Modelling Geospatial Application Databases using UML-based Repositories Aligned with International Standards in Geomatics

Jean Brodeur[1,2], M.Sc., PhD cand.
Brodeur@rncan.gc.ca

Prof. Yvan Bédard[1], PhD
Yvan.Bedard@scg.ulaval.ca

Marie-Josée Proulx[1], M.Sc.
Marie-Josee.Proulx@scg.ulaval.ca

Centre for Research in Geomatics[1]
Laval University
Cité universitaire
Quebec City, QC Canada G1K 7P4

Centre for Topographic Information - Sherbrooke[2]
Geomatics Canada
2144 King West Street
Sherbrooke, QC Canada J1J 2E8

Abstract: This paper presents the result of recent work on the use of geospatial repositories to store the conceptual content of object oriented application database schemas and dictionaries aligned with international standards in geographic information (ISO/TC 211 and OGC). According to software engineering and database concepts, a geospatial repository can be defined as a collection of (meta) data structured in a manner to provide information about the semantics, geometry, temporality, and the integrity constraint of data stored in a geospatial database. For the last 6 years, ISO/TC 211 and OGC have been developing standards about geographic information to enable access and interoperability of geographic information; parts of these standards impact directly on geospatial repositories. Our work demonstrates that it is possible to develop a geospatial repository aligned with these standards and to implement it in a UML-based visual modelling tool. Practical examples are given based on the tool *Perceptory* which is a freeware developed at the CRG.

Keywords: data repository, feature catalogue, geographic information standard, geospatial database modelling, interoperability, metadata, UML, Perceptory, ISO/TC 211, OGC

1. Introduction

Geospatial data are increasingly available from multiple providers (governmental agencies, private organizations) and accessible on the Web. Often, organizations have collected geospatial data for their immediate internal needs (such as censuses, resource inventory, land management, and routing), accumulating huge amounts of data over time. From the outset and until recently, regularly geospatial data were disseminated and further used without a formal definition.

Now that the market for geospatial data is rapidly expanding, many geospatial data infrastructures have sprung up to support data access and use (e.g., NSDI in the US, CGDI in Canada). They provide one-stop shopping for people and organizations searching for data in a given country. The very availability and diversity of geospatial data sets make it hard for users to match data sets to their user needs. Users need more information and knowledge about data to determine data-set appropriateness to purposes.

Accordingly, metadata are now critically important. Metadata, defined as “data about data” [META group, Inmon, Hart], refer to different categories of information: content, lineage (source, collecting process, etc.), quality (positional and content accuracy, etc.), vintage, resolution, format, etc. They provide detailed knowledge about the data. Building data warehouses without metadata is basically futile since without them, it is difficult to recognize data signification and fitness for use. This can cause end users to erroneously interpret analysis results [META Group]. This sets the stage for the need to implement geospatial data repositories in sync with the most recent development in geomatics.

In addition, properly defining the semantics, geometry, temporality, and integrity constraints of objects to be included in a new dataset is an essential part of good database design [Bédard(a)]. Such a good practice has recently become more widespread worldwide in the GIS community. This paper presents recent work on a geospatial data repository to describe geospatial database content aligned with international standards in geomatics. It proposes that such geospatial data repositories be built with a new modelling tool called *Perceptory* [Bédard(b), Bédard and Proulx] and expressed with the Unified Modeling Language (UML) class diagram extended with spatial and temporal stereotypes [Rumbaugh *et al.*].

With this in mind, we begin by defining “data repository,” followed by an introduction to related topics from the international standardization work carried out by ISO/TC 211 – Geographic information/Geomatics (herein ISO/TC 211) and OpenGIS Consortium (OGC). The integration of ISO/TC 211 and OGC standards into *Perceptory* for the building of data repositories is presented in a metamodel and illustrated with an example. This demonstrates how such a repository can be used to describe geospatial databases in accordance with international standards.

2. Data repository[1]

The concept of a data repository has been thoroughly described for the last fifteen years, see for example [Moriarty, Prabandham, Jones] and standards were developed in the early 90s, although the work of these authors is contemporary with the emergence of geospatial data warehouses. They have been implemented in CASE tools (Computer-Assisted Software Engineering) and in the major system development environments (e.g. IBM, Oracle, HP). Repositories refer to the use of certain metadata to document database content down to a level of details which allows one to develop consistent, maintainable and clearly specified database. We can define data repositories as collections of metadata structured to provide the semantics and structure of the objects stored in database [Griffin]. In the case of geospatial databases, they also provide information about the geometric and temporal properties of objects as well as about the references system used. For example, data repositories include the names and definitions of object classes, their attributes' name and definition, descriptions of attribute values and domains, data types, operations, geometry (shapes and specifications, datum, map projection, etc.), temporality (dimensions and specifications, datum, units, resolutions, etc.), relationships, constraints, lineage information, and so on.

Schema and dictionary are essential data-repository components [Griffin]. Repositories must be browsable to retrieve the objects content description in order to identify fitness for use, to integrate data sets in whole or in part properly (*cf.* warehousing), to facilitate multisource updating, to support semantic interoperability of databases, and the like. If they adhere to different standards, data repositories will face major interoperability problems. Thus, standards related to geomatics and information technology must be followed to ensure metadata portability and interoperability (e.g. ISO/TC 211, OGC, ISO/IEC 10027 (IRDS), Microsoft Repository).

3. Standardization work in geomatics

ISO/TC 211 and OGC are the two leading international-level organizations, which have made significant progress in developing geomatics standards. These standards introduce rules for building schemas, data types, cataloguing methodology, metadata models, and so on. The two next section summarize their results at the time of writing this paper.

[1] Data repository is used here in a wider perspective to enable object documentation includind operations.

ISO/TC 211

ISO/TC 211 is the ISO technical committee responsible for defining international standards in the field of digital geographic information. Committee's works include developing the methods, rules, and services needed to acquire, process, manage, analyze, and access geospatial data. They provide the foundations for developing geospatial applications. Table 1 lists ISO/TC 211's ongoing works. These items in bold affect the development of geospatial data repositories and are explained in the next paragraphs.

ISO 19103, *Conceptual schema language*, relates to selecting a conceptual schema language that fulfills the needs of geographic information models and ISO/TC 211's needs with respect to standards development. ISO/TC 211 has retained the static structure diagram (i.e. the class diagram) of the Unified Modelling Language (UML). This work also provides guidance in the use of UML to ensure interoperability between geospatial models [ISO 19103]. Development of geospatial repositories shall consider the use of UML basic constructs as retained by ISO 19103.

ISO 19107, *Spatial schema*, defines a set of standard spatial data types and operations for geometric and topologic spaces. Geometry supplies the means to describe shapes of objects with coordinates and mathematical functions. Geometric data types are classified into three subclasses: base geometry (GM_Primitive), complex geometry (GM_Complex), and multiple geometry (GM_Aggregate). Topology describes the property of geometry that remains invariant when the space is transformed. Topologic data types are subdivided into base topology (TP_Primitive) and complex topology (TP_Complex) [ISO 19107]. Definition of geometric characteristics in geospatial repositories shall refer to these standard data types.

