How to Enrich the Semantics of Geospatial Databases by Properly Expressing 3D Objects in a Conceptual Model

Suzie Larrivée[1,2,3], Yvan Bédard[1,2,3], and Jacynthe Pouliot[1,2]

[1] Dept of Geomatics Sciences
[2] Centre for Research in Geomatics
[3] Canada NSERC Industrial 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, jacynthe.pouliot}@scg.ulaval.ca

Abstract. Geospatial conceptual data models represent semantic information about the real world that will be implemented in a spatial database. When linked to a repository, they offer a rich basis for formal ontologies. Several spatial extensions [5, 15, 17] have been proposed to data models and repositories in order to enrich the semantics of spatial objects, typically by specifying the geometry of objects in the schema and sometimes by adding geometric details in the repository. Considering the success of such 2D spatial extensions as well as the increased demand for 3D objects management, we defined a 3D spatial extension based on the concept of PVL already used in Perceptory and elsewhere. This paper presents 3D concepts and 3D PVL to help defining the geometry of 3D objects in conceptual data models and repositories. Their originality stems from the fact that no similar solution exists yet for real-life projects. The enrichment of the meaning of 3D objects geometries is discussed as well as its impact on costs, delays and acquisition specifications.

1 Introduction

At the beginning of 70's, the Entity-Relationship Model has emerged to represent semantic information about the real world that implementation model cannot do [7, 8, 9]. Based on this conceptual formalism, many researchers [5, 15, 17] have worked to extend E/R with pictograms to represent and enrich the semantics of 2D spatial objects. Such conceptual data models, with their repository (or dictionary), have been used in several projects and proved useful to describe the semantics of spatial objects stored in spatial databases, including their geometric characteristics. In the most recent solutions, such extensions are used with UML (Unified Modeling Language).

Increasingly, 3D objects management is becoming a common requirement in spatial database systems [1, 6]. Nevertheless, 3D characteristics of objects are still poorly depicted in database conceptual schemas and repositories. Following the successful use of a 2D PVL (Plug-in for Visual Language) extension with UML and of the integration of a rich repository for 2D spatial database models, we enriched the developed PVL with 3D elements to better define the spatial characteristics of 3D objects and consequently have more meaningful database contents. This paper presents this new set of PVL pictograms which can be used to better define the

R. Meersman et al. (Eds.): OTM Workshops 2005, LNCS 3762, pp. 999 – 1008, 2005.

geometry of 3D objects in a conceptual data model and repository. We first define 3D concepts that help to reduce semantic confusion. Afterward, we present the developed solution to improve the semantics of 3D objects geometries and discuss the importance of properly describing the geometric meaning of 3D objects. We focus on the conceptual level and voluntarily do not go deep in the underlying concepts, hoping to help clarifying concepts that still remain widely confused in scientific literature. The sole diversity of meanings that still exists for "3" and "D" hampers the proper use of 3D concepts by practitioners. In addition, formal meta-modeling and concepts related to levels of modeling, multiple representations, spatial and temporal relationships, generalisation, constraints and human cognition have been discussed in previous papers or technical reports and they underly the present paper.

2 Fundamental 3D Concepts

There exist different definitions of 3D objects. Often, there is confusion between the dimensions of the object shape and the dimensions of the space in which these objects are located. For example, according to ESRI, a three-dimensional shape is: "*a point, line, or polygon that stores x-, y-, and z-coordinates as part of its geometry. A point has one set of z-coordinates; lines and polygons have z-coordinates for each vertex*" [10]. Such definition appears semantically incorrect because it does not refer to the number of dimensions of the object shape (point 0D, line 1D or polygon 2D) but to the number of dimensions needed to locate these objects in a 3D universe. The definition given by Euclid's Elements[1] web site presents a different view: "*A solid is that which has length, breadth, and depth*". Such view defining a 3D shape as a solid avoids confusion between the number of dimensions of the universe and those of the object, as it is the case in 2D topology with 0-cell, 1-cell and 2-cell objects. Mathworld Web site[2] proposes a good definition: "*the dimension of an object is a topological measure of the size of its covering properties. Roughly speaking, it is the number of coordinates needed to specify a point on the object. For example, a rectangle is two-dimensional, while a cube is three-dimensional. The dimension of an object is sometimes also called its "dimensionality"*." In other words, the number of dimensions of an object is the number of coordinates necessary to uniquely locate a point in this object: 0 in a point, 1 in a line, 2 in a polygon and 3 in a solid. This is the definition that we have adopted as it is mathematically more robust.

