

## Using CASE Tools in MDA Transformation of Geographical Database Schemas

Thiago Bicalho Ferreira, Jugurta Lisboa-Filho

Departamento de Informática  
Universidade Federal de Viçosa  
Viçosa, Minas Gerais, Brazil

e-mail: thiagao.ti@gmail.com, jugurta@ufv.br

Sergio Murilo Stempliuć

Faculdade Governador Ozanan Coelho (FAGOC)  
Ubá, Minas Gerais, Brazil  
e-mail: smstempliuć@gmail.com

**Abstract**— GeoProfile is a Unified Modeling Language (UML) profile proposed to standardize the conceptual modeling of geographic databases (GDB). GeoProfile can be used along with the Model Driven Architecture (MDA) added with integrity restrictions specified through the Object Constraint Language (OCL). Several Computer-Aided Software Engineering (CASE) tools provide support to those computational artifacts already consolidated by the UML infrastructure. Some CASE tools can be configured to automate the transformation of schemas at different levels of the MDA approach. The transformations in those tools occur from the Platform-Independent Model (PIM) abstraction level until to the generation of Structured Query Language (SQL) source codes. This study aimed to describe the evaluation process of a set of CASE tools with support to UML Profile technology based on specific requirements to use MDA approach, OCL restrictions, and other elements that aid in conceptual GDB modeling. This paper also describes an experience in using GeoProfile with one of the CASE tools evaluated, taking into account the tool's transformation language to allow for automated transformations among the different levels of the MDA approach.

**Keywords**- CASE tools; Enterprise Architect; MDA transformations; OCL; Geographical Database.

### I. INTRODUCTION

Based on the study developed in [1], this paper describes the assessment of CASE tools and the steps to reach vertical interoperability among schemas in the UML GeoProfile at the different levels of the Model Driven Architecture (MDA). GeoProfile is a Unified Modeling Language (UML) profile proposed by [2], employed in the conceptual modeling of Geographic Databases (GDB), which can use all of UML's infrastructure, which includes Object Constraint Language (OCL) to define integrity constraints and MDA for the transformation between its different abstraction levels [2][3]. Moreover, one of the advantages of using a UML profile is that it can be used in different CASE tools. However, not all tools offer the same features, making difficult to choose one. Examples of CASE tools with UML support include Enterprise Architect, Papyrus, StarUML, Visual Paradigm, and IBM Rational Software Architect.

In order to compare these tools and in the context of this study, some characteristics or features were prioritized such as the support to the UML Profile definition, validation of OCL constraints, and application of the MDA approach. The

key aspect is that the tools need to allow models to be created using UML GeoProfile, the transformation among the different levels established by the MDA approach, the syntactic and semantic validation of spatial OCL constraints, and that the models should be implemented from scripts generated for a selected database management system.

This paper aims to describe the evaluation of a set of CASE tools considering important requirements from the conceptual project to the implementation of the geographical database. This paper also reports on an experience of using the CASE Enterprise Architect tool, which had the best result, and on the challenges of the development of a vertical transformation mechanism for geographic databases using UML GeoProfile.

The remaining of the paper is structured as follows. Section II briefly explains the representation of geographical data, the UML GeoProfile, the MDA approach and the syntax to specify OCL expressions and the Oracle Spatial Geographic Database Management System (GDBMS). Section III presents a description of each CASE tool analyzed according to the goal of this study. Section IV shows the requirements, the methodology and the result of the tool evaluations. Section V presents the MDA transformation applied to GeoProfile in the CASE Enterprise Architect tool. Section VI describes the SQL code generation step in the Enterprise Architect tool for the Oracle Spatial GDBMS. Section VII presents the conclusions and future works.

### II. GEOGRAPHICAL DATABASE MODELING CONCEPTS

This section presents a literature review identifying the main concepts that contribute to the conceptual GDB modeling.

#### A. Representing Geographical Information in Computers

The representation of geographical space in computers is a challenge faced by researchers. According to Longley et al. [4], the world is infinitely complex and computing systems are finite, thus, it is up to the designer to limit the amount of details to be captured from the environment mapped. The two main approaches on computing are the continuous (fields) and discrete (objects) representations. Another representation also employed is in the form of networks, which takes into account graph theory.

Figure 1 shows part of a city with a sports center and represents part of this city focusing on the roads and the stadium. The GDB of Figure 1(b) must be conceptually modeled containing all structures of interest in the system while leaving aside other information such as the type of vegetation, vacant plots, terrain, and other characteristics that may be abstracted from Figure 1(a).

In order to design the conceptual data schema, first the vector structures used to represent the boundaries of each geographic entity must be understood, which is normally specified through basic geometric shapes: point, line and polygon (area) [5]. Figure 1(b) presents the use of these three types of vector structures. For instance, the stadium may be spatially represented as a point or as a polygon (multiple spatial representation); the main east road, as a line; and the sports center, as a polygon.

Additionally, presenting the structures, Figure 1(b) illustrates the relationship among the vector objects, which shows the stadium “is within” the sports center, the sports center “touches” the road to the stadium, the main west road “is near” the sports center, but does not “touch” it.

Such relationships are known as topological relationships and have been discussed by [6] and [7] and used by [8].

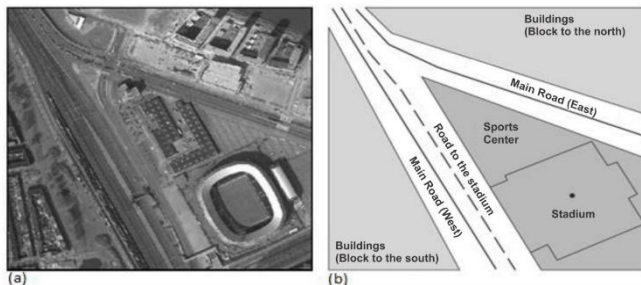


Figure 1. (a) Photograph of part of a city with a sports center between roads. (b) Spatial representation of this area. Source: Adapted from [9].

### B. Model-Driven Architecture (MDA)

According to Kleppe et al. [10], MDA is a framework standardized by the [11] for the development of software employing a Model-Driven Development (MDD) view.