Table 1: ISO/TC 211 Work Items

ISO 19101	*Geographic information - Reference model*	*ISO 19114*	*Geographic information - Quality evaluation procedures*
ISO 19102	*Geographic information - Overview*	***ISO 19115***	***Geographic information - Metadata***
ISO 19103	***Geographic information - Conceptual schema language***	*ISO 19116*	*Geographic information - Positioning services*
ISO 19104	*Geographic information - Terminology*	*ISO 19117*	*Geographic information - Portrayal*
ISO 19105	*Geographic information - Conformance and testing*	***ISO 19118***	***Geographic information - Encoding***
ISO 19106	*Geographic information - Profiles*	*ISO 19119*	*Geographic information - Services*
ISO 19107	***Geographic information - Spatial schema***	*ISO 19120*	*Geographic information - Functional standards*
ISO 19108	***Geographic information - Temporal schema***	*ISO 19121*	*Geographic information - Imagery and gridded data*
ISO 19109	***Geographic information - Rules for application schema***	*ISO 19122*	*Geographic information/Geomatics - Qualifications and Certification of Personnel*
ISO 19110	***Geographic information - Feature cataloguing methodology***	*ISO 19123*	*Geographic information - Schema for coverage geometry and functions*
ISO 19111	***Geographic information - Spatial referencing by coordinates***	*ISO 19124*	*Geographic information - Imagery and gridded data components*
ISO 19112	*Geographic information - Spatial referencing by geographic identifiers*	*ISO 19125*	*Geographic information - Simple feature access - SQL option*
ISO 19113	*Geographic information - Quality principles*		

ISO 19108, *Temporal schema*, is the counterpart of the spatial schema. It defines temporal characteristics and functions needed to describe events that occur in the time space within the geospatial context. The temporal schema provides geometric- and topologic-like data types, respectively TM_GeometricPrimitive and TM_TopologicalPrimitive [ISO 19108]. Again, definition of temporal characteristics in geospatial repositories shall refer to these standard data types to enable temporal information interoperability.

ISO 19109, *Rules for application schema*, defines the General Feature Model (GFM), which is the metamodel for abstracting real-world features. Application schema rules provide the principles about the abstraction process and the realization of application schemas that document one perception of the reality. This standard bonds together parts of the ISO/TC 211 suite of standards since it describes how they shall be used in developing application schemas. It describes the instantiation of real-world features in application schemas, the use of spatial and temporal data types, the inclusion of metadata and quality data, and so forth [ISO 19109]. This standard influences both the definition of application schemas and the geospatial repositories development in the way that rules can be seen as modelling tool requirements.

ISO 19110, *Feature cataloguing methodology*, defines a metamodel for documenting real-world features. Spatial and temporal characteristics are left out of the scope of this standard and are introduced in ISO 19109. ISO 19110 provides the interface definition or export view for semantics information sharing.

ISO 19111, *Spatial referencing by coordinates*, defines the methodology for documenting coordinate reference system supporting the description of a position. This document provides essential elements to be inserted in data repository for the documentation of the spatial reference framework associated with spatial data.

ISO 19115, *Metadata*, defines the contents and structure of metadata components for describing data sets. Description of metadata elements in a repository must be aligned with the requirement of that standard.

OGC (OpenGIS Consortium)

OGC is an organization involved with the development of specifications to overcome the problems hindering interoperability. OGC's long-term vision is to provide a complete integration in geospatial data and geospatial processing. OGC works are aligned with ISO/TC 211. These organizations have developed a co-operative agreement for the harmonisation of their mutual works and the development of future works. Table 2 shows undergoing OGC topics. Those in bold face are related to the development of geospatial repositories and explained hereafter.

Table 2: OGC topics

Topic 0	*Abstract Specification Overview*	*Topic 10*	*Feature Collections*
Topic 1	***Feature Geometry***	***Topic 11***	***Metadata***
Topic 2	***Spatial Reference Systems***	*Topic 12*	*The OpenGIS™ Service Architecture*
Topic 3	*Locational Geometry Structures*	*Topic 13*	*Catalog Services*
Topic 4	*Stored Functions and Interpolation*	*Topic 14*	*Semantics and Information Communities*
Topic 5	***The OpenGIS™ Feature***	*Topic 15*	*Image Exploitation Services*
Topic 6	*The Coverage Type*	*Topic 16*	*Image Coordinate Transformation Services*
Topic 7	*Earth Imagery Case*	*Tbd*	*Telecommunications SIG*
Topic 8	***Relationships Between Features***	*Tbd*	*WWW Mapping SIG*
Topic 9	*Quality*	*Tbd*	*Transportation SIG*

Feature Geometry (Topic 1) is the abstract specification of the geometric description of features. Its content is tightly aligned with ISO 19107 and as such provides data types for geometric information to be used in data repository.

Spatial Reference Systems (Topic 2) defines the model for representing and documenting a spatial reference, and the linkage of the spatial reference system with an instance of coordinates. This topic impact on data repository development in such that it provides the principles to associate the semantics on the values used to describe a direct position on the Earth.

The purpose of OpenGIS™ Feature (Topic 5) is mainly the description of the abstraction process of real-world features. It describes how to arrive at feature concepts starting from the identification of real-world features. Topic 5 specifies the implementation of feature that has to be considered in the perspective of data repository.

Since a feature does not always exist independently from others, OGC Topic 8 deals with relationships between features. It describes the abstraction of relationships between real-world features. Data repository in constrained by Topic 8 document for the representation of feature relationships.

Topic 11, Metadata, explains why metadata are essential to geographic information and outlines an abstract model for handling them. Metadata content element can be obtained from other standardized lists such as ISO 19115 or FGDC. Topic 11 influences the way metadata shall be introduced in the development of data repository.

All these topics provide basic requirements for the development of data repositories that could be interoperable with others.

4. Geospatial data repository: the example of Perceptory

Database deployment typically starts with the analysis process. Analysis refers to the action of understanding and describing user needs within a spatial database [Bédard(b)]. Conceptual schemas are recognized to be well-suited to expressing an important part of the results of the analysis process. It provides expressions of real-world phenomena in term of user perception, and organizes them into classes, characteristics, relationships, and operations.

Building conceptual schemas involves graphically representing the concepts of interest for the application and defining their semantics. Typically, this is accomplished with a formalism (e.g., Chen entity relationship, OML, UML) and their underneath semantics is detailed in a join document called dictionary. Such dictionaries also include information needed for code generation (e.g. data type, domains, and identifiers). The schema and the dictionary are the two components of a repository.

Geospatial database modelling has some peculiarities not common in information technology. In particular, the analysis process aims leaving out details of low importance. Thus, if we consider that GIS systems or DBMS spatial cartridges will eventually handle geospatial data with their built-in geometric data structure, there is no reason to redesign the spatial and/or spatio-temporal model in the conceptual schema.

Accordingly, it is desirable for geospatial repositories to support the encapsulation of geometric details and to offer simple solutions using evocative spatial and temporal constructs. Excessive complexity impede both the thinking process and the development process. As Hohnman states it "too much structure kills mental process" [Hohnman].

Based on twelve years of experiences and research, a symbiotic philosophy leading to simplicity has guided the development of *Perceptory*. The name *Perceptory* is derived from *perception*, referring to the process of phenomenon representation, and *repository*, for knowledge documentation. *Perceptory* aims at capturing and managing the representation of user's perceptions in the most natural way of thinking and to facilitate the development of the database supporting these perceptions.

Perceptory is a visual modelling tool (also called CASE) that comprises a object oriented conceptual schema building tool for geospatial conceptual models and an object dictionary database. It features spatial, temporal, and spatio-temporal information stereotypes. These introduce additional details for classes and attributes and associations, which are necessary for programming the database. The dictionary functionalities include capturing, storing, and browsing descriptions of the elements depicted in the object oriented schema as well as additional details, which are necessary for database programming or for non-ambiguous data acquisition. In other words, Perceptory is based on a repository especially developed for spatial and spatio-temporal databases.

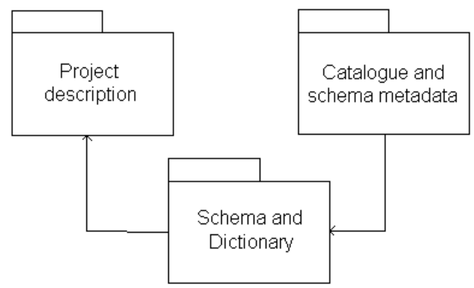


Figure 1: Perceptory's components

Perceptory is composed of three components: project description, catalogue and schema metadata, and schema and dictionary (Figure 1). This paper focuses on the presentation of the latter. The following sections provide a detailed description of the formalism retained for UML-based conceptual schemas, and the data repository structure with a special emphasis on spatial and temporal components provided with PVL (Plug-in for Visual Language) extensions (see Bédard(b) for an explanation of the PVL concept).