Such definition also implies that objects can serve as a universe to locate other objects. It is the case for example with roads which are 1D objects usually located in a 2D or 3D universe but which can also be used as a 1D linear referencing system (LRS) to locate other objects like accidents, road signs, etc. The next paragraphs clarify the concepts of universe dimensions and objects dimensions.

2.1 Dimensions of a Universe

The number of dimensions of a universe corresponds to the number of spatial axis (or coordinates) needed to uniquely locate objects in this universe. For example, a 2D

[1] http://aleph0.clarku.edu/~djoyce/java/elements/elements.html
[2] http://mathworld.wolfram.com/Dimension.html

universe has 2 axes and a 3D universe has 3 axes. Objects located in a universe cannot have more dimensions than the universe used to locate this object except if this universe is itself an object located in another universe having more dimensions. For example, a 2D parking can be located at the offset of a 1D road between two points on this road (the road being a 1D universe located in a 2D universe).

2.2 Dimensions of Objects

In this paper, object means an element or feature of the reality represented by a shape in a spatial universe. The number of dimensions of this object follows Mathworld' mathematical definition, i.e. it is the number of axes needed to locate a point within this object when it is used as a universe. It is based on the space occupied by the object itself (e.g. length, width, thickness) and not the space occupied by its minimum bounding rectangle (MBR) or minimum bounding box (MBB) which are usually defined parallel to the coordinate axes in a universe with more dimensions. Accordingly, a line is a 1D object whether it is a straight, curvilinear (included in a 2D MBR) or a non-planar line (included in a 3D MBB).

3 3D Database Modeling with 3D PVL Expressions

Nowadays spatial database applications ask for enriching the semantics associated to geographical objects to support a wide variety of tasks such as data integration, interoperability, knowledge reuse and spatial reasoning. It is the role of conceptual data model, as Chen says [6], “to incorporate some of the important semantic information about the real world”. It also is their role to contribute to building a formal semantics. Semantics has varying meanings in sciences like Linguistics, Philosophy, Anthropology and Artificial Intelligence. In this paper, we use the definition given to semantics by Logic Science, that is “the study of relationships between signs and symbols and what they represent”[18]. In cartography and 3D modeling, signs and symbols are combined with geometry to convey a meaning to what we see, to help recognize objects. Visual variables (ex. color, line weight, line style, patterns) bear meaning and explicitly take part to the semantics of objects. The components of spatial reference (position, shape, size and orientation) can be neutral but can also bear meaning and then contribute to semantics. For example, on a 2D paper map, red points, red rectangles and red detailed polygons can be interpreted as small, medium and large buildings or as residential, commercial and public buildings according to the legend of this map. The legend adds meaning to the shape of objects while geometry allows one to infere spatial relationships which are meaningful for the understanding of a phenomena. In digital maps, some meaning of the geometric feature comes from its name, identifier and attributes. However, these are not sufficient to understand the complete meaning beared by a geometry. One may ask "what types of buildings are represented by points, rectangles and detailed polygons? Do points represent residential buildings, small buildings or both? Do polygons represent public buildings, buildings larger than 200m² or both? Are roads all of the same width or are they symbolized? etc." It is possible that such meaning isn't explicitely stored into attributes or cartographic layers or object classes but can only be deduced from the geometry and symbology. Consequently, geometric definitions stored in repositories to describe data acquisition specifications as well as the derivation rules (ex. generalization) are important to understand the meaning of these geometries and of the objects they represent. Table 1 shows a semantic table adapted to spatial databases where values are geospatial objects (in columns) and geometric categories (in row) which combinations correspond to the semantic of features. The first column is the genus (semantic group with common attributes) and the other columns are the differentia (attributes that serve to distinguish genus from each others). In conceptual data schema, the geometric category of objects is represented with PVL pictograms (explained in the next section) while the differentia allowing to distinguish each building type are described in the repository as specifications.