The MDA approach consists of three abstraction levels, namely, CIM, PIM and PSM. Computation-Independent Model (CIM) does not show details of the system’s structure, but rather the environment in which the system will operate. Platform-Independent Model (PIM) is an independent model of any implementation technology containing the software requirements. Platform-Specific Model (PSM) specifies details about the platform in which it will be implemented. The artifacts produced by the MDA approach are formal models that can be processed by computers and, after undergoing transformations, will get to a final source-code step (top-down approach) or to high levels of abstraction (bottom-up approach). Figure 2 illustrates the action of transformation tools at MDA levels.

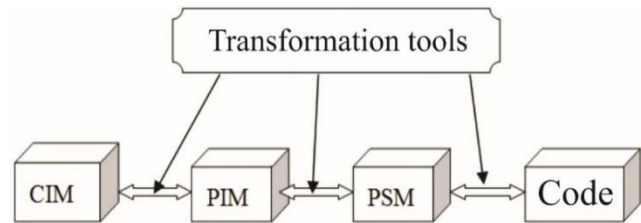


Figure 2. Use of transformation tools in the MDA approach. Source: Adapted from [10].

### C. Object Constraint Language (OCL)

Conceptual modeling makes the problem easier to be understood through abstraction, thus enabling risk management and contributing to error correction early in the project, which minimizes the cost of maintenance [12]. However, Warner and Kleppe [13] state that conceptual models may not be able to represent all requirements, resulting in problems to those who interpret them.

The OCL, adopted by OMG [14] since version 2.0, was defined as a formal language to complement the conceptual modeling using UML. Using OCL ambiguity-free integrity constraints can be created, which makes it possible to specify the data consistency wanted in the system at a high level of abstraction. Since it is a formal language, it can be processed by CASE tools until the source-code generation, which enables more powerful and satisfactory data consistency [13]. OCL is currently at version 2.4 [15].

The OCL expressions represent constraints that are needed in the system and not how they should be implemented. The evaluation of a constraint on the data always yields a Boolean value [13]. The syntax of a typical expression in OCL that represents a condition has the format presented by Code in Figure 3.

```

<context>
  inv:<expression>
  
```

Figure 3. Syntax of a typical expression in OCL.

Code in Figure 4 illustrates a hypothetical example of OCL constraint that specifies that a Brazilian municipality must be larger than 3,000 km<sup>2</sup> (note: The smallest Brazilian municipality, Santa Cruz de Minas, MG, is 3,565 km<sup>2</sup>). A detailed specification of the OCL can be found in [13] [15].

```

context Municipality
  inv:self.area > 3000
  
```

Figure 4. Hypothetical example of the use of OCL restrictions.

#### D. UML GeoProfile

In order to provide elements for specific domains without becoming excessively complex, UML has an extension mechanism called Profile. A UML Profile consists of: a metamodel; a set of stereotypes presented through texts in the form of <<text>> or through graphical icons called pictograms; tagged values; and constraints; all grouped in a stereotyped package called <<profile>>, thus formalizing the UML builder extension [16].

GeoProfile is a UML profile proposed for the geographical data modeling comprising the main characteristics of the existing models in the field [17]. GeoProfile is employed at the CIM and PIM levels of the MDA approach, using OCL constraints as a resource to validate the conceptual scheme generated by the designer [18].

The GeoProfile stereotypes are extensions of the Association and Class metaclasses. The stereotypes extended from the Class metaclass allow representing the geographic space in the discrete view (e.g., points, lined and polygons), in the continuous view (e.g., large cells and triangular networks), and through networks (nodes and arcs). The temporal aspects can also be represented with the stereotypes made up of tagged values that store instant and range values. The extended stereotypes of the Association metaclass allow representing topological relationships (e.g., touches and within) among the geographical stereotypes, and the temporal relationship (Temporal) among the temporal objects.

For the extended stereotypes of the Class metaclass, the abstracted stereotypes have been defined: <<Network>>, to group network stereotypes; <<GeoObject>>, to group the discrete view stereotypes; <<GeoField>>, to group the continuous view stereotypes; and <<Arc>>, to group the <<UnidirectionalArc>> and <<BidirectionalArc>> stereotypes that represent the possible links between the nodes of a network.

#### E. Oracle Spatial

Oracle Spatial is a database management system by Oracle Corporation that supports geographic information through the Spatial module. According to [9], Oracle supports three primitive geometry types (point, line, and polygon) and a collection made up of other primitive geometries. Figure 5 presents an example of Structured Query Language (SQL) involving the creation of tables with support to spatial data.

```
CREATE TABLE <table name>
{
    attribute_1 NUMBER,
    attribute_2 VARCHAR (25),
    geom          SDO_GEOMETRY
};
```

Figure 5. Table creation with spatial data support.

Unlike a conventional table, the code shown in Figure 5 has a geom attribute with the SDO\_GEOMETRY data type. According to [9], this data type is exclusive to Oracle GDBMS and is formed by five attributes that specify the geometry of the piece of data to be stored at the moment of its insertion, namely: SDO\_GTYPE, SDO\_SRID, SDO\_POINT, SDO\_ORDINATES, and SDO\_ELEM\_INFO. Further details on the use of each type of attribute can be obtained in [9].


### III. CASE TOOLS ANALYZED

The tools analyzed in this study were chosen according to the ease of access to the software and documentation. These tools are open source and commercial with some support to the UML profile and are well known by the software development community. The sub-sections below describe the results of the analysis made on the following CASE tools, exploring the resources they offer compared to the GeoProfile: Enterprise Architect (EA) version 12.0, Papyrus UML2 Modeler (Papyrus) version 1.12.3, StarUML–UML/MDA Platform (StarUML) version 5.0.2.1570, Visual Paradigm for UML (VP) version 10.2 and IBM Rational Software Architect (RSA) version 9.0.

#### A. Enterprise Architect (EA)

Enterprise Architect (EA) [19] is a commercial CASE tool licensed by Sparx Systems that allows the visual creation of UML profiles and insertion with syntactic validation of OCL expressions. EA does not offer resources for semantic validation of OCL expressions.

Additionally, being a modeling tool, it acts as an MDA transformation tool, with its own language for transformation between the model levels. This language can be modified so that the users are able to reach the last MDA approach level, the source code [20]. Since the modeling in this paper refers to GDB, the last MDA step is the Data Definition Language (DDL) source code, which EA is able to generate.

