spatial, which was generated from the PIM model shown in Fig. 8. This model already takes into account details of the platform in question, for example, the data types of the platform. Some attributes were also marked with the stereotype <<PK>> and <<FK>>, which represent the primary and foreign keys, respectively. The purpose of this step is to make the model as close as possible of the chosen platform to automate the generation of the script database.
Table 1 Correspondence between the GeoProfile elements and the ISO 191xx standards.

<table>
<thead>
<tr>
<th>Requirements of GeoDB modeling</th>
<th>GeoProfile</th>
<th>Classes in the ISO standards</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Geographical objects</strong> in the object view</td>
<td><strong>Point</strong></td>
<td><strong>GM_Point</strong></td>
<td>ISO 19107</td>
</tr>
<tr>
<td></td>
<td><strong>Line</strong></td>
<td><strong>GM_Curve</strong></td>
<td>ISO 19107</td>
</tr>
<tr>
<td></td>
<td><strong>Polygon</strong></td>
<td><strong>GM_Surface</strong></td>
<td>ISO 19107</td>
</tr>
<tr>
<td></td>
<td><strong>ComplexSpatialObj</strong></td>
<td><strong>GM_Complex</strong></td>
<td>ISO 19107</td>
</tr>
<tr>
<td><strong>Geographical objects</strong> in the field view</td>
<td><strong>TIN</strong></td>
<td><strong>CV_TINCoverage</strong></td>
<td>ISO 19123</td>
</tr>
<tr>
<td></td>
<td><strong>Isolines</strong></td>
<td><strong>CV_SegmentedCurveCoverage</strong></td>
<td>ISO 19123</td>
</tr>
<tr>
<td></td>
<td><strong>AdjPolygons</strong></td>
<td><strong>CV_DiscreteSurfaceCoverage</strong></td>
<td>ISO 19123</td>
</tr>
<tr>
<td></td>
<td><strong>GridOfPoints</strong></td>
<td><strong>CV_DiscreteGridPointCoverage</strong></td>
<td>ISO 19123</td>
</tr>
<tr>
<td></td>
<td><strong>GridOfCells</strong></td>
<td><strong>CV_GridCell</strong></td>
<td>ISO 19123</td>
</tr>
<tr>
<td></td>
<td><strong>IrregularPoints</strong></td>
<td><strong>CV_DiscretePointCoverage</strong></td>
<td>ISO 19123</td>
</tr>
<tr>
<td><strong>Network elements</strong></td>
<td><strong>Node</strong></td>
<td><strong>TP_Node</strong></td>
<td>ISO 19107</td>
</tr>
<tr>
<td></td>
<td><strong>Arc</strong></td>
<td><strong>TP_Edge</strong></td>
<td>ISO 19107</td>
</tr>
<tr>
<td></td>
<td><strong>UnidirectionalArc</strong></td>
<td><strong>TP_DirectedEdge</strong></td>
<td>ISO 19107</td>
</tr>
<tr>
<td></td>
<td><strong>BidirectionalArc</strong></td>
<td><strong>TP_DirectedEdge</strong></td>
<td>ISO 19107</td>
</tr>
<tr>
<td><strong>Temporal objects</strong></td>
<td><strong>TemporalObject</strong></td>
<td><strong>TM_Object</strong></td>
<td>ISO 19108</td>
</tr>
<tr>
<td></td>
<td><strong>Instant</strong></td>
<td><strong>TM_Instant</strong></td>
<td>ISO 19108</td>
</tr>
<tr>
<td></td>
<td><strong>Interval</strong></td>
<td><strong>TM_Period</strong></td>
<td>ISO 19108</td>
</tr>
</tbody>
</table>

Fig. 8 An example of a model at the PIM level of abstraction

Listing 1 in the Appendix shows a small part of the transformation code from the CIM model, shown in Fig. 7, to the PIM model presented in Fig. 8, using the ATL models transformation language. The definition of transformations in ATL starts with the transformation module statement as well as the source and target models. The module is defined using the keyword `module` followed by the module name. The keyword `create` indicates the source and target models [15]. After this step, the transformation rules are defined. Those rules are written using ATL syntax, are
Fig. 9 Example of modeling at the PSM level of abstraction

saved in files with the extension .atl and can use either a declarative or an imperative style. The code presented in Listing 1 shows one of the transformation rules. This rule is responsible for creating the classes that have geographic information, that in this case are represented by the GeoProfile stereotypes, and for creating the elements that were not contained in the CIM such as, for example, the geometry attribute, whose type need to conform with the ISO standard.

