SPATIAL DATABASE MODELING PICTOGRAMMIC LANGUAGES

Yvan Bédard, Suzie Larrivée
Dept of Geomatics Sciences
Centre for Research in Geomatics
Canada NSERC Industrial Research Chair in Geospatial Databases for Decision Support
Laval University
Quebec City, Canada
Telephone: 1-418-656-2131
Fax: 1-418-656-7411
{suzie.larrivee, yvan.bedard}@scg.ulaval.ca

SYNONYMS

Spatial modeling language extension, Spatio-temporal modeling language extension, Perceptory extensions.

DEFINITION

"Spatial databases" consist of large groups of data structured in a way to represent the geographic features of interest to the users of a system. Spatial database models are schematic representations of these data. Database models are created to design and document the system, to facilitate communication and to support programming. They are created using CASE tools (computer-assisted software engineering). CASE tools support schema drawing, dictionaries and code generation. Database schemas are typically represented with a graphical language such as UML (Unified Modeling Language; http://www.uml.org).

"Database models" can represent (1) users' real-life views of the data of interest, (2) developers' views of the potential organization of these data for a family of technologies, or (3) their final implementation on a specific platform. For example, in the standard Model-Driven Architecture (MDA) method (http://www.omg.org/mda/), these three levels of models are respectively called CIM (computation-independent model), PIM (platform-independent model) and PSM (platform-specific model). In other methods, they may be called conceptual, logical and physical models as well as analysis, design and implementation models.

"Pictograms" are symbols aimed at facilitating modeling. Different sets of pictograms have been proposed. This chapter presents those used by the CASE tool Perceptory (http://sirs.scg.ulaval.ca/perceptory) since they are the most widely used, they were designed to allow developers to keep using their method, and they were thoroughly tested as implementations of UML stereotypes. In Perceptory, they aim at hiding the complexity of geometric primitives in CIM and PIM models. They can serve other purposes as well and have been implemented in other CASE tools [14].

HISTORICAL BACKGROUND

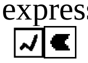
In the field of GIS, pictograms were first proposed in 1989 by Bedard and Paquette [6] to simplify how Entity-Relationship (E/R) models depicted the geometry of cartographic features. It was then called "Sub-Model Substitution" technique as the main goal was to remove from the spatial database model those geometric primitives with their data elements and relationship (considered of no interest to the user) and to replace them by simple symbols showing only the information of interest to the users (i.e. the features' shape). This first solution was tested in several projects and enhanced over time to lead to the development of Modul-R [4,5], the first spatio-temporally extended E/R which led to Orion, the first GIS-compatible CASE tool in 1992 [5]. This first solution has influenced several researchers afterwards. Examples of methods or tools using pictograms for spatial databases include Perceptory [1,3] which is used in over 30 countries, Software Development Process Model with Objecteering [see A, 14], MADS [see B, 17], CONGOO [16], UML-Geoframe with ArgoCASEGEO [10], and STER [see C, 21].

In 1996, Modul-R pictograms were revisited to integrate three paradigms: object-orientation (OO), plug-in (module, blade, cartridge) and a pragmatic symbiotic approach [3]. Object-orientation allowed for more expressive power and was first tested with UML in its pre-release days. The plug-in approach led to define the pictograms and their syntax as a module, i.e. a specialized language designed to extend standard languages (e.g. UML, E/R, English). This allowed for enriching one's modeling language and tool rather than requiring to adopt new ones. For instance, in addition to Perceptory, these pictograms have been used with commercial and open-source CASE tools such as Oracle Designer, Objecteering and others while being also used to describe spatial integrity constraints, to compare database semantics and as commercial user-interface components. With regards to the symbiotic approach, it came from cognitive studies and pragmatics lessons resulting from several projects with practitioners, including very complex ones. It helped to find a better balance between human abilities, language requirements, database design methods and commercial software constraints. Practical projects clearly indicated the need to better support unexpected complex situations, to simplify the pictograms along with their syntax, and to better balance the content of the graphical schema with the ontological content of the dictionary (i.e. simpler schemas, increased use of natural and formal languages in the dictionary). This was a departure from the trend of that period to rely increasingly on graphical depictions. Such novel approach and the arrival of UML led to developing Perceptory. This approach also goes beyond the leading tendency to perceive "modeling" solely as a schema-building exercise while in fact it is not; a schema without clear and complete semantics is meaningless and its robustness cannot be validated. Accordingly, good spatial database modeling becomes an ontological exercise. For example, Perceptory provides specialized spatial and temporal sections in its dictionary (as can be added to other CASE tools). In the remaining of this chapter, we present the scientific fundamentals of modeling spatial databases with pictograms, using examples from the UML-based Perceptory CASE tool.

SCIENTIFIC FUNDAMENTALS

"Pictograms" aim at supporting the expression of any feature's spatial and spatio-temporal properties into a consistent manner that is compatible with various human-oriented languages (ex. UML, Entity-Relationship, English, French).

"Syntax rules" dictate the way to combine and position pictograms in a model or document. These rules also dictate how to use special characters (0-9 N ,). Properly combining pictograms, with or without characters, makes it possible to express complex cases of geometry and spatio-temporality, namely: facultative, mandatory, alternate, aggregate, multiple, and derived.