Components of the UML-based conceptual schema

Conceptual schemas are built to represent an abstraction of a subset of real-world features of interest to the users of a database. It is typically expressed through the use of a formal language. There are two types of formal languages: lexical and graphical. A lexical language uses textual like descriptions that agree to an underlying grammar (e.g. Express, Bakus-Naur Formalism or BNF). A graphical language (e.g. Chen E/R, DFD, UML) is made of graphical constructs to represent model elements also supported with an underlying grammar. Graphical languages are usually narrower in scope than lexical languages.

Perceptory uses a graphical language which build on UML class diagram [OMG(a)]. It introduces a novel approach based on the use of UML stereotypes to handle spatial and temporal properties into conceptual schemas. Many reasons led to the adoption of UML. First it was rapidly recognized as a *de facto* standard in the information technology community. It is also an OMG (Object Management Group) standard [OMG(c)]. In addition, ISO/TC 211 and OGC have already chosen UML for geospatial data modelling and it is becoming widespread in the GIS community. Finally, UML provides an extension mechanism made especially for the definition of new model element such as spatial and spatio-temporal elements. Essentially, *Perceptory* model elements are those required by ISO/TC 211 [ISO 19103]: class, attribute, operation, package, association, generalization/specialization, constraint, note, and stereotype (Figure 2).

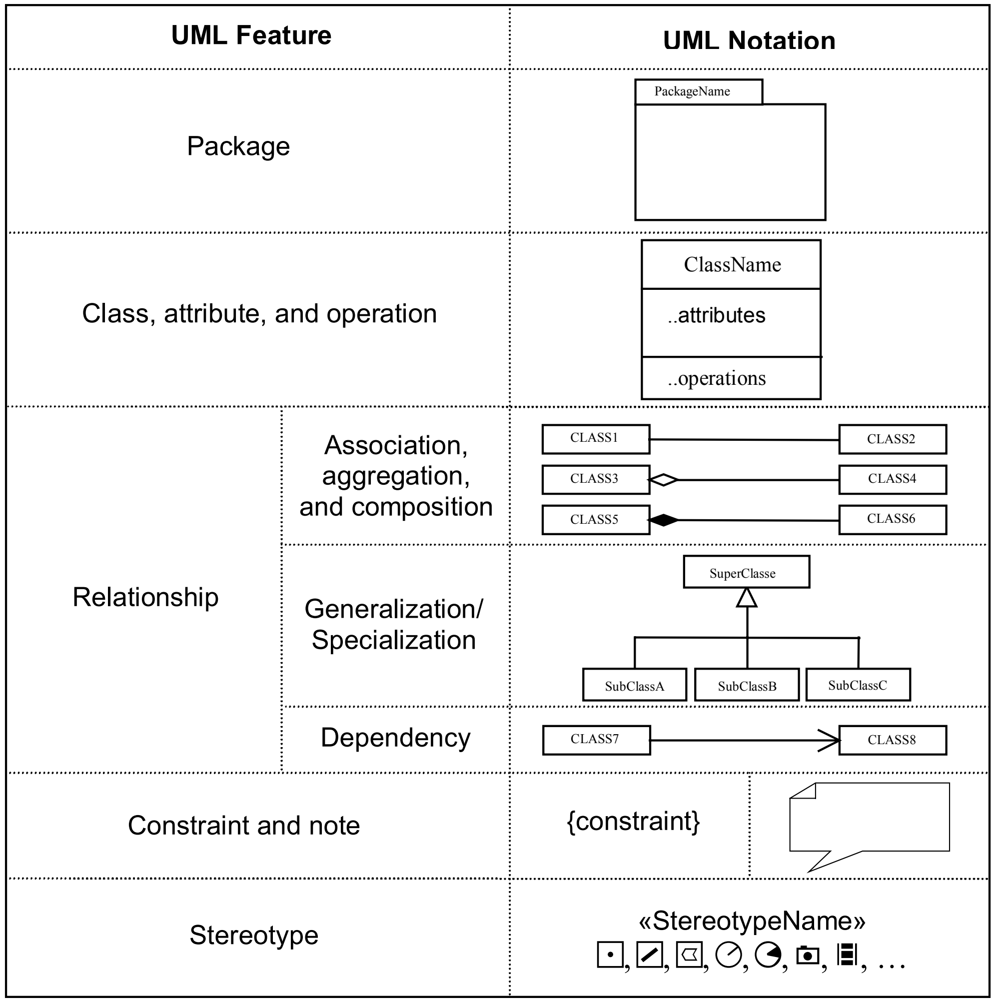


Figure 2: Perceptory model element and graphic notation

Dictionary

The dictionary is a collection of metadata that records knowledge about the data stored in the application databases or legacy systems. This metadata collection can be recorded in a word processor-like file, in a spreadsheet file, or, even better, in a database environment.

The dictionary is typically populated during the analysis process. The designer is responsible for getting consensus from the parties involved in the database development. Database items have to be defined carefully, eliminating out as much confusion as possible about the content. To that aim, classes, attributes, operations, attribute values, and so on must be defined. This sets up the content specification. Defining database content is usually an iterative and incremental process.

Geospatial data repository metamodel

These considerations were taken into account by recent research work proposing a metamodel for geospatial data repositories (Figure 3) that takes benefits of these driving forces. As such, it uses UML object class metamodel elements (see OMG(b) for the UML metamodels) and integrates ISO/TC211 as well as OGIS necessary elements, to fit within Perceptory, the needs of spatial and spatio-temporal database analysts and designer. The metamodel defines geospatial data repository structure and content.