The GeoProfile stereotypes in the EA tool can be represented graphically  or textually <<point>>. The tool also offers resources for multiple stereotype representation, e.g., depending on the scale, a city may be modeled as a point or a polygon <<point, polygon>>.

The advantage at using EA is that it makes possible to specify some constraints. For example, it does not allow the insertion of extended stereotypes of the Class metaclass in Association elements and vice versa. The problem is that it allows the use of abstract stereotypes in conceptual models, e.g. the abstract GeoProfile stereotypes: <<Arc>>, <<GeoField>>, <<GeoObject>>, <<Network>> and <<NetworkObj>>.

#### B. Papyrus UML2 Modeler

Papyrus UML2 Modeler [21] is an open-source tool based on the Eclipse environment and licensed by Eclipse (Eclipse Public License). It has a visual environment to insert UML profiles, thus providing support to insertion and syntactic validation of OCL constraints. However, it does not semantically validate these constraints.

Adding graphical icons to the stereotypes is possible. Thus, a class or association can be represented by stereotypes as follows: only text, only graphical icon, or graphical icon and text. The Papyrus tool allows multiple representation to be specified through stereotypes, but, in case the graphical representation is used, only the first stereotype used by the designer is presented.

Additionally, restricting the use of abstract GeoProfile stereotypes in conceptual models, in this CASE tool other GeoProfile stereotypes can only be used with correct UML elements, i.e., an extended stereotype of the Association metaclass cannot be used in a class defined by the Class metaclass.

The Papyrus tool does not support the MDA approach, the transformation language and DDL code generation.

### C. StarUML

StarUML [22] is an open-source tool whose profile insertion is done through an Extensible Markup Language (XML) document. This tool does not support OCL and, despite being considered MDA, the features offered are incomplete. What it allows is transforming a model (PIM) into source code without going through the PSM. The source codes can be generated for the languages Java, C++ and C#. StarUML does not have a transformation language and the conceptual models produced from GeoProfile cannot be transformed into DDL source code.

Although multiple stereotype representation is not supported by the tool, the designer can choose between graphical and text representation, but only text is supported in associations. Therefore, the possible class stereotype representations are: textual, graphical, and textual and graphical. The tool can also restrict the use of abstract stereotypes at the same time that the others can be properly used with UML elements.

### D. Visual Paradigm for UML (VP)

With an intuitive modeling environment, the commercial tool Visual Paradigm for UML [23] supports the visual creation of UML profiles. The stereotypes can be presented graphically or textually, with support for multiple representation with the graphical ones.

The tool does not allow the use of extended stereotypes of different metaclasses, as described in Section III.A, however, it does allow abstract GeoProfile stereotypes to be used during conceptual modeling.

The tool allows incomplete MDA approach, transforming PIM straight into source code. Nevertheless, it does not support DDL code generation from UML class diagrams, just only from those created through the ER model. Thus, the GeoProfile conceptual models cannot be transformed into DDL code.

Also, this tool does not support the syntactic and semantic validation of OCL constraints on conceptual models created from GeoProfile.

### E. Rational Software Architect (RSA)

RSA [24] is a commercial CASE tool licensed by IBM that allows the visual creation of UML profiles. This tool supports

the use of profiles and is designed to allow syntactic and semantic validation of OCL constraints applied to UML diagrams.

The representations by the stereotypes in an association or class may take place as follows: only textual stereotype, only graphical stereotype, and representation by the textual and graphical stereotypes. However, the multiple representation by the stereotypes can take place in two ways: All stereotypes applied to the class or association must be in textual format or the first stereotype applied takes on the graphical format and the others on textual format.

The tool does not allow inserting extended stereotypes of the Class metaclass in association elements and vice versa, and the stereotypes defined as abstract in GeoProfile cannot be used in the UML elements.

RSA has incomplete support to MDA since it does not natively allow DDL source-code generation. Although there is a transformation mechanism in which the origin, target, and some settings regarding the mapping in the transformation from model into source code can be determined, RSA does not have an MDA transformation language. Therefore, with RSA's native features and mechanisms, these transformations cannot be performed on models created from the GeoProfile.

## IV. RESULTS OF THE CASE TOOLS COMPARISON

This section initially presents a set of requirements the CASE tools must meet to support conceptual GDB modeling based on the GeoProfile. Next, it presents the method used in the evaluation, the results and the final classification of the CASE tools analyzed.

This method originally proposed by Rosario and Santos Neto [25] was used in exploratory research involving software project management tools. This method was also applied by Câmara et al. [26] on comparison of development environments for systems of Volunteered Geographic Information (VGI).

### A. Requirements Survey

Based on the literature and on the descriptions of each CASE tool, this paper proposes requirements to evaluate which tool has the greatest number of features to support the GeoProfile use, aiming the transformation of data models at the different MDA levels and to specify integrity constraints at conceptual level using OCL. Table I lists these requirements.

### B. Evaluation Method of CASE Tools

In the context of this study, the requirements were classified as follows:

- Requirements that are Essentials: Weight 3;
- Requirements that are Important: Weight 2;
- Requirements that are Desirable: Weight 1.

Additionally, to the weight attributed to requirements, a scale must be defined for how well the tools satisfy each one. They may not satisfy (NS), partially satisfy (PS), or satisfy (S) a requirement. Therefore, the following scales can be attributed:

- Does not satisfy the requirement: A scale with value 0 is attributed;

- Partially satisfies the requirement: A scale with value 1 is attributed;
- Satisfies the requirement: A scale with value 2 is attributed.

Based on this evaluation, the classification of each tool was calculated by adding up the products of the importance weight (W) and the satisfaction scale (S) for each requirement (n), represented in Figure 6. Access [25] for more details for this method.

$$X = \sum_{i=1}^n S_i \cdot W_i$$

Figure 6. Formula used to calculate and sort CASE tools.