Table 1. Semantic table combining geometry and cartographic semiology with Genus and Differentia to distinguish different categories of buildings

	Genus	Differentia	
	Building	Public	Small
Point	+	-	+
Rectangle	+	-	-
Polygon	+	+	-

Now, let's suppose that roads are part of a 3D spatial database. What is a road in the context of this spatial database? At very large scale, which part of the road is used to digitize the polygon representing its boundaries: the main pavement ? the shoulders? cadastral boundaries? Is there a minimum length to create a new road? What is the granularity of the polygonal information, does-it include all small variations of its width at intersections? Is it the real shape or a simplified shape? How does-it start and ends at intersections? Does it cover all the roads of the city or only those connected to numbered roads? etc. Having the geometric description associated to this object class contributes to enrich the semantics of the objects by stating that the polygons representing the paved roads correspond to the limits of their pavement, that only roads longer than 250m and larger than 15m are digitized, that polygons don't include geometric changes smaller than 10% of the width of the road, and so on. Thus, in this specific example, shorter roads are not considered to be a road, minor width modifications are not represented, etc. It is thus important to define geometry in a conceptual model and a repository to better describe the meaning or semantic of objects and better understand the impact on spatial relationships.

To help database designers to describe the geometry of spatial objects, we developed the concept of Plug-in for Visual Language (PVL) that helps describing the geometry of object classes in a conceptual schema and a repository. Such PVLs are technology-independent and visual modeling-language independant, they help the analyst to better describe what users want without worrying about implementation issues. The next section presents the 3D PVL and the repository forms used to describe the geometry in detail. The PVL grammar rules are presented after.

3.1 PVL Pictograms and Their Repository Forms

PVL is a language composed of a small number of signs (called pictograms) and of a few grammar rules. PVL is meant to act as an extension to any modeling language and as such, is a specialized language of its own. PVL expressions, made of pictograms and rules, can represent visually in a conceptual schema the geometry and temporality of objects classes, attributes and processes of a spatiotemporal database. The three fundamental spatial pictograms of the PVL for a 2D universe are ⊡, ☑ and ◩ which designate respectively a 0D object , a 1D object and a 2D object. The two fundamental temporal pictograms are ◷ and ◔ respectively for instants (0D) and intervals (1D). Several articles have been written to describe these pictograms and the grammar used to combine them to describe complex geometries, complex temporalities and spatio-temporal objects [1, 3, 4, 5], they are not repeated here.

A specialized repository is always joined to the schema to detail PVL expressions when needed. In addition to including most of the semantic information found in ontologies, this repository contains information about digitizing objects or acquisition specifications, processes to derive objects' geometry and temporality, sources for geometry or temporality, etc. (Fig. 1). Metadata about spatial reference systems and quality information are stored in additional forms.

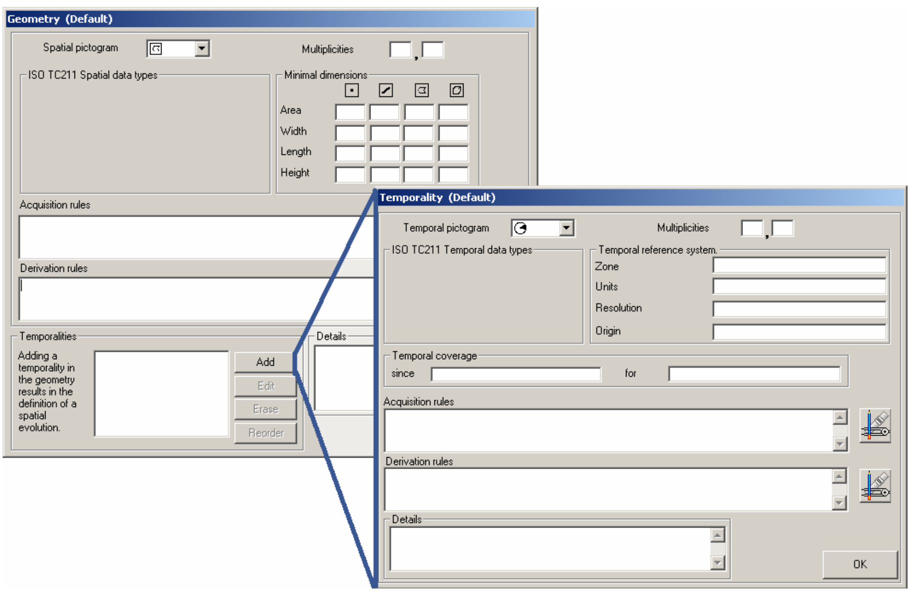


Fig. 1. Perceptory's repository forms used to enrich the semantics of objects and to detail the mapping specs in addition to storing their spatial and temporal pictograms

Recently, we added new pictograms to the PVL for 3D conceptual database modeling [1, 13], which differ from 2D spatial pictograms by showing a box to include the geometry instead of a square. This box represents the geometry of the universe (3D) that includes the shape of the object class. Explicitely depicting the number of dimensions of the universe in which is located an object has become necessary with the possibility of the most recent technologies to store several objects from different data sources in a same data warehouse or to produce different views in different universes (ex. Oracle Spatial). It has also become necessary with the increased importance of multi-representation databases.