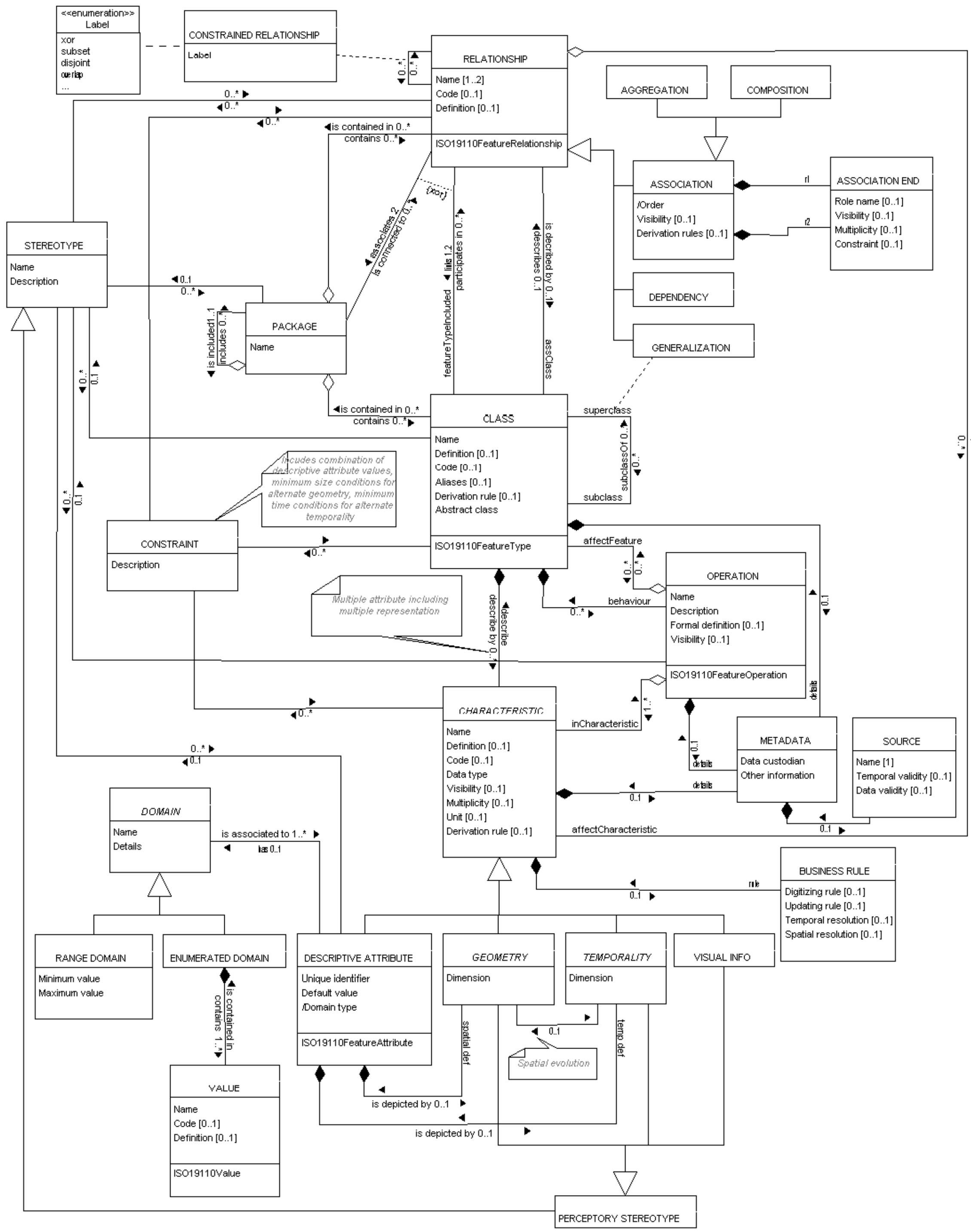


Figure 3: Perceptory's repository metamodel[2]

In Perceptory, the CLASS is the centrepiece metamodel. An object class describes a set of tangible phenomena, living beings, concepts, or events of the real world which are of interest to the users of a system and which are group into the same category, having the same set of characteristics and operations. According to [ISO 19110], it has a *name* and, optionally, a

2 This figure draws the metamodel of the next Perceptory's repository version. The actual version is a subset of this metamodel.

definition, a *code* (unique identifier), and a set of *aliases*. Usually, a spatial database class is persistent and its instances are obtained from measurements (e.g. from photogrammetry or field surveys). However, it can be derived from other classes as specified by a predefined *derivation rule*. A class that is defined as “not-instantiable” at the conceptual level (e.g. some superclasses resulting from generalisation, some derived classes) is an *abstract class*. In Figure 3, the *ISO19110FeatureType* operation of CLASS repackages together all information and returns a FC_FeatureType object that complies with ISO 19110.

A class usually has CHARACTERISTICS. A characteristic is a descriptor of one distinct character of that class which values may vary from one instance to the others. It shall be unique within the context of the class and its superclasses [ISO 19103]. A characteristic shall have a *name* and a *data type* [ISO 19110]. *Definition*, *code*, *visibility*, *multiplicity*, and *measurement unit* are optional details. *Visibility* refers to the level of attribute encapsulation (public, protected, or private). *Multiplicity* is the min and max number of times the characteristic may appear for an object (the default value is “1..1” or simply “1”). A *derivation rule* is included when the value can be obtained from calculation. METADATA give *details* such as the *data custodian* responsible for the information, *other information* (unstructured information of interest), and the description of source data (*name* and *validity*). BUSINESS RULES describe data acquisition specifications and lineage characteristics. Four types of characteristics are introduced: DESCRIPTIVE ATTRIBUTE, GEOMETRY, TEMPORALITY, and VISUAL INFO.

DESCRIPTIVE ATTRIBUTES are those characteristics which are not used for the geometry, temporality, and visual information of object classes but which are rather used as the semantic properties. They refer to the so-called thematic attributes of GIS and DBMS. Three aspects are important for descriptive attributes: *unique identifier*, *default value*, and *domain type*. *Unique identifier* is a Boolean value set to *True* when the characteristic plays such a role. *Default value* is the value assigned to the attribute when no value is supplied at the instantiation time. *Domain type* is a derived attribute. The value comes from the subtypes of the DOMAIN class: range, or enumerated. RANGE DOMAIN is characterized by a lower (minimum value) and upper (maximum value) bound value pair. ENUMERATED DOMAIN is a list of all possible values. In this case, each value is identified by its *name*. A *code* and *definition* are optionally supplied [ISO 19110]. As per CLASS, *ISO19110FeatureAttribute* and *ISO19110Value* operations return ISO 19110 compliant FC_FeatureAttribute and FC_Value objects, respectively.

GEOMETRY, TEMPORALITY, and VISUAL INFO are specialized characteristics describing, respectively, the object’s position and shape, the instant time or the span life, and multimedia type of information. Geometry and temporality will be discussed below. Image, picture, drawing, and video are media from which one or a collection of features can be recognized as well as some characteristics not necessarily explicited in a database field and in certain cases the surrounding environment of an object. Aerial photos and satellite images are good examples of multimedia information.

OPERATION describes an object’s behaviour. An operation shall have a *name* and a *description*. The *description* is given in natural language and provides operation's purpose, participating objects and characteristics, and the operation's algorithm. A *formal definition*–the operation description in a formal language (computer or scientific notation, such as BNF and Z) [ISO 19110]–is added optionally. The *visibility* describes the level of encapsulation (public, protected, or private). Associated METADATA add details on operation parameters. Input values also come from CHARACTERISTIC. An operation’s outcomes can affect other feature classes. *ISO19110FeatureOperation* returns an ISO 19110 FC_FeatureOperation object.

A relation is a structural or logical link that ties two concepts within a specific context. A relation is identified by *names*, one for each direction (at least one), a *code*, and a *definition*. Two or more relations can be constrained. For instance, the case when only one of multiple relations is applicable (*xor* constraint) and the case when one relation is the subset of another (*subset* constraint). Again, *ISO19110FeatureRelationship* returns an ISO 19110 FC_FeatureRelationship object. Relationships are specialized by ASSOCIATION, GENERALIZATION and DEPENDENCY.

ASSOCIATION is a structural relationship between two classes. An *order* indicator tells if elements are ordered. It is derived form constraints on association ends. Associations have optional *visibility* and *derivation rules*. A derivation rule provides an equivalent path to that association. Two association ends bound an association. ASSOCIATION END identifies the role *(role name)* that a class plays in an association with its related details: *visibility*, *multiplicity*, and *constraint* (basically an ordering constraint).

AGGREGATION and COMPOSITION are specialized types of association. Aggregation and composition are non-symmetric relationships where one association end has a predominant role over the other. In fact, the general meaning of these associations is respectively "*Have*" and "*Contain*" (or "*Part of*" when reading in the opposite direction). One object of class A, acting as container, contains *N* objects of class B, acting as "containees." In an aggregation relationship, the "containees" may exist by themselves and survive to the destruction of the container objects. It is called a weak aggregation [OMG(a)]. For example, a car's wheels can survive the destruction of the vehicle and still exist as wheels. On the other hand, in a composition relationship, "containees" cannot exist without the container. When the container is deleted, the "containees" are also deleted. This is referred to as strong aggregation [OMG(a)]. For example, when a house is deleted, its constituent parts (roof, walls, etc.) are also deleted.

Generalization/specialization is a typological relationship between classes of objects. Its fundamental meaning is "IS A." For example, *ocean* is a *waterbody*; *lake* is a *waterbody*. It is used to build a hierarchy between classes. It introduces inheritance of superclass properties (characteristics, operations, and relationships) by subclasses.

A DEPENDENCY relationship is a client/server type of relationship. It is a unidirectional relationship that links two model elements where one is dependent on the other. This relationship is used when one model element makes use of another. As an example, we use such a relation in the geometry metamodel (Figure 5) where the GEOMETRY class depends on the SPATIALREFFRAMEWORK class of the PROJECT DESCRIPTION package for describing the datum and the coordinate system.

A PACKAGE is an organizing element that delivers the capability of splitting a schema into logical modules, which can contains N levels of submodules (or smaller packages). Modules can be complete subsystem schemas. Packages can be dependent on other packages and they are identified by their *name*.