TABLE I. REQUIREMENTS TO EVALUATE CASE TOOLS

	Requirement description
Rq 01	Correct attribution of GeoProfile stereotypes in the UML elements
Rq 02	Restriction to the use of abstract stereotypes in elements of the model
Rq 03	Support to syntactic validation of OCL constraints
Rq 04	Support to semantic validation of OCL constraints
Rq 05	Support to MDA transformations
Rq 06	Support to transformation language
Rq 07	Support to graphical exhibition of profile stereotypes
Rq 08	Support to multiple representation through stereotypes
Rq09	Support to visual profile creation
Rq 10	Support to DDL code generation
Rq 11	Open-source tool

### C. Evaluation of the CASE Tools

In order to evaluate each CASE tool and its practical capacity regarding the theoretical functionalities predicted for a UML profile, particularly GeoProfile, the requirements presented in Table I were classified according to the following criteria:

- The requirements considered essential are those that support MDA;
- Requirements that aid in transformations between MDA models are considered important;
- Requirements that care for quality of the GDB models are considered important;
- Requirements that facilitate understanding and contribute to the adoption of the tool are considered desirable.

Table II presents the classification of the requirements regarding their importance level, which are *Essential*, *Important* or *Desirable*. Table III presents the way each CASE

tool satisfies the requirements of Table I. At the end, the summary of the evaluation based on Formula (4) is presented using the data from Tables II and III.

TABLE II. CLASSIFICATION OF THE REQUIREMENTS BASED ON THE IMPORTANCE LEVEL.

Importance	Requirements
Essential	Rq05
Important	Rq 01, Rq 03, Rq 04, Rq 06, Rq 08, Rq 10
Desirable	Rq 02, Rq 07, Rq 09, Rq 11

Table III shows the level of satisfaction for each of the CASE tools analyzed, considering each of the 11 requirements. A CASE tool may or may not support a requirement, or provide partial support. For example, EA offers full support for Rq 01. The assigned scale for this level of satisfaction is 2. Meanwhile, Rq 01 was classified as "important" in Table II, therefore receiving weight 2. So, when Formula (3) is applied, the sum of (scale x weight) is calculated for all requirements. Thus, the total sum for EA is 30. The same method was used for all the other tools.

An analysis of Table III shows that the Enterprise Architect tool was the one that best satisfied the requirements for the transformation of conceptual models so that the OCL constraints can be used in the tool. Since it has a customizable transformation language, the OCL constraints can be transformed into integrity constraints along with the SQL code generated in the last MDA level.

Another situation that can be observed in Table III is that the CASE tool RSA provides the best features to use the OCL constraints since it allows for both syntactic and semantic validations.

## V. MDA TRANSFORMATION APPLIED TO GEOPROFILE IN THE CASE ENTERPRISE ARCHITECT TOOL

Based on the evaluation described in the previous section and using the Enterprise Architect (EA) CASE tool version 11.0.1106, this study sought to reproduce the different abstraction levels specified by the MDA approach.

Although GeoProfile, in its definition, works with the modeling of a CIM, the transformations from CIM into PIM were considered unnecessary for this research since, during the creation of the diagrams, abstract and specific concepts were found to mix in the EA tool. The diagram created with GeoProfile in that tool can be considered a CIM due to the abstraction and use of textual stereotypes, while it can be considered a PIM for allowing the specification of the types of data of the class attributes. By classifying its diagrams as PIM, in Figure 7, the classes City and Deposit are polygon spatial objects. The CoalMine class has multiple spatial representations as a point and polygon, however, only one stereotype can be represented graphically. The CoalCompany class is a class with no geographic representation.

TABLE III. CLASSIFICATION OF THE CASE TOOLS

CASE	Enterprise Architect			Rational Software Architect			Visual Paradigm			Papyrus			StarUML		
	S	PS	NS	S	PS	NS	S	PS	NS	S	PS	NS	S	PS	NS
Rq 01	X			X			X			X			X		
Rq 02			X	X					X	X			X		
Rq 03	X			X					X	X					X
Rq 04			X	X					X			X			X
Rq 05	X				X			X				X		X	
Rq 06	X					X			X			X			X
Rq 07	X			X			X			X			X		
Rq 08	X			X			X			X					X
Rq 09	X			X			X			X					X
Rq 10	X					X	X					X			X
Rq 11			X			X			X	X			X		
Total	30			25			19			20			12		

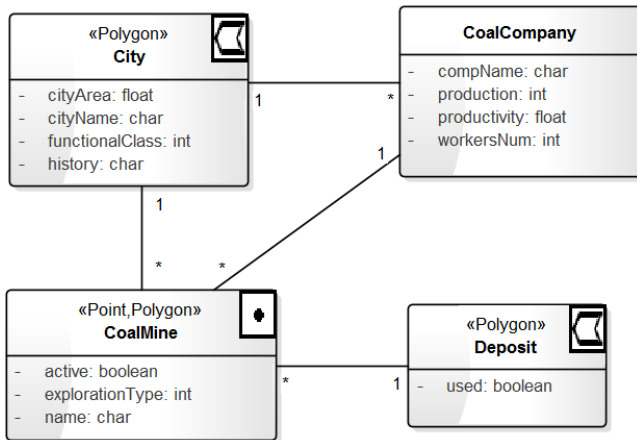


Figure 7. Diagram built in the Enterprise Architect tool based on UML GeoProfile, representing the PIM of the MDA approach.

After the PIM was obtained, the next step was to obtain the PSM. To that end, the tool provides transformation models in the Tools/Model Transformation (MDA) menu that spawn PSMs for C#, EJB, Java, DDL and other languages. However, since those are generic transformations, they only support data types common to those languages and not the specificities related to the use of stereotypes.

A transformation model into PSM closer to the generation of a database schema was the Data Definition Language (DDL) since it features transformation from class diagrams into table diagrams and allows for the transformation of data types according to the GDBMS selected. However, by default, DDL does not perform the transformation of class stereotypes,

but the Settings/Transformations Templates menu features source codes in a stereotypes language for the transformation models that can be modified for specific transformation tasks of the GeoProfile diagrams.

In addition to modifying the codes of the templates, one may also create new transformation models. For a new MDA transformation model from PIM into GeoProfile PSM, codes from the DDL transformation model were reused for the transformation of classes into tables, transformation of the relationships, and creation of primary and foreign keys. The code referring to the creation of packages, common to all transformation models, was also reused and only the name of the package to be created was changed.