A “pictogrammic expression” includes one or several pictograms which are positioned in a precise manner with pertinent digits according to a syntax. Such a pictogrammic expression completely describes the spatial, temporal or spatio-temporal properties of either (1) a feature, (2) where and when an attribute value is valid within an object geometry or existence, or of (3) a relationship between features. For example, in Perceptory, the simple expression  is made of only one pictogram and represents a simple 1D geometry in a 2D universe. Similarly, the expression  represents the same geometry in a 3D universe while the expression  adds thickness to this geometry. On the other hand, the expression  has a different meaning from the previous ones and from the expression  or from the expression  0,N. In a similar manner, the simple expression  represents one instant, the expression  represents one period of time and more complex temporal and spatio-temporal expressions can be made.

Grouping pictograms and syntactic rules commonly used together allows one to form a specialized graphical language called “PVL” (Plug-in for Visual Languages). A PVL, as introduced in [3], allows extending a modeling language with a tested method that is compatible with other PVLs of the same family if needed. For example, one may decide to use only a small group of Perceptory pictograms to make a 2D spatial PVL (i.e. a language to depict plane geometries of geographic features) while later on, if needed, use a larger group that make a 3D spatio-temporal PVL. A pictogrammic expression is sometimes called a PVL expression.

The pictograms high level of abstraction facilitates the making of database models, reports, specifications, spatio-temporal integrity constraints, user interfaces, and similar tasks of a system development workflow. They hide the complexity inherent to the description of geometric and temporal primitives and relationships as well as implementation and standard-related issues. In particular, they facilitate the building, editing, communication and validation of spatio-temporal database models as well as their translation into efficient data structures. In spite of such translation rules, the PVL are independent from commercial software and numerous standards.

The pictograms were first created for spatial database modeling and are best described in such a context. Accordingly, the present chapter describes the pictograms implemented as UML stereotypes in Perceptory object class model. In such a context, the PVL allow the analyst or designer to describe the spatial and temporal properties of the elements depicted in an object class schema. Perceptory pictograms support 0D, 1D, 2D and 3D geometries for objects located in 2D or 3D universes (see table 1). Supported temporalities are 0D (instant) and 1D (period) (see table 2). Supported combinations are simple, complex (aggregate), alternate (exclusive OR), multiple (AND), spatio-temporal and hybrid (combinations of any of the above) (see tables 3, 4 and 5). Supported minimum multiplicities include facultative (0), mandatory (1), specific number and many (N) while maximum multiplicities include the three latter. Special cases are "any possibility”, "not yet defined" and "complicated", the latter pointing to a textual description in the repository (when easier to read). All geometries and temporalities can be indicated as measured or as derived from other attributes, objects, relationship using calculations, spatial or temporal analysis. No geometry or temporality is also accepted. Pictogrammic expressions may describe object classes, association classes, attributes and may be used within operations.

	2D space	3D space	Examples of cases
0D geometry			hydrants when they are all represented by points
1D geometry			road segments when they are all represented by lines
			electric poles when they are all represented by vertical lines
2D geometry			lakes when they are all represented by polygons
			walls when they are all represented by vertical plans
3D geometry			buildings when they are all represented by solids

Table 1: simple pictogrammic expressions for geometry.

		Examples of cases
0D temporality		Existence of accidents; traffic flow of a road segment
1D temporality		Existence of a building; duration of its commercial use; duration of its ownership by a given person

Table 2: simple pictogrammic expressions for temporality.

Geometry	Examples of syntax	Examples of cases
Aggregate geometry (complex)		Hydrographic networks composed of 1D rivers and 2D lakes (i.e. aggregate of different geometries)
Aggregate geometry (simple)	1,N	Some municipalities may include several 2D geometries such as islands (i.e. aggregate of similar geometries)
Alternate geometry (on same line)	 	Buildings having a 0D shape if area < 1 hectare OR a 2D shape if area > 1 hectare (Exclusive OR)
Facultative geometry	0,1	Buildings in database may have no geometry if area < 0.2 hectare, or a 0D shape if area > 0.2 hectare
Multiple geometry (on different lines)	 	Every municipality has a 2D shape AND a OD location (ex. downtown)
N.B. same syntax for 2D and 3D pictograms		

Table 3: syntax for advanced 2D and 3D spatial pictogrammic expressions.

Temporality	Examples of syntax	Examples of cases for feature existence and states
Alternate temporality (on same line)		Forest fires lasting several days OR 1 day (if temporal resolution is 1 day); water level data varying continuously when opening/closing the dam OR remaining stable for a period once a level is reached
Facultative temporality	0,1	Houses in database may need NO construction and demolition dates IF area < 0.2 hectare
Multiple temporality (on different lines)		Hurricane existence defined by a date of beginning and a duration for some purposes, AND by a unique date of maximum peek for other purposes. Buildings commercial value considered stable for the whole year for tax purposes but as being valid only the day when the building was assessed for market analysis purposes.
Spatio-temporality		Position of a moving vehicle. The temporal pictogram affects the spatial pictogram on its left
N.B. Selecting between or depends on the temporal granularity defined into the repository for each class, attribute and geometry.		

Table 4: syntax for advanced temporal and spatio-temporal pictogrammic expressions.