A constraint is a semantic relationship that specifies a condition or proposition that must be true [OMG(a)]. Restriction on the use of descriptive attribute values and on object depiction are examples of constraints. For example, let us say we have a class *ROAD* characterized by *Category*, *NumberOfLanes*, and linear geometry. The constraints could be:

```
Road
self.NumberOfLanes >= 4 implies self.Category = #highway
self.Category = #street implies self.geometry.length() >= 100
```

Constraints can be described in natural language to make them easier to read for the majority of users. However some specialists prefer to use formal languages such as Object Constraint Language (OCL), which convey an unambiguous meaning and are easier to manipulate by computers.

UML introduces the stereotype mechanism for extensibility purposes. Stereotypes make it possible to refine or specialize the semantics of model elements. The designer has the flexibility of developing his own modelling elements such as specialized packages, classes, and characteristics. *Perceptory* uses this mechanism to introduce spatial, temporal, and multimedia stereotypes in defining classes and descriptive attributes. A stereotype is identified by a *name* and has a *description*. The description defines the semantics of the stereotype, i.e., the nature of the stereotype, and if it is abstract or not. Classes, relationships, descriptive attributes, operation, and package can be stereotyped.

Spatial and Temporal Definition

Perceptory introduces spatial and temporal information using PVL (Plug-in for Visual Languages), which is a simple but powerful graphical notation depicting geometric and temporal properties of objects and attributes [see Bédard(b), Bédard and Proulx for more details]. It offers a high level of abstraction to facilitate analysis/conceptual spatial and temporal modelling without bothering with implementation issues. Physical implementation is left out, leaving the designers concentrated on real-world features. PVL can be used with any modelling technique, it uses five basic constructs represented by pictograms (Table 3) which can be combined four different ways (simple, alternative, complex, multiple) with cardinalities in order to properly represent spatial, temporal and spatio-temporal properties of objects and attributes.

Table 3: Perceptory pictograms

	0-dimensional geometry, point
	1-dimensional geometry, line
	2-dimensional geometry, area
	0-dimensional temporality, punctual time
	1-dimensional temporality, durable time

In Percewptory, the stereotype extension mechanism of UML is used to introduce PVL pictograms in conceptual database schemas. The desired PERCEPTORY STEREOTYPES are easily built on-demand from GEOMETRY, TEMPORALITY, and VISUAL INFO characteristics (Figure 3).

Carrying on with our road example, Figure 4 illustrates the use of *Perceptory* stereotypes in classes and attributes to identify the geometry. First, the class *Road* is stereotyped with a 1 dimensional geometry (), which means that all road instances are depicted as line. Secondly, the 1 dimensional geometry stereotype associated with the attribute *NumberOfLanes* means that, from the point of view of the users, this attribute has a spatial (linear) distribution *within* object instances (a given road) without creating new object types (ex. Road segments). Such perception of objects is frequent in applications using linear referencing (ex. Road networks) and should be depicted as such at the conceptual level (although this may lead to creating road segments at the implementation level or to using dynamic segmentation). Very diverse and more complex examples could be given here to illustrate the simplicity and high power of expression of the PVL, but this goes beyond the goal of the present paper and a comprehensive description of the use PVL is provided in [Bédard and Proulx].

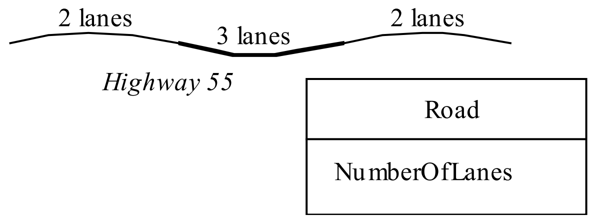


Figure 4: Example of the use of *Perceptory* pictograms

A study of Spatial PVL, ISO 19107 - *Spatial schema* and Topic 1 - *Feature Geometry (OGC)* demonstrates that Perceptory's geospatial repository is aligned with ISO/TC 211 and OCG standards. Figure 5 shows the *Perceptory* geometry metamodel with regards to ISO 19107.

This figure shows that in Perceptory's repository, geometry is a characteristic subtype that is further specialized in simple and complex geometry. PERCEPTORY STEREOTYPE is partly generalized from GEOMETRY.

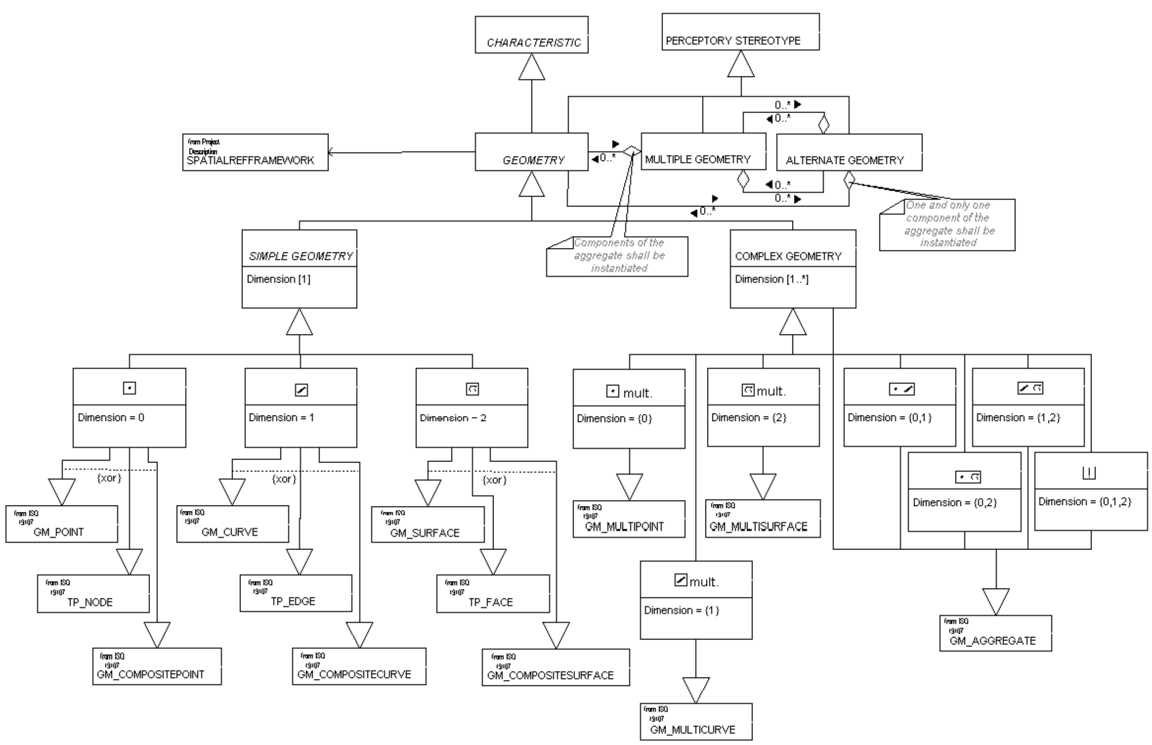


Figure 5: Perceptory geometry metamodel correspondance with ISO 19107

SIMPLE GEOMETRY is a type of geometry that behaves as non-decomposable objects. The perception is that features are depicted by points (), lines, () or areas (). A data type is assigned to the geometry by going deeper into the design process. For this purpose, standard data type are allowed by subtyping ISO 19107 spatial data types with the SIMPLE GEOMETRY class [ISO 19107, OGC(a)]:

: GM_Point, GM_CompositePoint, or TP_Node,
: GM_Curve, GM_CompositeCurve, or TP_Edge,
: GM_Surface, GM_CompositeSurface, or TP_Face.

These data types are detailed in depth in [ISO 19107] and [OGC(a)].