The code presented in Figure 8 illustrates the creation of the GeoProfile\_PSM package and must be run whenever the GeoProfile transformation model is requested. The transformation of stereotypes was performed based on conditionals that assess the geographic type of a stereotype (point, line, polygon, etc.). Every geographic stereotype in a class diagram must become a column in its respective table. The code presented in Figure 9 illustrates the transformation of the stereotypes Point and Polygon and is similar to the transformation for the stereotype Line. For the Point stereotype, the column names are formed from the combination of the class name with the string Point, and the data type, which did not exist up until then (empty), converted into GM\_Point. The same occurs for the type Polygon, which is formed by a combination with the string Polygon, and the data type GM\_Polygon. Those are types of spatial data of Oracle GDBMS.

```

Package
{
    name = "GeoProfile_PSM" namespaceRoot = "true"
    %list="Namespace" @separator="\n\n" @indent= "  "%
}

```

Figure 8. Code to create the GeoProfile\_PSM package.

```

%IF classStereotype == "Point"%
COLUMN
{
    name = %q%% CONVERT_NAME (className,"Pascal Case", "Camel Case")%Point%qt%
    typ= %qt%%CONVERT_TYPE (genOptDefaultDatabase,GM_Point")%qt%
}
%endIf%

%IF classStereotype == "Polygon"%
COLUMN
{
    name = %qt%%CONVERT_NAME (className, "Pascal Case", Camel Case)%Polygon%qt%
    typ= %qt%%CONVERT_TYPE (genOptDefaultDatabase,GM_Polygon")%qt%
}
%endIf%

```

Figure 9. Code to transform classes with geographic stereotype.

Although the EA CASE tool allows the use of multiple stereotypes, they cannot all be processed during the transformation. It can be observed that, in Code 9, the first line has an *if* command to compare a variable *classStereotype* with a string (Point). That variable has only a string of the first stereotyped class and no solution has been found so far to capture the others. Figure 10 illustrates the CoalMine class, with multiple geographic representations, and the stereotype point separated by a comma from the stereotype Polygon. Figure 4 also presents the properties of that class, with special attention to the place where, despite being a combo box, it contains only the first stereotype (Point), also stored in the variable *classStereotype*.

Moreover, a code was developed for the transformation of the data types provided by the tool at the PIM level, e.g., Character and Integer, into data types recognized by the Oracle Spatial GDBMS. In case some type is different from those specified in the transformation, they are forwarded to the PSM as empty fields. The code presented in Figure 11 illustrates the transformation of the data type Character into Varchar.

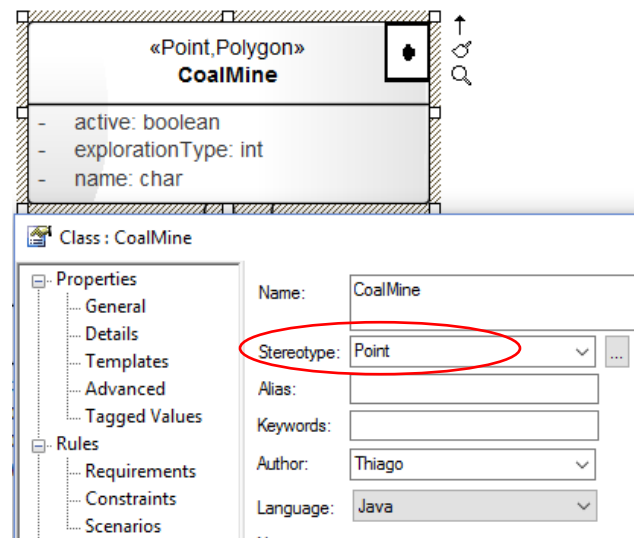


Figure 10. Issues in the geographic multirepresentation.

```

Column
{
  %TRANSFORM_CURRENT("type", stereotype), "collection", "constant", "containment",
    "ordered", "static", "volatile")%
  $type1 = %attType%

  %if $type1 == "Character"
    %Type = %qt%%
    CONVERT_TYPE(genOptDefaultDatabase, "Varchar2") %%qt%
  %endif%
}

```

Figure 11. Issues in the geographic multirepresentation.

However, there was no need to develop a code to attribute field sizes such as Varchar2(30) and Number(8,2) since the tool can be pre-configured to attribute values to those data types, as shown in Figure 12.

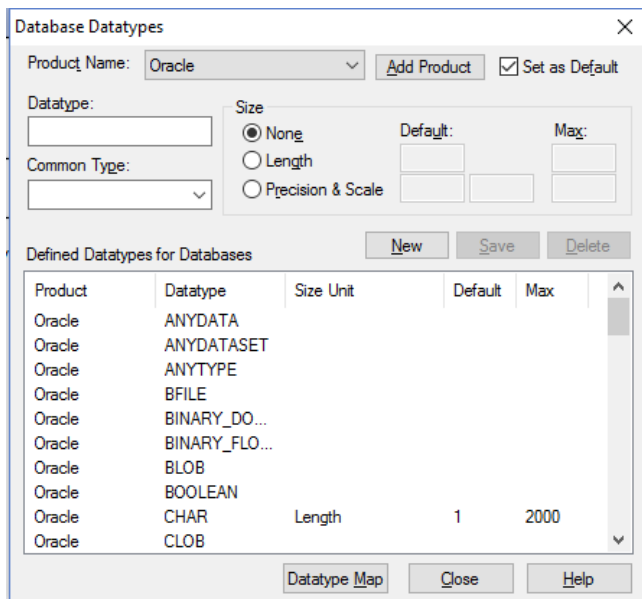


Figure 12. Customization of the data types of the attributes used during the transformation from PIM into PSM.

After following the steps (see the tutorial on the GeoProfile project's site <http://www.dpi.ufv.br/projetos/geoprofile>) for its incorporation into the EA tool and the concepts that have been presented so far, the Tools/Model Transformation (MDA) menu can be used and the GeoProfile\_PSM transformation can be selected so that the PSM of Figure 13 can be generated from the PIM of Figure 7.

After following the steps described for PSM generation and comparing the diagram illustrated in Figure 7 (PIM) with the diagram in Figure 13 (PSM), it can be seen that, for semantic relationships, i.e., those that do not involve topological relationships among the geographic objects, the foreign keys are automatically created in the classes. For example, for the semantic relationship between the classes City and CoalMine, the foreign key cityID was created in the last class.