After the transformation of the PIM model, the output model is generated in the XML Metadata Interchange (XMI) format [26], which is a standard format for exchanging UML models among CASE tools.

5 Case Study

This section describes an example of using a customized GeoProfile in the Rational Software Modeler (RSM) CASE tool by IBM®. The study addresses a hypothetical system for managing a sugarcane crop, which has been widely used for biofuel production. This case study was chosen because it could use a large number of elements from the GeoProfile, giving to the reader a good understanding of the profile and how it can be used for domain engineering. Besides, fuel and renewable resources are subjects that are widely discussed nowadays. A brief description of the case study on cultivation of sugarcane for ethanol production is presented below.

The investment in the production of cleaner fuels that can replace, with no economic loss, traditional fuels (e.g., fossil) has been carried out with tax incentives from the Brazilian government. These new fuels, or biofuels, pollute less because the production process tends to be cleaner and have more balanced CO$_2$ emissions.
The main biofuel currently used in Brazil is the ethanol produced from sugarcane. However, its production requires a large amount of natural resources (e.g., areas planted with monoculture) for cultivation and subsequent production of ethanol. The planting, fertilizing and harvesting (e.g., manual or mechanized), as well as loading, transporting, weighing, unloading and cleaning operations, are crucial for a good industrial performance. Many environmental problems can arise in the production of ethanol, since most crops occupy wide contiguous areas, isolating and/or suppressing forest reserves, as well as the likely deforestation of catchments, siltation of streams, and others.

Based on the above description, one can realize that the problem involves important geographic phenomena that requires spatial analysis and therefore need to be stored in a database. As an example, we can mention some natural resources (e.g. soil, topography, vegetation, hydrography) and anthropogenic activities (e.g. production, transport, labor force).

From the description of the problem, a CIM model was initially developed (Fig. 10) using the UML GeoProfile. In that diagram, each class represents an entity of the real world and shows how these can or should be linked. For example, in the association element between the classes Farm and Plot, the specification of a spatial restriction (stereotype <<In>>) indicates that the spatial component of each Plot should be geometrically within the spatial component of a Farm.

Notice that the theme ReedPlantation deals with information related to sugarcane farms, such as the location, the division of a farm into plots, varieties that are grown in the farm, and the data on the farm’s owner.

Also, in this theme, the temporal association (1:1) between the classes Plot and Planting indicates that each plot should have only one type of variety, in a given period, but it can record temporal evolutions of the associations of this plot with other types of variety. That is, there cannot be two different sugarcane crops on the same plot in the same period of time.

Data about access roads and some natural phenomena such as topography, vegetation and hydrology are modeled in specific packages (themes). These phenomena have spatial components, so that information can be retrieved through spatial analysis operations, such as transport routes within a farm, calculation of buffer zones next to water courses, calculation of slope based on the relief and queries on vegetation types that occur in the area of the farm, etc.

The model in Fig. 10, however, is at a high level of abstraction; this is the CIM level in the MDA approach; as seen previously, it is useful for the user and the designer to understand the problem’s domain in question. The transformation process of this initial model into a PIM model, the next level of the MDA, follows the same way described in Section 4. Fig. 11 shows the PIM model resulting from this transformation.

Notice that in this step new specifications were added to the schema. For example, the temporal association (1:1) was transformed into an association (1:*), and one attribute was included in the class Planting to store the period in which a sugarcane crop was grown in a plot. Attributes related to the geometry and the object identifiers of the classes were also added.
In the next step the PIM is transformed into a PSM model, which is the lowest MDA level. As stated in Section 4, a PIM can be transformed into many PSMs, according to the platforms that will be used by the user. However, for this case study, only one example of mapping is shown, namely an object-relational database model. The PSM model resulting from this transformation is shown in Fig. 12.

With the model at the PSM level of abstraction, it is already possible to extract all the information needed to generate the database code. An example of the geographic database script generated from the theme ReedPlantation of Fig. 12 using the DBMS Oracle Spatial® is shown in Listing 2 of the Appendix.

6 Concluding Remarks

The development of the GeoProfile was mainly motivated by the fact that UML can be used, along with all its available resources, for example, CASE tools, to concep-
Fig. 11 Schema using the ISO 191xx standards (the PIM level)

tually model a geographic database. The GeoProfile has in its definition the main requirements for geographic applications and has the features of the main existing conceptual data models for GIS applications.