Derived geometry or temporality "italic pictogram"		Municipality centroids derived from their polygons; 3D buildings derived from 2D buildings with number of floors; duration of commercial use derived from permits
Hybrid expression (combination of any pictos above)	1,N	A set of individual cyclists continuously moving during a race or forming a group that changes its size during the race
Default multiplicity	If no multiplicity is written immediately after a pictogram, the 1,1 multiplicity is implied	
Any possibility		"wildcard pictogram" meaning no predefined shape or temporality, and no restriction on the geometry or temporality
Complicated		Better explained textually in the dictionary than using a complicated PVL expression in a schema. Replaces a long hybrid expression if desired.
Not yet defined		During the process of designing a database, we may know we'll need a geometry or temporality, but not which one (this will be replaced by regular pictograms)

Table 5: syntax and pictograms for special cases

Examples of the use of pictogrammic expressions for UML object classes are presented hereafter. Figure 1 describes an accident as a an instantaneous event located positioned as a point. Figure 2 shows a case where users want to keep information about the existence of commercial buildings (construction and destruction), about the evolution of their commercial value (values, period of validity) and of its polygonal representation if it is enlarged or modified. Figure 3 illustrates a case of aggregated complex geometry while figure 4 shows cases of simple and alternate geometries. At last, figure 5 shows a case of multiple geometry where the first pictogram expresses the fact that every building is represented by simple polygon at large scales and the second line of pictograms indicates that some (but not all) buildings may have a second geometry, either a point or a line, depending on their size, for small scale maps (usually to properly place symbolic representations).

id	gravity	nb_death	nb_injuries	nb_vehicles	date	x_geom	y_geom
1	2	0	0	2	07/07/2005	-71,2	46,78
2	2	0	1	3	03/04/2006	-71,35	45,9
3	1	3	2	2	04/05/2006	-71,56	45,64

Figure 1: Example of simple pictogrammic expressions for the geometry and existence of a UML object class Accident.

Figure 2: Example of a spatio-temporal pictogrammic expression, a temporal expression for the existence of the UML object class and of another one to keep track of the evolution of one attribute.

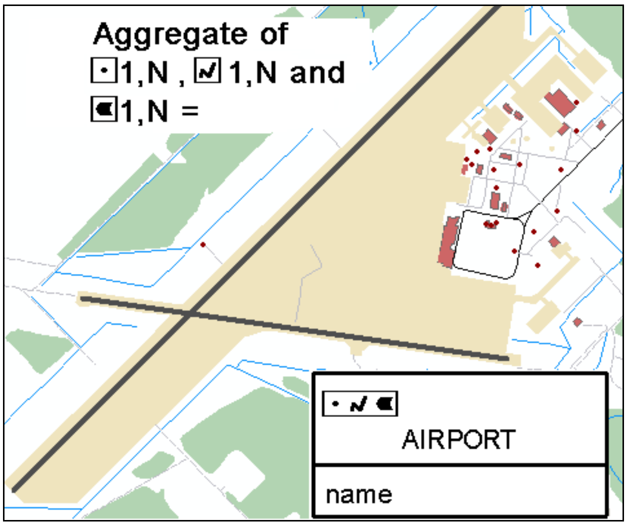


Figure 3: Example of a complex aggregate geometry for the Airport object class, that is an aggregate of points, lines and polygons (Data from ministère des Ressources naturelles et de la faune du Québec)

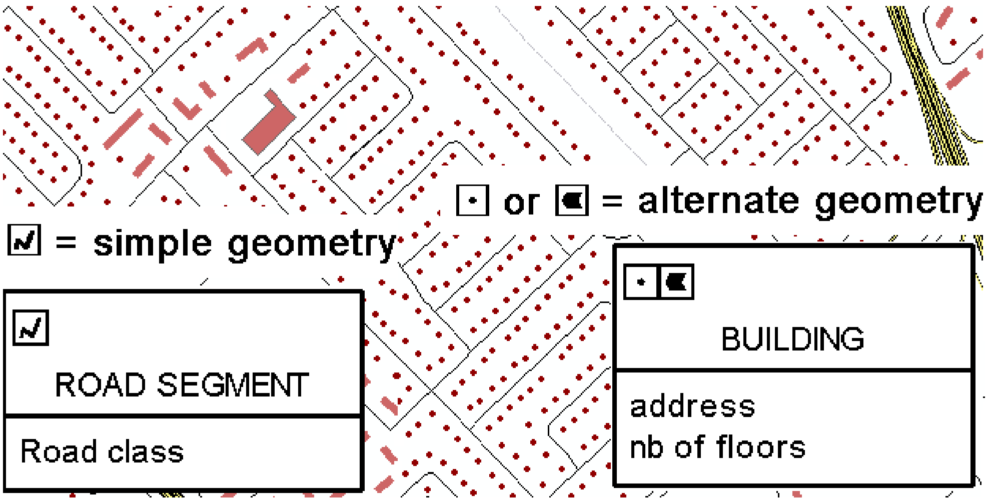
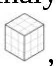
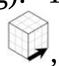


Figure 4: Example of a simple geometry pictogrammic expression (where each instance is represented by one line) and of an alternate geometry (where small buildings are represented by a point and large ones by a polygon) (Data from ministère des Ressources naturelles et de la faune du Québec)

From a UML point of view, these pictogrammic expressions are implemented as stereotypes (a formal way of extending UML) and are built on-the-fly in Perceptory. Using such pictogrammic expressions has also proved to be useful to model spatial multidimensional databases (or datacubes) as used in spatial data warehousing and SOLAP (Spatial On-Line Analytical Processing). These datacubes pictogrammic expressions include datacube , data dimension , member , measure  and are compatible with the previous spatial and temporal pictograms. They are all supported by Perceptory.