The relational model today no longer poses great challenges regarding the transformation between the PIM and PSM models. However, the spatial characteristics, which involve new data types and topological relationships, add difficulties during this transformation, which requires the investigation of an extension of the rule set for the relational model. Another difficulty is the lack of standardization concerning the implementation of those data types and relationships in the different GDBMSs.

Consequently, it is helpful that the new transformation rules can be specified in the CASE tools, which is available in the EA tool from the possibility of customizing or creating a new transformation language. Figure 12 shows that new data types can be created during the transformation into a particular GDBMS type, which helps specify spatial attributes. Nonetheless, some complications can occur when the types temporal, network, and field view are taken into account since the concept itself is different from the conventional types.

Furthermore, the transformation of topological relationships into PSM is also more complex since they specify integrity restrictions among the spatial types involved and such task cannot simply be performed by creating foreign keys.



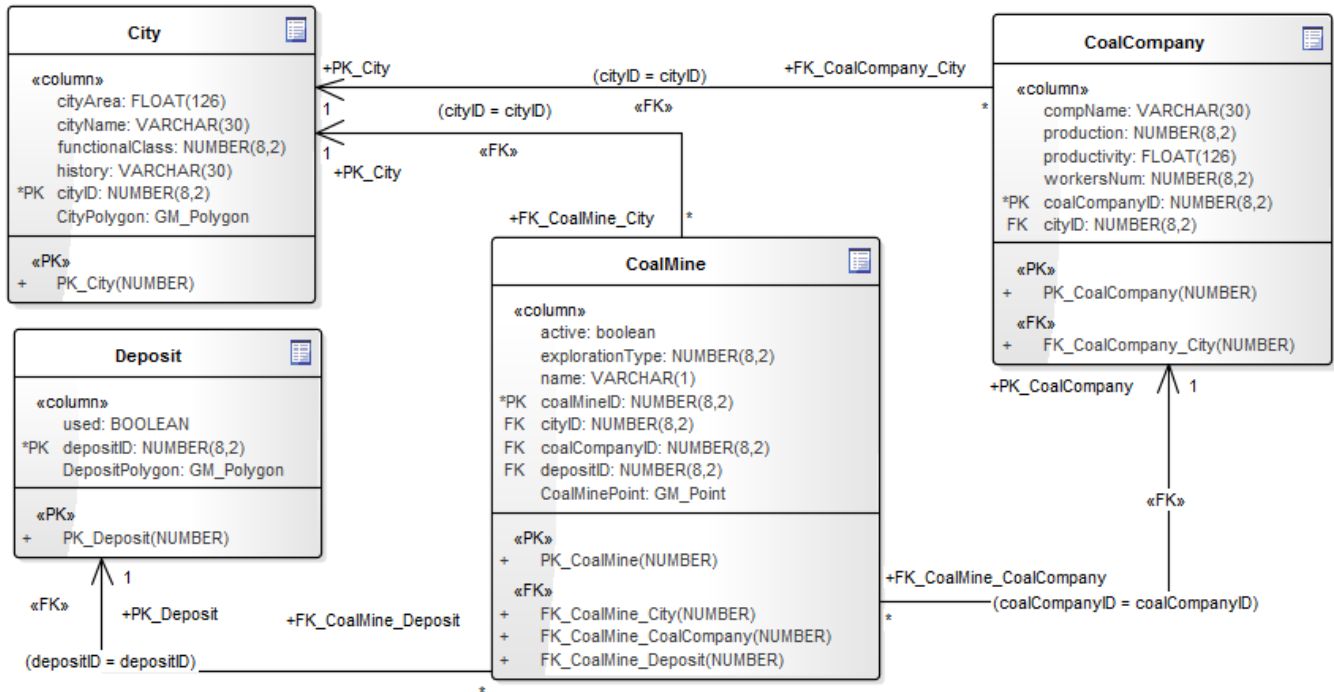


Figure 13. PSM generated from MDA transformations.

## VI. SQL CODE GENERATION FOR ORACLE SPATIAL GDBMS

The SQL code with the table structure creation script, relationships, and integrity restrictions is the last step of the MDA so that the implementation of what was initially specified at high abstraction level (CIM) can be automated. The EA tool natively provides the generation of the SQL script for the classes with the stereotype Table present in the PSM. In order to perform this task, one must select the properties of the GeoProfile\_PSM package and then use the transformation option Generate DDL so that the options for SQL code generation are provided. Figure 7 shows that, besides generating the table with the respective columns, the EA tool provides some options for SQL code generation. The steps are:

- In Figure 14, select the tables to be transformed into SQL codes;
- Indicate the place where the source code will be stored (in Single File);
- Select the options for SQL code generation, such as: Primary Key, Foreign Key, and Stored Procedures;
- Run the transformation of PSM into SQL code using the button Generate.

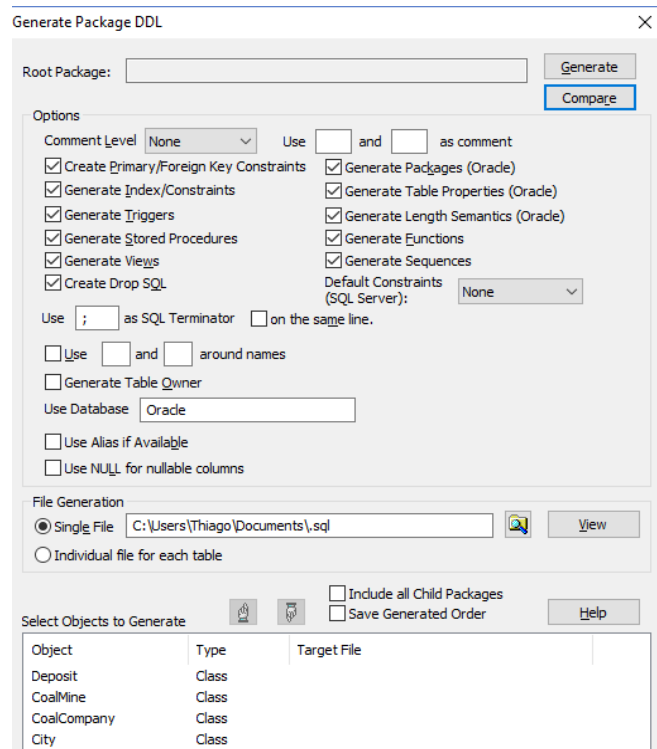


Figure 14. Window to generate SQL code in the GeoProfile\_PSM package.