To obtain the shape of the objects located in a 3D universe, we transpose each shape of the 2D spatial pictograms in the 3D box in a way that preserves their ground trace (2D aerial view) and that gives them a thickness (or elevation) or not. We obtain the six pictograms shown in Table 2.

Table 2. How a 2D geometry can be transposed in a 3D universe

Objects in a 2D universe	become in a 3D universe:	
	flat objects or objects draped on a DTM	objects with height or thickness
		
		
		

The PVL can be used with any CASE tools because its pictograms are included in a font. They are also used with Perceptory, a CASE tool developed especially for geospatial databases. In all cases, the number of dimensions of the universe and of each object class is depicted visually as explicitly as possible.

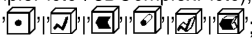
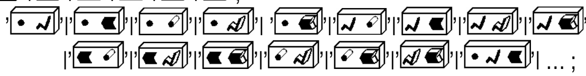
3.2 Grammar Rules to Combine Pictograms into More Complex 3D Geometries

The grammar rules used to generate the appropriate 3D PVL expressions are described in the next paragraph using the EBNF (Extended Backus-Naur Form) standard ISO/IEC 14977 [10].

Table 3. Used EBNF notation

Symb.	Meaning	Symb.	Meaning
=	Defining-symbol	( )	Start and End-group-symbol
,	Concatenate-symbol	[ ]	Start and end-option-group
|	Definition-separator-symbol	{ }	Start and End-repeat-symbol
’	Quote symbol	;	Terminator-symbol
(* *)	Start and End-comment-symbol	-	Exclusion-symbol

Hereafter, we present the rules concerning only the 3D PVL although it is possible to combine 3D pictograms with 2D pictograms in the case of multiple-representations spatial databases or with the temporal pictograms for spatiotemporal databases. It is also possible to use them for attributes and methods the same way as the other pictograms.

3DPicto = (3DSimplePicto | 3DComplexPicto), [Multiplicity];
3DSimplePicto = ;
3DComplexPicto =  | ... ;

Multiplicity = MinCardinality, ',' , MaxCardinality;
MinCardinality = Number *(*equal or greater than '0'*)*;
MaxCardinality = Number | 'N' *(* equal or greater than MinCardinality- '0'*)*;
3DDerivedPicto = 3DPicto *(*in italic to remind the UML derivation symbol '/'*);*

3DSimpleGeometry = 3DSimplePicto;
3DFacultativeGeometry = 3DPicto *(*MinCardinality =' 0'*)*;
3DAggregateGeometry= (3DSimpleAggregateGeometry | 3DComplexAggregateGeometry);
3DSimpleAggregateGeometry = 3DSimplePicto, Multiplicity *(*MaxCardinality – '1'*)*;
3DComplexAggregateGeometry = 3DComplexPicto;
3DAlternateGeometry = (3DSimpleGeometry | 3DFacultativeGeometry | 3DAggregateGeometry)|
{3DSimpleGeometry | 3DFacultativeGeometry | 3DAggregateGeometry }*(*3D spatial pictograms are adjacent on a same line*)*;
3DMultipleGeometries = (3DSimpleGeometry | 3DFacultativeGeometry | 3DAggregateGeometry | 3DAlternateGeometry),{3DSimpleGeometry | 3DFacultativeGeometry | 3DAggregateGeometry | 3DAlternateGeometry} (*3D spatial pictograms are one above the other on differents lines*);

The examples of Fig. 2 illustrate those rules where:

- *road segments* have a non-planar line geometry,
- *rivers* have a complex geometry, i.e. each object is represented by a combination of non-planar lines (narrow river segments) ***and*** polygons (large river segments) to create a unique complex geometry,
- *historical monuments* have an alternate geometry, i.e each object is represented by a vertical line (ex.: statue) ***or*** a simple solid (ex.: building) but not both (i.e. XOR),
- *buildings* have multiple geometries, i.e each object is represented by an aggregate of solids for large scales ***plus*** a derived simple solid for small scales;