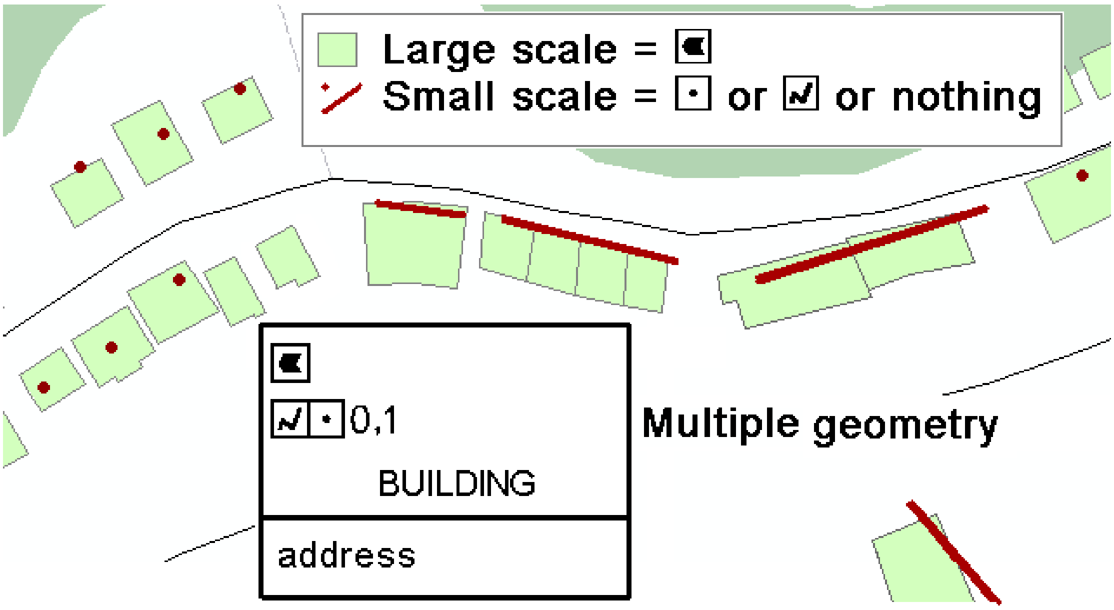


Figure 5: Example of a multiple geometry pictogrammic expression where, at large scale (ex. 1:1000), buildings are represented by a polygon and at small scale (ex. 1:20 000), they are represented by a point, a line or nothing (Data from Research and Development Defence Canada and from ministère des Ressources naturelles et de la faune du Québec)

KEY APPLICATIONS

Pictogrammic languages, if sufficiently expressive and usable, can serve several purposes. The following paragraphs further describe the key application, i.e. spatial database modeling, and other applications of interest.

Using pictogrammic languages for spatio-temporal database modeling

Modeling databases for GIS applications has always posed several challenges for system analysts, system developers as well as for their clients whose involvement into the development of such a project is not a familiar endeavor. Used with well-known modeling techniques, pictogrammic expressions help to meet these challenges [1, 17, 16, 10, 21] and are commonly used in different methods, for example in relational database design (ex. UML relational stereotypes [15]). Extending CASE tools and modeling methods in such a way allows analysts and designers to work at a higher level of abstraction for the first steps of a spatial database project. As presented in figure 6, once high-level models completed, they can be translated into more technical models which are closer to implementation such as presented in Chapter D. Such multi-level approach is typical of good software engineering methods, for example:

- The Object Management Group (OMG) Model-Driven Architecture (MDA) having three levels of models: Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM);
- Zachman Framework having business or enterprise model, system model and technology model (also called semantic, logical and physical models);

- Rational Unified Process (RUP) having domain model, analysis model, design model and implementation model.