Figures 15 and 16 show the SQL codes generated in the EA tool as the last artifact of the MDA approach. Both the code presented in Figure 15 and the one presented in Figure 16 originate in the PSM presented in Figure 13, however, they have been separated for better understanding and adequacy to the present study.

Figure 15 presents the SQL script responsible for the creation of the database tables, in this case, Oracle Spatial.

```
USE DATABASE Oracle;
DROP TABLE Deposit CASCADE CONSTRAINTS;
DROP TABLE CoalMine CASCADE CONSTRAINTS;
DROPTABLE CoalCompany CASCADE CONSTRAINTS;
DROP TABLE City CASCADE CONSTRAINTS;

CREATE TABLE Deposit
(
    used                BOOLEAN,
    depositID           NUMBER(8,2) NOT NULL,
    DepositPolygon      GM_Polygon
);

CREATE TABLE CoalMine
(
    Active              BOOLEAN,
    explorationType     NUMBER(8,2),
    name                VARCHAR(1),
    coalMineID          NUMBER(8,2) NOT NULL,
    cityID              NUMBER(8,2),
    coalCompanyID       NUMBER(8,2),
    depositID           NUMBER(8,2),
    CoalMinePoint       GM_Point
);

CREATE TABLE CoalCompany
(
    compName            VARCHAR(30),
    production          NUMBER(8,2),
    productivity         FLOAT(126),
    workersNum          NUMBER(8,2),
    coalCompanyID       NUMBER(8,2) NOT NULL,
    cityID              NUMBER(8,2)
);

CREATE TABLE City
(
    cityArea            FLOAT(126),
    cityName            VARCHAR(30),
    functionalClass     NUMBER(8,2),
    history             VARCHAR(30),
    cityID              NUMBER(8,2) NOT NULL,
    CityPolygon         GM_Polygon
);
```

Figure 15. DDL code for table creation.

The code presented in Figure 16 shows the changes made to the tables created for the inclusion of Primary Keys and Foreign Keys.

```
ALTER TABLE Deposit
    ADD CONSTRAINT PK_Deposit
    PRIMARY KEY (depositID)
    USING INDEX ;

ALTER TABLE CoalMine
    ADD CONSTRAINT PK_CoalMine
    PRIMARY KEY (coalMineID)
    USING INDEX;

ALTER TABLE CoalCompany
    ADD CONSTRAINT PK_CoalCompany
    PRIMARY KEY (coalCompanyID)
    USING INDEX;

ALTER TABLE City
    ADD CONSTRAINT PK_City
    PRIMARY KEY (cityID)
    USING INDEX;

ALTER TABLE CoalMine
    ADD CONSTRAINT FK_CoalMine_City
    FOREIGN KEY (cityID)
    REFERENCES City (cityID);

ALTER TABLE CoalMine
    ADD CONSTRAINT
        FK_CoalMine_CoalCompany
    FOREIGN KEY
        (coalCompanyID)
    REFERENCES
        CoalCompany (coalCompanyID);

ALTER TABLE CoalMine
    ADD CONSTRAINT
        FK_CoalMine_Deposit
    FOREIGN KEY
        (depositID)
    REFERENCES Deposit (depositID);

ALTER TABLE CoalCompany
    ADD CONSTRAINT
        FK_CoalCompany_City
    FOREIGN KEY (cityID)
    REFERENCES City (cityID);
```

Figure 16. DDL code for table alteration.

As shown in Figure 7 (representing the PIM), Figure 13 (representing the PSM), and by codes 15 and 16 (representing the SQL source code), the MDA transformation can be performed in the PIM, PSM, and spatial SQL source code steps using GeoProfile in the EA tool. However, the tool enforces some restrictions; in the example at hand, the transformation of the multiple geographic representation of

the CoalMine class. Since the SQL code is generated from the PSM model presented in Figure 6, the CoalMine table does not have the spatial features Polygon and Point initially specified in the PIM model. Despite this hurdle, the tool is able to generate an SQL code for implementation in the Oracle Spatial GDBMS.

The source code can be generated in the EA tool in several ways, however, the DDL model provides no customization option either for the PSM or SQL code generation. Customization for PSM generation was only achieved by using customizable templates, but, for now, SQL code generation used only direct transformation of the tool through the option Generate DDL.

It is evident that the option in the Settings/Code Generation Templates menu will also provide customizable languages and even the possibility of developing a new specific language to transform the PSM diagram of GeoProfile into a text file with the SQL source code for database creation.

## VII. CONCLUSIONS AND FUTURE WORK

From this paper, it is possible to observe that the tools evaluated do not have features to meet all the theoretical needs of UML, mainly regarding the use of profiles, MDA and OCL. However, they all support conceptual GDB modeling using GeoProfile.

The results of the comparison show that at the time this paper was written the EA could be considered the best CASE tool regarding transformations at the different MDA levels of models created using the GeoProfile. The RSA can be considered the tool that best supports OCL constraints due to its semantic validation, which makes the conceptual models less prone to errors. Among the free-software tools, Papyrus stood out compared to StarUML for supporting the GeoProfile.

Based on the results in this paper, a designer intending to use GeoProfile can know which CASE tool currently best meets the needs of the GDB project. However, it is important to point out that all tools analyzed are being constantly improved, which can change the results of this comparison at any moment.

This study also showed that diagrams created based on GeoProfile in the CASE Enterprise Architect tool can be subjected to MDA transformations from the PIM up to the SQL source code using customizable transformation languages. Despite some momentary issues, such as a lack of resources for the transformation of multiple stereotypes into a class, the tool provides interesting resources to automate the generation of all models of the MDA approach, which ensures higher fidelity between what is specified at a high level and what is actually implemented in the GDBMS.

The method employed, originally proposed by Paranhos and Santos Neto [33], can be used for different comparisons so that designers can establish their own requirements and assign importance weights and satisfaction scales to each one.