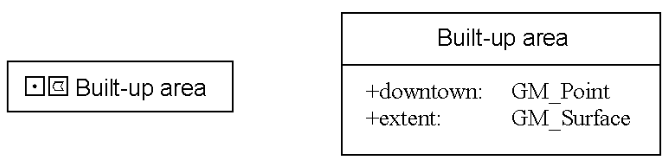
COMPLEX GEOMETRY is a spatial definition with multiple spatial components that, when taken together, constitute the geometry of an object. This type of geometry is used to map objects represented by a spatial aggregation such as *rapids* in a river (aggregation of disjoint lines), or *park* or *city* delineated by an aggregate of areas. Complex geometry can be constructed from 0D, 1D, or 2D primitives exclusively, or combinations of them.

Subtype of ISO 19107 types of aggregate by COMPLEX GEOMETRY is again introduced to support standard data types:

mult. .... GM_MultiPoint
mult. .... GM_MultiCurve
mult. .... GM_MultiSurface
.... GM_Aggregate
.... GM_Aggregate
.... GM_Aggregate
.... GM_Aggregate

Note: *mult.* stands for multiplicity, i.e., minimum and maximum number of spatial components.

MULTIPLE GEOMETRY includes stereotypes comprised of one or several GEOMETRY with or without one or several ALTERNATE GEOMETRY. All components of MULTIPLE GEOMETRY shall be instantiated and all have specific meanings. ISO 19107 has no equivalent data type for multiple geometry; this issue is covered by ISO 19109. An ISO 19109 equivalent representation must introduce as many spatial attributes (i.e., attribute of a spatial type) as pictograms comprising the stereotype. For example, suppose we have a *BUILT-UP AREA* depicted by a point representing its downtown location, and which is also delimited by an area (Figure 6a). An ISO 19109 equivalent representation could have two mandatory spatial attributes, *downtown* and *extent* (Figure 6b).

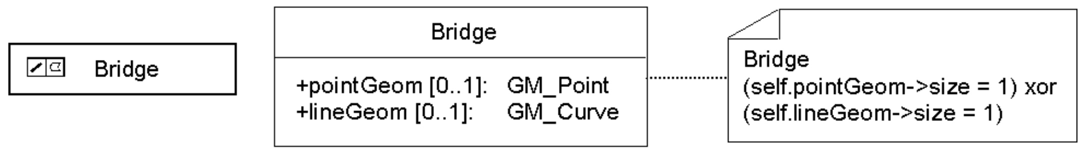


6a) Built-up area as modelled in *Perceptory*

6b) Built-up area as modelled in ISO 19109

Figure 6: Built-up area example

ALTERNATE GEOMETRY includes the stereotypes that use GEOMETRY or MULTIPLE GEOMETRY components. However, objects shall instantiate one and only one geometric component at a time. For example, suppose we have a class *BRIDGE*. Bridge instances under 25 metres in length are shown as points; all others as lines. This means that the class *BRIDGE* needs to support two spatial attributes. However, one bridge instance will never have more than one geometric description. As per multiple geometry, this issue is part of ISO 19109. The class *BRIDGE* introduces two optional spatial attributes with a constraint (Figure 7).



7a) Bridge as modelled in *Perceptory*

7b) Bridge as modelled in ISO 19109

Figure 7: Built-up area example

Temporal definition

The support of temporal definition is approached similarly to that for spatial definition. Figure 8 shows the temporality metamodel of Perceptory's repository. TEMPORALITY is a type of characteristic further specialized with punctual (◷) and durable (◕) classes. An accident occurring at 11:00 a.m. is an example of punctual time. A house erected on May 5, 1968 and demolished on April 17, 1989 is an example of durable time. Punctual (◷) and durable (◕) classes inherit standard data types from ISO 19108. From a geometric temporal standpoint, 0D temporal objects (◷) inherit from TM_Instant and 1D temporal objects (◕) from TM_Period (Figure 8). From the topologic standpoint, 0D temporal objects (◷) inherit from TM_Node and 1D temporal objects (◕) from TM_Edge. Consequently, *Perceptory* temporal characteristics are aligned with ISO 19108.

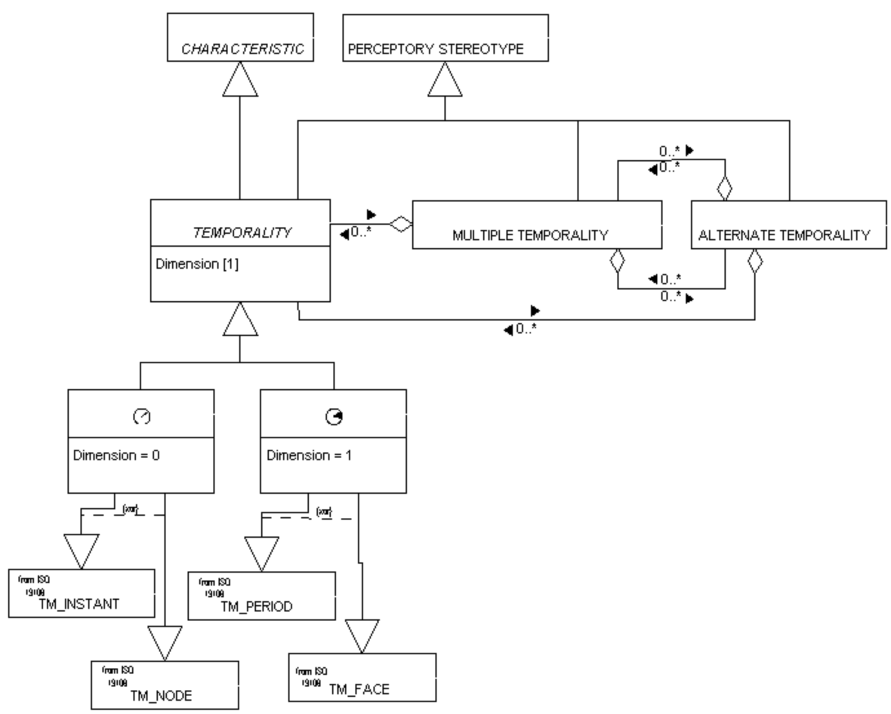


Figure 8: Perceptory's repository temporality metamodel correspondance with ISO 19108

These temporal characteristics, similarly to simple, also serve as *Perceptory* stereotypes. This is possible through the Generalization relationships between TEMPORALITY and PERCEPTORY STEREOTYPE. MULTIPLE- and ALTERNATE TEMPORALITY are respectively replicates of MULTIPLE-, and ALTERNATE GEOMETRY (see description thereof).

Spatiotemporal definition

Spatiotemporal definition is an issue not yet covered by ISO/TC 211. A new work item will be introduced to define a spatiotemporal schema. However, *Perceptory* introduces the concept of spatial evolution in its metamodel (Figure 3). Spatial evolution refers to the simultaneous description of objects in geometric and temporal space. In other words, the spatial depiction supports a time component. For example, the path of a car through time and space requires that the position and time of each vertex be recorded. The *supports* relationship between GEOMETRY and TEMPORALITY classes expresses the capability of geometry to have a temporal depiction.

5. A practical example

This section outlines an example that illustrates the various concepts presented above and sustained by *Perceptory*. The example has been extracted from the National Topographic Data Base (NTDB). The NTDB is a comprehensive topographic database covering the Canadian landmass at the 1:50 000 and 1:250 000 scales. The Centre for Topographic Information in Sherbrooke (Geomatics Canada) is now using *Perceptory* to model the next version of the NTDB and Perceptory's geospatial repository to store detailed information.