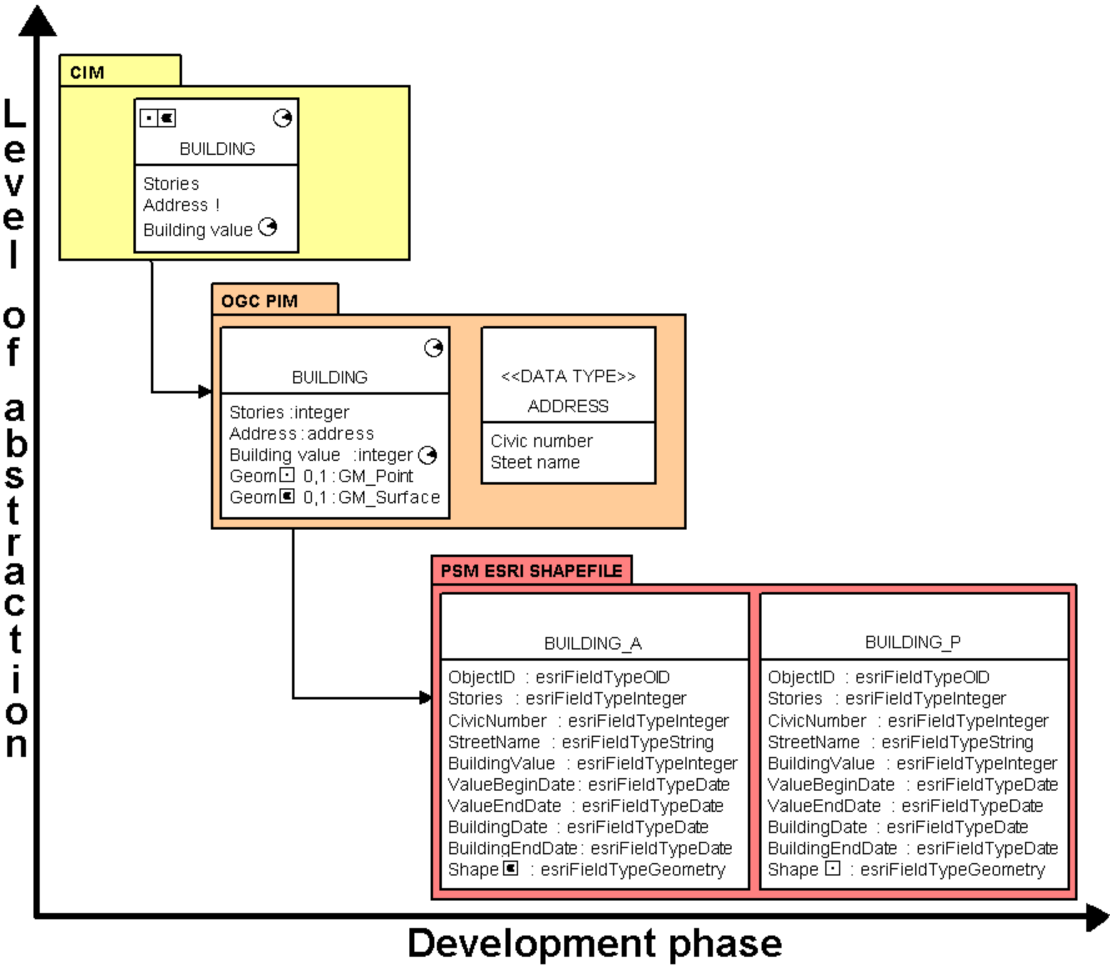


Figure 6: examples of CIM, PIM and PSM levels of abstraction of the MDA method where information encapsulated in the higher levels using pictograms are expanded in lower levels.

Since the pictograms are aimed at facilitating modeling by being closer to human language than typical modeling artefacts, they are primarily used in high-level models. Regarding the MDA method, pictogrammic expressions are more widely used for CIM than for PIM and PSM:

- CIM: “A *computation independent model* is a view of a system from the computation independent viewpoint. A CIM does not show details of the structure of systems. A CIM is sometimes called a domain model and a vocabulary that is familiar to the practitioners of the domain in question is used in its specification.”[13]
- PIM: “A *platform independent model* is a view of a system from the platform independent viewpoint. A PIM exhibits a specified degree of platform independence so as to be suitable for use with a number of different platforms of similar type.” [13]
- PSM: “A *platform specific model* is a view of a system from the platform specific viewpoint. A PSM combines the specifications in the PIM with the details that specify how that system uses a particular type of platform.” [13]

Furthermore, since pictograms are not tied to a specific natural language, they facilitate the translation of database models. For example, in Canada, several schemas and repositories are available in English and French. Figure 7 shows such French and English schemas that are synchronized thru the same repository and pictograms. The use of formal ISO-19110 labels (in blue) further facilitates communication while the use of pictograms facilitates automatic GIS code generation and bilingual reporting.