This chapter showed how GeoProfile meets the international standards for geographic information, using the ISO 191xx standards. The use of standards is essential in the geographic database project. The MDA approach made it possible to show how the GeoProfile is linked to the international standards.

The graphical notation of GeoProfile was customized in the RSM CASE tool, and some examples of conceptual schemes were modeled. The tendency is that the CASE tools, in general, start supporting this mechanism of UML extension, provid-
Fig. 12 Customized schema for the Object-Relational Model (the PSM level)
ing a greater number of options for the designer. Finally, a case study was presented, showing how to design a geographic database step by step, using the MDA approach with the aid of a CASE tool that supports an UML2.0. More information about GeoProfile, with examples of how to customize different CASE tools, can be obtained at www.dpi.ufv.br/projetos/geoprofile.

Acknowledgements This project was partially financed by CNPq - National Council for Technical and Scientific Development, MCT - Ministry of Science and Technology and FAPEMIG - Foundation for Research and Development of Minas Gerais.

Appendix

Listing 1 An example of an ATL transformation rule.

```
rule stereotypeClass
{
  from
    input : geoProfile!Class(
      not thisModule.emptyGeometry(input.stereotype))
  to
    output : ISO!Class(
      name <- input.name, reference <- input.reference ->
      collect(e | thisModule.getReferences(e)).asSet(),
      attribute <- input.attribute ->
      collect(e | thisModule.getAttributes(e) ).asSet(),
      attribute <- id,attribute <- geometry
    ),
    id : ISO!Attribute{
      name <- 'id' + input.name,
      type <- thisModule.integerDataType()
    },
    Geometry : ISO!Attribute{
      name <- input.name + 'Geometry',
      type <- if( thisModule.isPolygon( input.stereotype ) )
        then thisModule.polygonDataType()
        else thisModule.pointDataType()
      endif
    }
}
```
**Listing 2** Geographic database script generated from the theme ReedPlantation using the DBMS Oracle Spatial® (the PSM level).

```sql
CREATE TABLE CITY (  
  NAME VARCHAR(30),  
  POPULATION NUMBER,  
  IDCITY NUMBER,  
  GEOMETRY SDO_GEOMETRY,  
  CONSTRAINT pk_City PRIMARY KEY (IDCITY));
CREATE TABLE OWNER (  
  NAME VARCHAR(30),  
  GENDER VARCHAR(1),  
  MARITALSTATUS VARCHAR(10),  
  HOUSENUMBER NUMBER,  
  DISTRICT VARCHAR(30),  
  CITY VARCHAR(30),  
  STATE VARCHAR(30),  
  IDOWNER NUMBER,  
  CONSTRAINT pk_Owner PRIMARY KEY (IDOWNER));
CREATE TABLE FARM (  
  NAME VARCHAR(30),  
  AREA NUMBER,  
  IDFARM NUMBER,  
  GEOMETRY SDO_GEOMETRY,  
  IDCITY NUMBER,  
  IDOWNER NUMBER,  
  CONSTRAINT pk_Farm PRIMARY KEY (IDFARM),  
  CONSTRAINT fk_City FOREIGN KEY(IDCITY) REFERENCES CITY(IDCITY),  
  CONSTRAINT fk_Owner FOREIGN KEY(IDOWNER) REFERENCES OWNER(IDOWNER));
CREATE TABLE PLOT (  
  AREA NUMBER,  
  IDPLOT NUMBER,  
  GEOMETRY SDO_GEOMETRY,  
  IDFARM NUMBER,  
  CONSTRAINT pk_Plot PRIMARY KEY (IDPLOT),  
  CONSTRAINT fk_Farm FOREIGN KEY(IDFARM) REFERENCES FARM(IDFARM));
CREATE TABLE VARIETY (  
  KIND VARCHAR (30),  
  DESCRIPTION VARCHAR (30),  
  AVERAGEPRODUCTION NUMBER,  
  IDVARIETY NUMBER,  
  CONSTRAINT pk_Variety PRIMARY KEY (IDVARIETY));
CREATE TABLE PLANTING (  
  QUANTITYPRODUCED NUMBER,  
  GEOMETRY SDO_GEOMETRY,  
  IDPLOT NUMBER,  
  IDVARIETY NUMBER,  
  CONSTRAINT pk_Planting PRIMARY KEY (IDPLOT, IDVARIETY),  
  CONSTRAINT fk_Plot FOREIGN KEY(IDPLOT) REFERENCES PLOT(IDPLOT),  
  CONSTRAINT fk_Variety FOREIGN KEY (IDVARIETY)
```
References