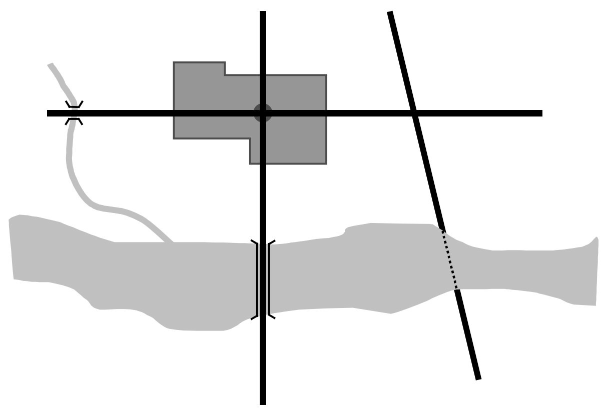


Figure 9: Sample of map features

Figure 9 shows a sample of map features that can be found in NTDB. This example shows the features road, bridge, ferry route, built-up area, and river, represented in conceptual schemas shown in Figure 10. All features are introduced as classes with corresponding spatial stereotypes (Figure 10a) or attributes (Figure 10b). From Figure 10a, we intuitively recognize that the ROAD and FERRY ROUTE features are depicted with line geometry. BRIDGE uses an alternate point/line geometry. BUILT-UP AREA is depicted with multiple geometry: extent is delineated by an area; the downtown position with a point. The RIVER complex is shown using an aggregate of lines and areas.

ROAD is characterized by three descriptive attributes. *Classification* is a descriptive attribute that has an enumerated domain of value. *RoadNumber* is an optional attribute constrained by a range domain of values. The attribute *NumberOfLanes* is mapped using line geometry. This means that each portion of road with a specific number of lanes is depicted separately (whatever the mode of implementation: dynamic segmentation or a priori segmentation with an implementation-level class Road Segment).

10a) Conceptual schema with Spatial PVL

10b) Conceptual schema ISO/TC 211

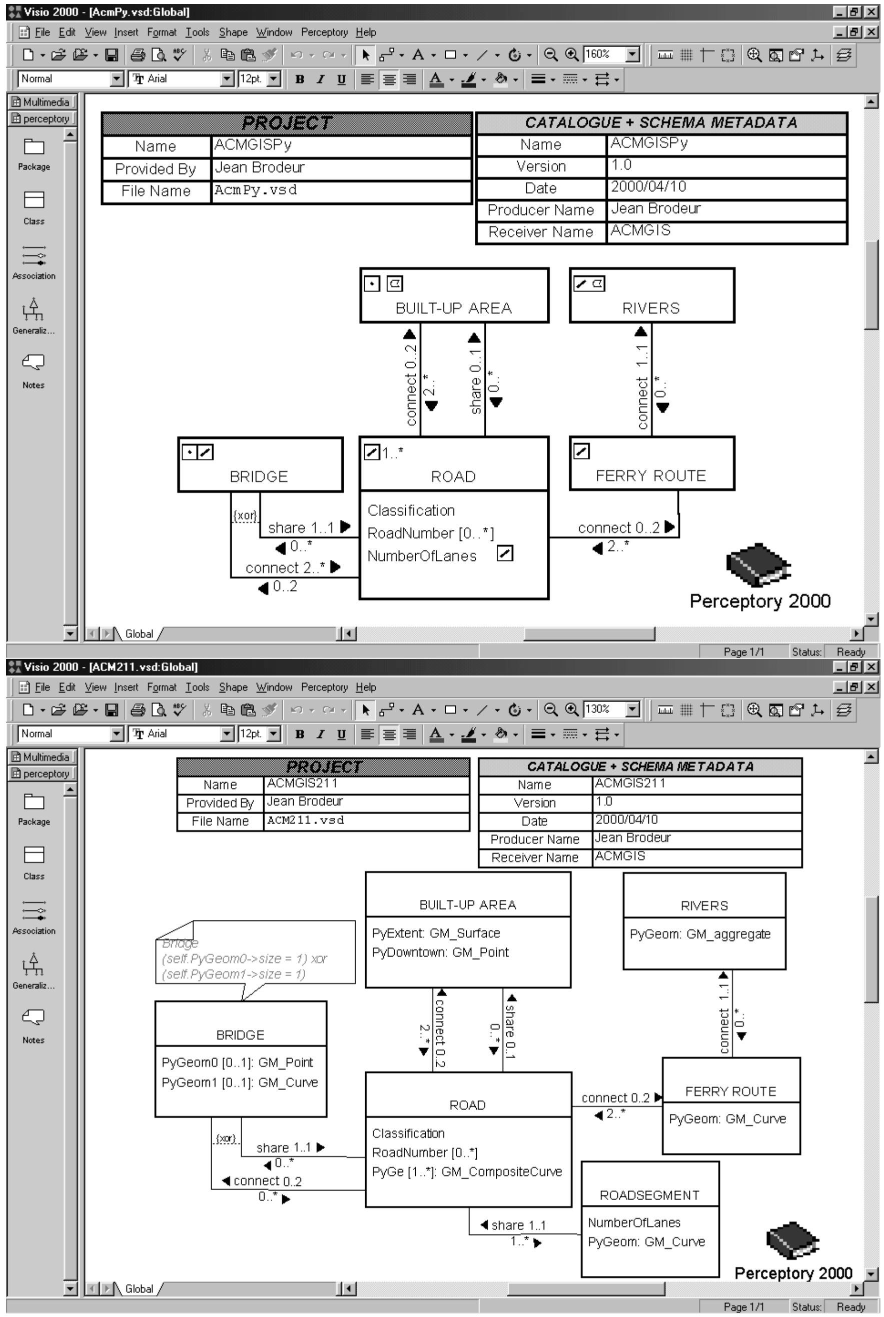


Figure 10: Conceptual schemas with PVL and ISO/TC 211

This model, however, can be translated into an application schema according to ISO/TC 211 rules. Figure 10b shows all the classes in Figure 10a, plus ROADSEGMENT. All pictograms have been translated into spatial attributes with their proper data types. ROAD has an attribute *PyGe* of GM_CompositeCurve sharing geometry with ROADSEGMENT. ROADSEGMENT is introduced for the geometric depiction of the *NumberOfLanes* attribute. It has an attribute *PyGeom* of type GM_Curve. BRIDGE has two optional spatial attributes, *PyGeom0* and *PyGeom1* with a constraint imposing that only one will be instantiated. FERRY ROUTE has one spatial attribute *PyGeom* of the type GM_Curve. BUILT-UP AREA has two mandatory attributes (*PyExtent* and *PyDowntown*) mapping the multiple attribute stereotype in Figure 10a.

These two schemas are equivalent. *Perceptory* uses an intuitive approach with spatial stereotypes. A *Perceptory*-like schema is suited for high levels of abstraction and for simplicity of building/reading. ISO/TC 211 uses a detailed structure. Although still conceptual, ISO/TC 211-like schemas are, however, a step further in the analysis process and address issues of less immediate interest to the final users. Details from these two conceptual schemas must be stored in the geospatial repository for greater database comprehension.

Figure 11 shows a sample of *Perceptory*'s repository interfaces used to store the details of feature classes. At the top level, the class interface gives the semantics of the feature class with name, definition, and all other information shown in the metamodel. This view shows the ROAD class from our preceding example. This interface is composed of tabs for accessing all related characteristics and operations. The attribute tab (second from right) gives access to descriptive attributes. Selecting Attribute brings up the attribute interface, from which all characteristics from semantics to spatial evolution can be accessed via the tabs. The second interface in Figure 11 illustrates the attribute *Classification* and its semantics. If an attribute domain has been created, selecting the domain tab will reveal attribute values from its proper domain. The third interface shows the attribute value *Highway* with its semantics as part of an enumerated domain of values.