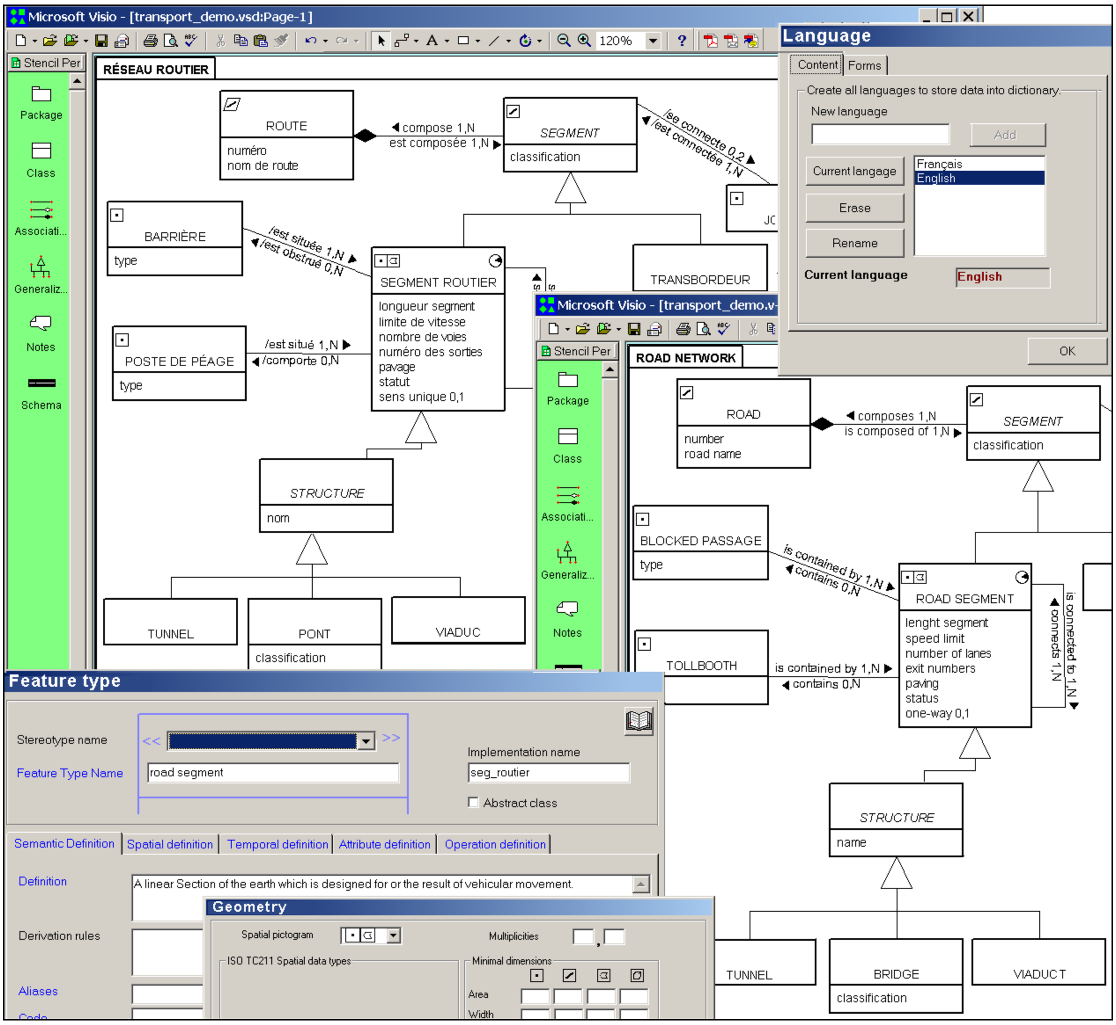


Figure 7: Example of common pictograms in a French and an English CIM synchronized for a same spatio-temporal database using Perceptory multi-standard and multi-language capabilities.

At the CIM level, pictogrammic expressions are intuitive and independent of domain ontologies and technology-oriented standards. No technology artefacts nor standardization elements must appear unless they are useful and intuitive. When the CIM is well defined, it can be translated and enriched to produce lower-level models semi-automatically. Then, technology-oriented artefacts and standard-based elements replace the pictogrammic expressions. For example, in Figure 6, the CIM evolves in a PIM where the geometry is expressed according to ISO/OGC. Then, the PSM shows the structure of two shapefile needed to implement Building Points and Building Areas.

In addition to hiding the technical complexities of GIS and Universal server database engines, using pictogrammic expressions also hides the intricacies of international standards such as ISO/TC-211 and OGC. For example, ISO jargon doesn't express directly all possible geometries (ex. alternate and facultative geometries) and they are not cognitively compatible with clients' conceptual view who assumes a topologically consistent world (ex. GMPoint vs TPNode, GMCurve vs TPEdge, GMSurface vs TPFace, Aggregate vs Multi).

Using pictogrammic expressions to define spatial integrity constraints

Spatial integrity constraints can also be defined efficiently with pictogrammic expressions. For example, in figure 8, the upper window shows a user interface for the definition of spatial integrity constraints between two object classes, with or without considerations to specific attribute values. The lower window shows a report showing the defined spatial integrity constraints. The last window shows an example of using pictogrammic expressions in a 3x3 e-relate matrix.