Proposals for future works include enhancing the transformation language presented to enable the transformation of all GeoProfile stereotypes, which included the temporal, network, field view, and topological relationships aspects. In addition, it must be observed that, from the PSM model, the transformation must consider the data types used by the different GDBMSs both for the conventional attributes and geographic and temporal ones.

Other work to be developed involves studies which are being done and aim to reach interoperability of conceptual geographical data models created from different conceptual metamodels specific for geographical databases, whose transformation base is the GeoProfile metamodel.

## ACKNOWLEDGEMENTS

Project partially funded by the Brazilians agencies FAPEMIG and CAPES. We also thank the support of Faculdade Governador Ozanan Coelho (FAGOC) and CEMIG.

## REFERENCES

- [1] T. B. Ferreira, J. Lisboa and S. M. Stempluc, "Evaluation of CASE Tools with UML Profile Support for Geographical Database Design," Proc. of Geoprocessing, Venice, Italy, 2016, pp. 1-6.
- [2] G. B. Sampaio, F. R. Nalon, and J. Lisboa-Filho, "GeoProfile-UML Profile for Conceptual Modeling of Geographic Databases," Proc. Int. Conf. on Enterprise Information Systems (ICEIS), Funchal-Madeira, Portugal, 2013, pp. 409-412.
- [3] F. R. Nalon, J. Lisboa-Filho, K. A. V. Borges, J. L. Braga, and M. V. A. Andrade, "Using MDA and a UML Profile integrated with International Standards to Model Geographic Databases," Proc. Brazilian Symposium on Geoinformatics (GeoInfo), 2010, pp. 146-157.
- [4] P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, *Geographic Information Science and Systems*, Danvers: Wiley, 2010.
- [5] G. Câmara, "Computational representation of spatial data," In: M. A. Casanova, G. Câmara, C. A. Davis Jr., L. Vinhas, and G. R. Queiroz (Org.), "Bancos de Dados Geográficos". Curitiba: EspaçoGeo, cap. 1, 2005, pp. 1-44. (in Portuguese)
- [6] E. Clementini, P. Di Felice, and P. Oosterom, "A Small Set of Formal Topological Relationships Suitable for End-User Interaction," Proc. Int. Symposium on Advances in Spatial Databases, 1993, p. 277-295.
- [7] M. J. Egenhofer and R. D. Franzosa, "Point-set topological spatial relations," *International Journal of Geographic Information Systems*, vol. 5, no. 2, 1991, pp. 161-174.
- [8] A. A. A. Ribeiro, S. M. Stempluc, and J. Lisboa-Filho, J., "Extending OCL to specify and validate integrity constraints in UML-GeoFrame conceptual data model," Proc. Int. Conf. on Enterprise Information Systems (ICEIS), Angers Loire Valley, France, 2013, pp. 329-336.
- [9] R. Kothuri, A. Godfrind, and E. Beinat, *Pro Oracle Spatial for Oracle Database 11g*, USA: Apress, 2007.
- [10] A. Kleppe, J. Warmer, and W. Bast, *MDA Explained: The Model Driven Architecture: Practice and Promise*, Boston: Addison-Wesley, 2nd ed., 2003.
- [11] OMG., *MDA Guide*, OMG Document formal/2003-06-01 edition, Needham, MA, USA, Version 1.0.1, 2003.
- [12] G. Booch, J. Rumbaugh, and I. Jacobson, *UML: user guide*, Elsevier. Rio de Janeiro, 2nd ed., 2005.
- [13] J. Warmer and A. Kleppe, *The Object Constraint Language: Getting Your Models Ready for MDA*, Boston: Addison Wesley, 2nd ed., 2003.

- [14] OMG., Object Constraint Language, OMG Document formal/2006-05-01 edition, Needham, MA, USA. Version 2.0, 2006.
- [15] OMG., Object Constraint Language, OMG Document formal/2014-02-03 edition, Needham, MA, USA. Version 2.4, 2014.
- [16] H. Eriksson, et al., "UML 2 Toolkit", Wiley Publishing. Indianapolis. 552p, 2004.
- [17] J. Lisboa Filho, G. B. Sampaio, F. R. Nalon, and K. A. V. Borges, "A UML profile for conceptual modeling," Proc. Int. Workshop on Domain Engineering (DE@CAISE), 2010, pp. 18-31.
- [18] F. R. Nalon, J. Lisboa-Filho, J. L. Braga, K. A. V. Borges, and M V. A. Andrade, "Applying the model driven architecture approach for geographic database design using a UML Profile and ISO standards," Journal of Information and Data Management, vol. 2, no. 2, 2011, pp. 171-180.
- [19] Sparx Systems, Enterprise Architect 12.1. [Online]. Available from: <http://www.sparxsystems.com.au/> 2016.04.13
- [20] T. B. Ferreira, S. M. Stempluc, and J. Lisboa-Filho, "Data Modeling with UML Profile GeoProfile and Transformations in MDA tool Enterprise Architect," Actas. Conferencia Ibérica de Sistemas y Tecnologías de Informacion (CISTI), Barcelona, AISTI | ISEGI, 2014, pp. 603-608.
- [21] Eclipse Foundation, Papyrus Modeling Environment. [Online]. Available from: <http://www.eclipse.org/papyrus/> 2016.04.13
- [22] Star UML, StarUML 2: A sophisticated software modeler. [Online]. Available from: <http://staruml.io/> 2016.04.13
- [23] Visual Paradigm International, Visual Paradigm, [Online]. Available from: <https://www.visual-paradigm.com/> 2016.04.13
- [24] IBM, Rational Software Modeler. Rational Software Architect Family. [Online]. Available from: <http://www-03.ibm.com/software/products/en/rational-software-architect-family> 2016.04.13
- [25] R. D. D. Paranhos and I. Santos Neto, Comparative study of change control tools in the software development process, Salvador: Universidade Católica do Salvador, 2009.
- [26] J. H. S. Câmara, T. Almeida, D. R. Carvalho, et al., "A comparative analysis of development environments for voluntary geographical information Web systems," Proc. of Brazilian Symposium Geoinformatics (GEOINFO), 2014, pp. 130-141