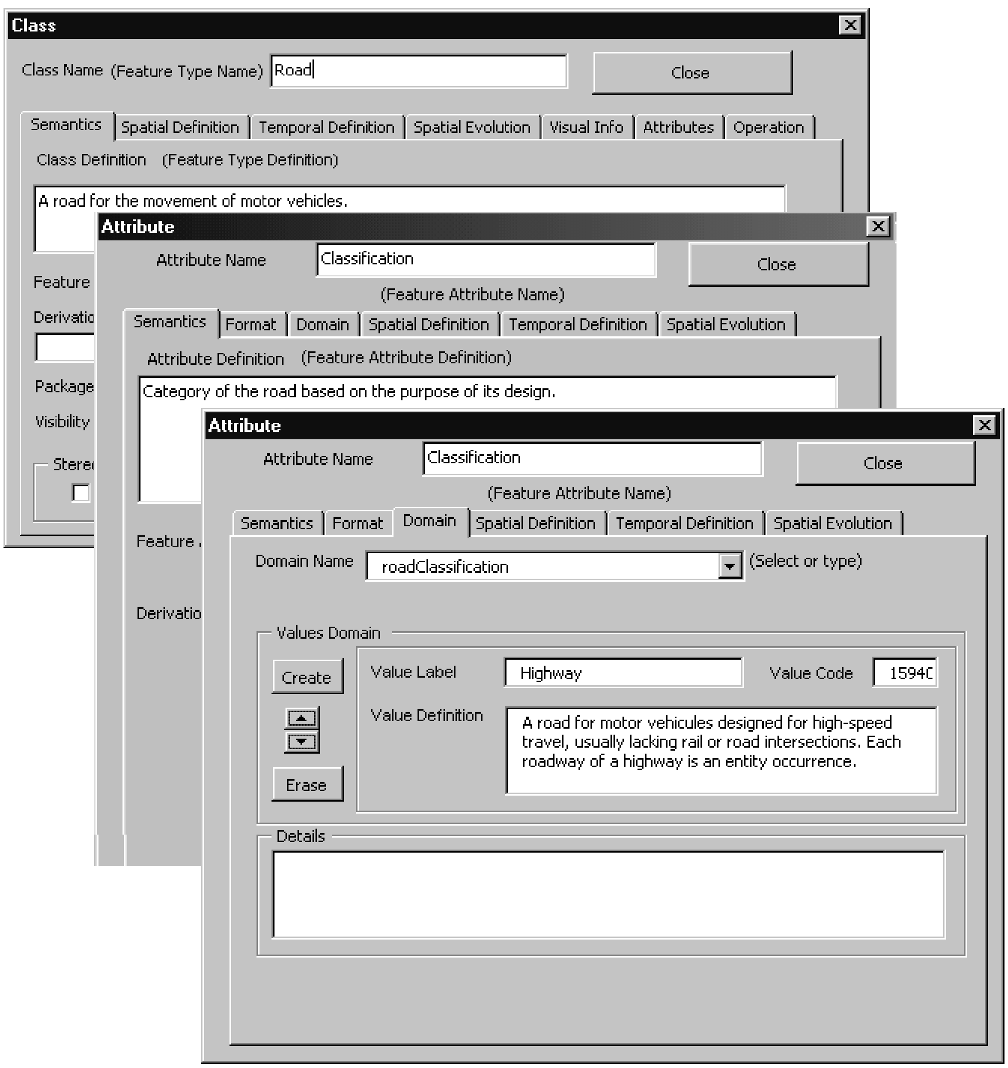


Figure 11: Sample of data repository interfaces

6. Conclusion and future work

This research focuses on the importance of documenting geospatial database content. A database without a rich description of its contents is useless to other people and organizations. This type of documentation can be provided with geospatial repositories. Conceptual schemas and data dictionaries are key components of geospatial data repositories. Documentation of geospatial data must be aligned with international standards in geomatics to ensure interoperability.

A UML object class diagram extended with PVL-based stereotypes conveys the appropriate knowledge of geospatial and temporal database structure. PVL and ISO/TC 211-like conceptual schemas are good complements. Alignment of the two types of schemas has been demonstrated using inheritance of ISO 19107 and ISO 19108 data types; the mapping of complex and multiple geometry with spatial attributes.

The contents of a geospatial database can be described by implementing a data dictionary. ISO 19110 provides a methodology for documenting geospatial features. By introducing operations, *Perceptory* model elements behave as ISO 19110 data types in supporting this standard. It is our experience that the Perceptory freeware provides a good environment to build a geospatial repository aligned with international standards.

Several research issues need further consideration. Multimedia information stereotypes need to be aligned with ongoing work on coverage through OGC, ISO/TC 211, and ISO SQL3/MM. Access to geospatial repositories via the Internet is a major concern, while translating data dictionaries into XML will be investigated. Defining a view manager dealing with subset representation of conceptual schemas is also of some interest.

Up to now, we have concentrated our work on the UML static diagram. Integration of other types of models, such as Use Case, Collaboration, and Sequence diagrams, need to be investigated in a geospatial context.

The authors wish to acknowledge the support of the Natural Sciences and Engineering Research Council of Canada and of Geomatics Canada (Centre for Topographic Information in Sherbrooke) for this project as well as the contribution of Théophane Langlais from Université de LaRochelle, France.

References

Bédard(a), Yvan, 1999. Principles of Spatial Database Analysis and Design, Geographical Information Systems, vol. 1., 2nd edition, ed. P.A. Longley, M.F. Goodchild, D.J. Maguire, and D.W. Rhind, pp. 413-424.

Bédard(b), Yvan, 1999. Visual Modelling of Spatial Database Towards Spatial PVL and UML, Geomatica, Vol. 53, No. 2, pp. 169-185.

FGDC, 1998. Content Standard for Digital Geospatial Metadata, FGDC-STD-001-1998, http://www.fgdc.gov/metadata/contstan.html .

Griffin, Jane, 1995. Repositories: Data Dictionary Descendant Can Extend Legacy Code Investment. Application Development Trends.

Hart, David and Hugh Phillips, 1998. Metadata Primer -- A "How To" Guide on Metadata Implementation, NSGIC, 1998, http://www.lic.wisc.edu/metadata/metaprim.htm .

Hohnmann, L., 1997. Journey of the Software Professional, Prentice Hall PTR, ISBN 0-13-237013-4.

Inmon, W. H., 1993. Developing Client/Server Applications. John Wiley & Sons Inc.

ISO 19103, 1999. CD 19103 (formerly 15046-3) Geographic Information – Conceptual schema language, 75 p.

ISO 19107, 1999. 2. CD 19107 Geographic Information – Spatial Schema, 143 p.

ISO 19108, 1999. CD 19108 Geographic Information – Temporal Schema, 50 p.

ISO 19109, 1998. CD 19109 Geographic Information – Rules for Application Schema, 71 p.

ISO 191010, 1998. 2. CD 19110 Geographic Information – Feature Cataloguing Methodology, 51 p.

Jones, Mark, 1991. Brave New World: A Vision of IRDS, Database Programming and Design.

META Group, 1996. Briefing Paper: What is Metadata, Data Warehouse Tools Bulletin. http://www.computerwire.com/bulletinsuk/212e_1a6.htm .

Moriarty, Terry, 1990. Are You Ready for a Repository?, Database Programming and Design, vol. 3, no. 3.

OCL, 1997. Object Constraint Language Specification, Version 1.1, 32 p.

OGC(a), 1999. The OpenGIS™ Abstract Specification - Topic 1: Feature Geometry, Version 4, Open GIS Consortium, 85 p.

OGC(b), 1999. The OpenGIS™ Abstract Specification - Topic 5: Features, Version 4, Open GIS Consortium, 45 p.

OGC(c), 1999. The OpenGIS™ Abstract Specification - Topic 8: Relationships between Features, version 4, Open GIS Consortium, 17 p.

OGC(d), 1999. The OpenGIS™ Abstract Specification - Topic 11: Metadata, Version 4, Open GIS Consortium, 12 p.

OMG(a), 1997. UML Notation Guide, Version 1.1, 141 p., http://www.rational.com/uml/

OMG(b), 1997. UML Semantics, Version 1.1, 162 p., http://www.rational.com/uml/

OMG(c), 2000. UML Resource Page, http://www.omg.org/uml/

Prabandham M. and al., 1990. The Role of IRDS, Database Programming and Design.

Bédard,Y., M.J. Proulx, 2000. Perceptory Web site, http://sirs.scg.ulaval.ca/Perceptory/

Rumbaugh, J., I. Jacobson and G. Booch, 1999. The Unified Modeling Language: Reference Manual, Addison-Wesley, 550 p.