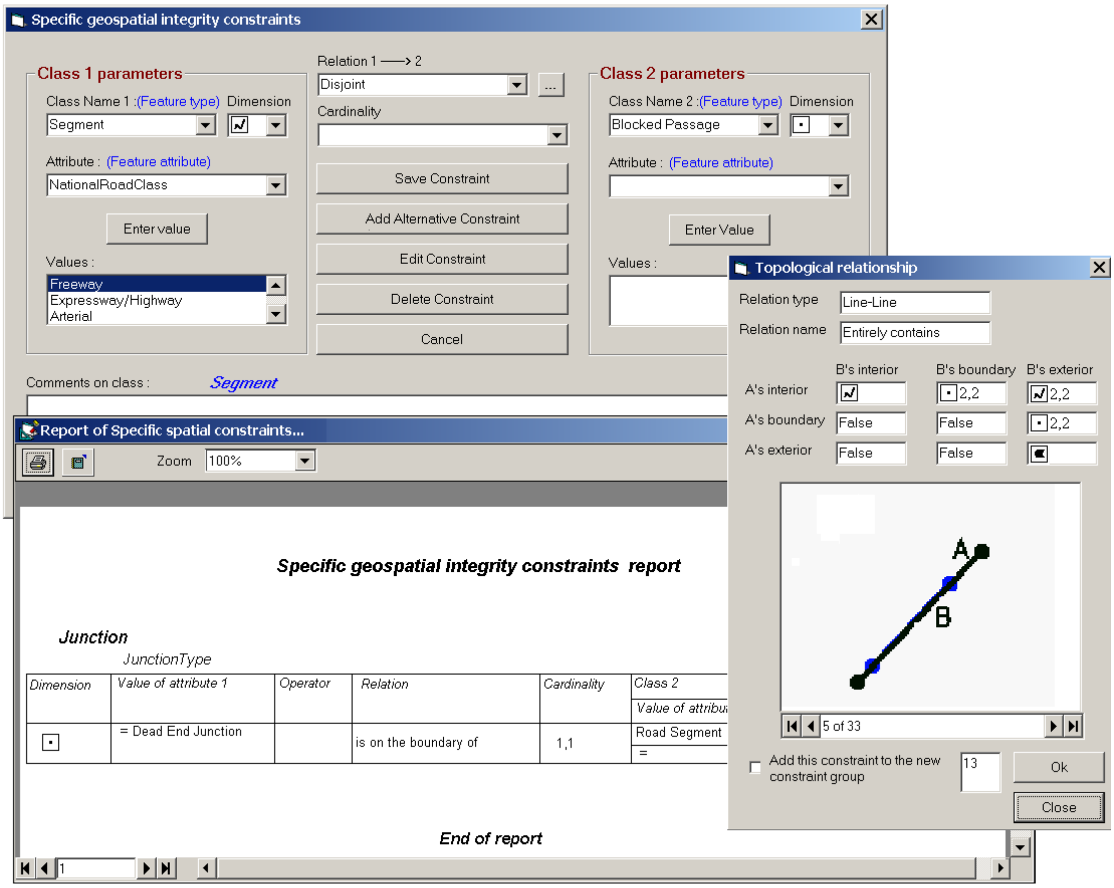


Figure 8: Examples of pictogrammic expressions to define topological constraints between two object classes (upper left), to print them in a report (lower left) and to describe them in an extended ISO e-relate 3X3 matrix.

Additional usages of pictogrammic expressions: software user interfaces, reports and semantic proximity analysis.

Pictogrammic expressions are regularly used in a text to express the spatiality and temporality of concepts. They have been used in reports, data dictionaries and data set specifications. They were also used for semantic proximity analysis [9] and integrated in a commercial package (JMap SOLAP, Figure 9).

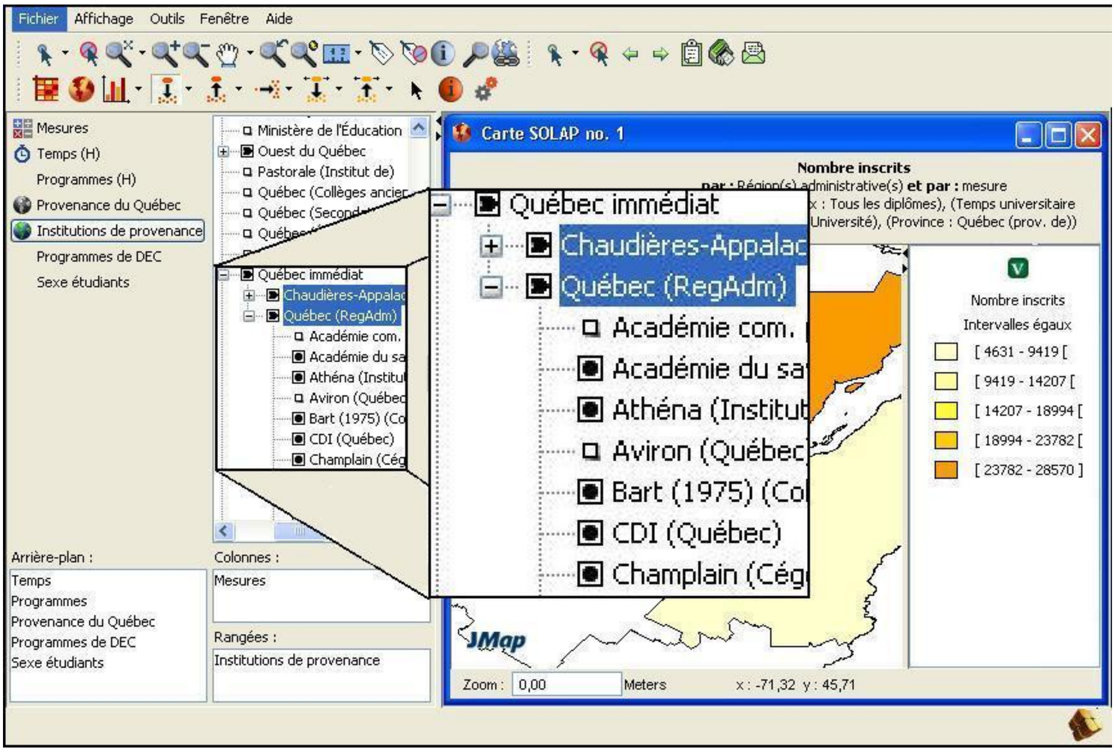


Figure 9: JMap SOLAP interface using pictogrammic expressions.