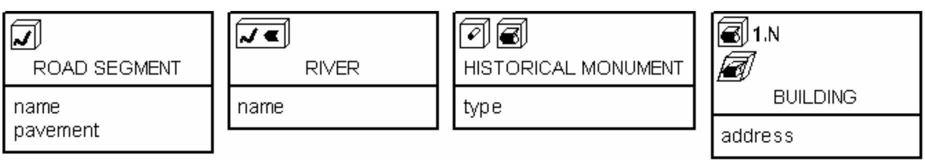


Fig. 2. Examples of PVL grammar rules in UML classes for objects respectively having simple, complex, alternate and multiple geometries (the latter including 2 geometries, one of them being a multipolygon aggregate)

4 Properly Describing the Geometric Meaning of 3D Objects

The introduction of 3D pictograms in conceptual schemas serves several roles. At the outset, it helps users to see more clearly what they want and are willing to support, update and pay for. For example, figures 3 and 4 illustrate what appears to be semantically the same features but they depict different 3D definitions. In the conceptual object classes of Fig. 3, x,y and z coordinates are meant to be measured in 3D for each object class. Trees are intended to be vertical lines, walls to be vertical plans, and buildings aggregates of plans (i.e. not full solids). In Fig. 3, the desired geometries are meant to be measured in a 2D universe. Then, 3D-like geometries are derived through two processes: one to give the bottom-z, one to give the top-z to those objects having an attribute 'height'. This will give, for example, buildings with flat roofs.

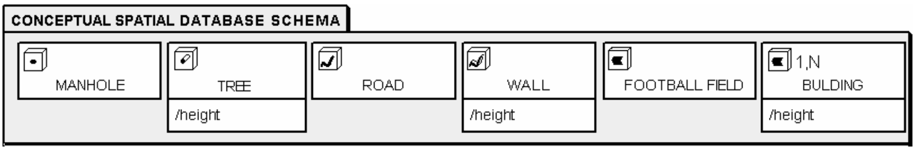


Fig. 3. Subset of a conceptual 3D data schema (not involving associations and methods)

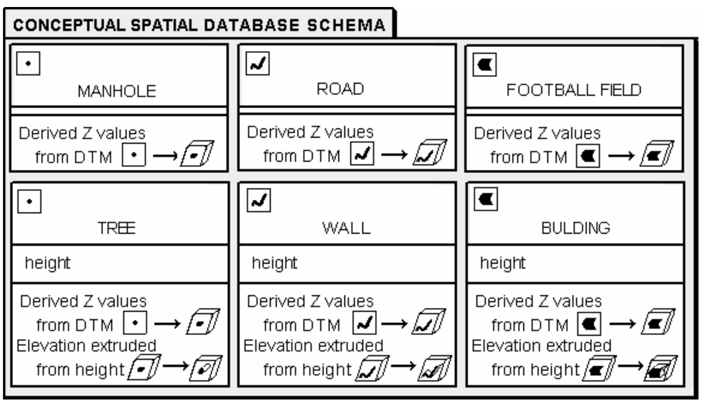


Fig. 4. Conceptual 3D data schema where objects are initially meant to be 2D but where Z-base can be derived from a DTM and Z-top from an attribute height

Such PVL expressions clearly highlight the concerns about the manipulation and analysis of the third dimension, thus helping to choose the best 3D software. For example, some GIS are only 2D and are limited to generate Grid and TIN. Other GIS offer 2½D environments, they can only have one z coordinate for each x,y pair and consequently do not support solids and don't allow perfectly vertical lines and plans (one needs to create microscopic offsets in order to have 2 z values for practically the same x,y pair). “Current GIS 3-D representation does not truly exist. Many existing GIS models are actually modeling a 2.5-D environment.” [18]. Using such strategy allows this category of GIS to add dimensionality to a 2D universe and give thickness to flat objects. Another alternative is to store x, y and z coordinates for each vertex as is supported by CAD systems or some DBMS, thus offering 3D.

These examples highlight the different meanings, geometrically speaking, that 3D objects may have. They also highlights the need to describe 3D objects in a way that meaningfully depicts what they are intended to be, independant from implementation.

Deciding the type of 3D objects a user truly needs requires a good understanding of the possibilities since the final choice about the intended 3D geometries has a major impact on the cost, delay and complexity of data acquisition, data processing and software selection. To properly express 3D needs, one can to use the proposed 3D pictograms in a way that meaningfully expresses these needs. This conceptual step is more delicate than in 2D because of the largest impact. We need to say more than "I want 3D objects", we need to have explicit meanings for 3D geometries. More meaningful 3D geometric information helps the user to interpret the true sense of objects to be included in a map or 3D model. When needed, details can be stored in the repository in natural or formal language, leading to richer 3D ontologies and better understanding of potential 3D relationships between objects.

5 Conclusion

The semantic of objects has been defined for several years with conceptual models and repositories. Including geometric definitions contributes to semantics. While spatial extensions for modeling languages have existed for over 15 years, nothing specific to 3D objects and universes existed insofar. The proposed 3D PVL paired with a repository allows one to define the subtelties of 3D geometry. It allows the analyst to create a conceptual database model that depicts a clear understanding of the several issues concerning 2D vs 2½D vs 3D, acquisition vs derivation of Z, 2D-thick objects vs true 3D-shape objects, etc. It allows one to see more clearly what he can expect from the database (ex. volumetric and 3D topological analysis) and the spatial relationships he can infere from it. In other words, it adds meaning to data models in a way that is cognitively compatible with most users and systems analysts.

Aknowledgements

The authors wish to acknowledge the financial support of Canada NSERC, Laval University, R&D Defense Canada. We are thankful to the users who send us feedbacks. We also thank the partners of the NSERC Indutrial Chair in geospatial database for decision support: Hydro-Quebec, RDDC, Intelec, KHEOPS, Syntell, DVP, Holonics, Transport Quebec, Natural Resources Canada.

References

1. Arens, C., Stoter, J., van Oosterom, P.: Modelling 3D objects in a Geo-DBMS using a 3D primitive. Computers & Geosciences, 31: (2005) 165-177.
2. 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

3. Bédard, Y., Pouliot, J., Larrivée, S., Frenette, P., Normand, P., Brisebois, A.: Création d’un modèle 3D urbain de la recherche de données à l’exploitation du modèle 3D. Research Report realised for Defence Research and Development Canada (2002)
4. Bédard, Y.: Visual Modelling of Spatial Database towards Spatial PVL and UML. Geomatica, 53(2) (1999) 169-185
5. Bédard, Y., Paquette F.: Extending entity/relationship formalism for spatial information systems. AUTO-CARTO 9, April 2-7, Baltimore (1989) 818-827
6. Billen, R. Nouvelle perception de la spatialité des objets et de leurs relations. Développement d'une modélisation tridimensionnelle de l'information spatiale. Ph.D. Theses, Université de Liège, Faculté de Sciences, Département de géographie, 2002.
7. Chen, P. P.: The Entity-Relationship Model - Toward a Unified View of Data. ACM Transactions on Database Systems (TODS), vol 1, n° 1 (1976) 9-36
8. Deheneffe, C., Hennebert, H., Paulus, W.: Relational model for data base. Proc. IFIP Congress, North-Holland Pub. Co., Amsterdam (1974) 1022-1025
9. Hainaut, J.L., Lecharlier, B.: An extensible semantic model of data base and its data language. Proc. IFIP Congress, North-Holland Pub. Co., Amsterdam (1974) 1026-1030.
10. ISO/IEC 14977: Information technology -- Syntactic metalanguage -- Extended BNF (1996)
11. Karman, M., Amdahl, G.: Dictionary of GIS terminology. ESRI Press (2001)
12. Kraus, K.: Photogrammetry: Advanced Methods and Applications, Dümmler-Verlag (1997)
13. Larrivée, S., Bédard, Y., Pouliot, J.: Modélisation conceptuelle des bases de données géospatiales pour des applications 3D. Revue internationale de géomatique, numéro spécial: Information géographique tridimensionnelle: théories, systèmes et applications (2006)
14. Molenaar, M.: A formal data structure for 3D vector maps. Proceedings of EGIS'90, Amsterdam, The Netherlands (1990) 780-781
15. Parent, C., Spaccapietra, S., Zimanyi, E., Donini, P.: Modeling Spatial Data in the MADS Conceptual Model. Int. Symp. on Spatial Data Handling, Vancouver (1998) 138-150
16. Schmid, H. A., Swenson, J. R.: On the semantics of the relational model. Proc. ACMSIGMOD, Conference, San Jose, Calif. (1975) 211-233
17. Shekhar, S., Vatsavai1, R.R., Chawla, S., Burk, T. E.: Spatial Pictogram Enhanced Conceptual Data Models and Their Translation to Logical Data Models. ISD'99, LNCS, Vol. 1737, Springer Verlag, Berlin Heidelberg New York (1999) 77-104
18. The American Heritage Dictionary of the English Language. Houghton Mifflin (1981)
19. Thurston, J.: Geo-Visualisation : Current Issues / Future Potentials. GISCafe.com (2001)