FUTURE DIRECTIONS

Over the last two decades, different pictogrammic language have emerged to improve the efficiency of systems analysts and to improve the quality of spatial database design. The language presented in this chapter was the first such language and has become the most widely used one, not only within Perceptory but also in other CASE tools and in diverse applications as it is downloadable font (http://sirs.scg.ulaval.ca/YvanBedard/english/others.asp). Such languages will likely evolve in two major directions. First, they should further expand and be tested to accommodate the most recent spatial database trends, that is spatial datacube structures such as those existing in data warehousing and SOLAP applications. Second, as they can be translated into ISO and OGC primitives [8], official adoption of such a language should be put forward to improve interoperability between spatial application database schemas, between ontologies and other documents.

CROSS REFERENCES

A. Spatio-Temporal Model Driven Architecture Enrichment

B. Spatio-Temporal and Multiple-Perception Extensions
C. Spatio-Temporal Database Concepts and Constructs: Towards a Spatio-Temporal Entity Relationship Model
D. Spatio-Temporal Model with ISO 191xx Standards

RECOMMENDED READING

1. Bédard, Y, Larrivée, S., Proulx, M-J., Nadeau, M.: Modeling Geospatial Databases with Plug-Ins for Visual Languages: A Pragmatic Approach and the Impacts of 16 Years of Research and Experimentations on Perceptory. In: S. Wang et al. (Eds.): Conceptual Modeling for Advanced Application Domains. Lecture Notes in Computer Science, Vol. 3289, Springer-Verlag, Berlin Heidelberg New York (2004) 17–30
2. Bédard Y, Proulx MJ, Larrivée S, Bernier E (2002) Modeling Multiple Representation into Spatial Datawarehouses: A UML-based Approach. ISPRS WG IV/3, Ottawa, July 8-12.
3. Bédard Y (1999) Visual Modelling of Spatial Database towards Spatial PVL and UML, Geomatica, 53(2), 169-185
4. Bédard Y, Caron C, Maamar Z, Moulin B, Vallière D (1996) Adapting Data Model for the Design of Spatio-Temporal Database. Comp. Env. and Urban Systems, 20(l) 19-41.
5. Bédard Y, Pageau J, Caron C (1992) Spatial Data Modeling: The Modul-R Formalism and CASE Technology. ISPRS Symposium, Washington, August 1-14
6. Bédard Y, Paquette F (1989) Extending entity/relationship formalism for spatial information systems, AUTO-CARTO 9, April 2-7, Baltimore 818-827
7. Bédard Y, Larrivée S (1992) Développement des systèmes d'information à référence spatiale: vers l'utilisation d'ateliers de génie logiciel. CISM Journal ACSGC, 46(4) 423-433
8. Brodeur, J., Bédard, Y., Proulx, MJ.: Modelling Geospatial Application Database using UML-based Repositories Aligned with International Standards in Geomatics, ACMGIS, November 10-11, Washington DC, (2000) 36-46
9. Brodeur, J., Y. Bédard, G. Edwards & B. Moulin, 2003, Revisiting the Concept of Geospatial Data Interoperability within the Scope of Human Communication Processes, Transactions in GIS, Vol. 7, No. 2, pp. 243-265.
10. Filho, J.L., V.D.F. Sodre, J. Daltio, M.F. Rodrigues Junior, V. Vilela (2004). A CASE Tool for Geographic Database Design Supporting Analysis Patterns. Lecture Notes in Computer Science, Vol. 3289, Springer Berlin/ Heidelberg, pp.43-54.
11. Fowler, M.: UML 2.0, CampusPress (2004) 165 p.
12. Frankel, D. S., Harmon, P., Mukerji, J., Odell, J., Owen, M., Rivitt, P., Rosen, M., Soley, R., M. (2003) The Zachman Framwork and the OMG's Model Driven Architecture. Business Process Trends, Whitepaper.
13. Miller, J. & Mukerji , J. 2003 J. Miller and J. Mukerji, (eds.), MDA Guide Version 1.0, OMG Document: omg/2003-05-01, http://www.omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf, May 2003.
14. Miralles, A. (2006). Ingénierie des modèles pour les applications environnementales. PhD Thesis, Université des Sciences et Techniques du Languedoc, Computer Sciences, 338 p.
15. Naiburg EJ., Maksimchuk, RA.: UML for Database Design, Addison-Wesley (2001) 300 p.

16. Pantazis, D., J.P. Donnay (1996). La conception SIG: méthode et formalisme. Hermès, Paris, 343 p.
17. Parent, C., S. Spaccapietra, E. Zimányi (2006). Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach. Springer, 466 p.
18. Priebe, T., Pernul, G.: Metadaten-gestützer Data-Warehouse-Entwurf mit ADAPTed UML, 5th Int.Tagung Wirtschaftsinformatik (WI 2001), 19.-21. September, Germany (2001)
19. Sowa, J.F Zachman J.A.:.Extending and Formalising the Framework for Information Systems Architecture, IBM Systems Journal, Vol 31, No 3, 1992. IBM Publication G321-5488
20. Thurston, J.: Geo-Visualisation : Current Issues / Future Potentials. GISCafe.com (2001)
21. Tryfona N., R. Price and C. S. Jensen. (2003) Conceptual Models for Spatio-Temporal Applications. Chapter 3 in Book: Spatiotemporal Databases: The CHOROCHRONOS approach, Lecture Notes in Computer Science, Vol. 2520, Springer-Verlag.