TOPOLOGY FOR 3D SPATIAL OBJECTS

Siyka Zlatanova[1], Alias Abdul Rahman[2], and Wenzhong Shi[3]

[1]GIS Technology Section,
Delft University of Technology
Thijssweg 11, 2629 JA, Delft, The Netherlands.
s.zlatanova@citg.tudelft.nl

[2]Center for Geographic Information and Analysis (CGIA)
Universiti Teknologi Malaysia
81310 UTM Skudai, Johor, Malaysia.
alias@fksg.utm.my

[3]Advanced Research Centre for Spatial Information Technology
Department of Land Surveying and Geo-Informatics
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong.
lswzshi@polyu.edu.hk

KEY WORDS: 3D topology, 3D data structures, Object-oriented structures, 3D analysis, spatial operations

ABSTRACT:
Topology is one of the mechanisms to describe relationships between spatial objects and thus the basics for many spatial operations. In this paper, we present models that are built on the topological properties of the spatial objects. They are usually called topological models and are considered by many the best suited for complex spatial analysis (i.e. shortest path, line of view). There are a number of topological models that are utilized for 2D and 2.5D spatial objects by experimental and commercial software. However, when we move to the next dimension (i.e. 3D), many difficulties are encountered in establishing the topology for the objects (consisting of points, lines, faces, and solids). This paper describes some existing topological models and a comparison between them is made. The advantages and disadvantages of the models and the recent experiments by the authors towards formalizing a 3D topology for 3D objects is discussed. The paper considers the models in object-oriented (OO) environment as well. Finally, we summarise the application of the 3D topological model, highlight the current approaches and the outlook of the works.

1 INTRODUCTION

Spatial analysis is often considered the most important task of a geo-information processing. 3D analysis is still one of the most challenging topics for research. Two aspects can be distinguished here, i.e. how to represent the objects and their spatial relationships and what kind of techniques to apply for detecting the relationships. Maintaining information about neighbouring objects (primitives, elements), i.e. topology, is the most widely used approach for representing relationships. In this respect, many data structures encapsulating different spatial relationships have been already reported in the literature. Frameworks to detect relationships (independently of the data structure) are also available. This paper gives a short overview.

2 3D TOPOLOGICAL MODELS

Discussing data structures, many application-related issues has to be taken into consideration, e.g. the space partitioning (full, embedding), the object components (volumes, faces), the construction rules (planarity, intersection constraints, etc.). The data structures reported currently in the literature can be subdivided in two large groups: structures maintaining objects and those maintaining relationships. While in the first group (object oriented), most of the relationships between the objects have to be derived, in the second group (topology oriented), the representation of the objects has to be derived. Many structures, which are a typical example of explicit storage of objects, maintain also explicit storage of relationships, i.e. singularities.

2.1 3D topological models with explicit representation of objects

3D FDS: The Formal Data Structure is the first data structure that considers the spatial object an integration of geometric and thematic properties. A conceptual model and 12 conventions (rules for partitioning of physical objects) define the structure (Molenaar 1990). Rikkers et al, 1993 propose mapping into a relational database (Figure 1). The model consists of three fundamental levels: feature (related to a thematic class), four elementary objects (*point*, *line*, *surface* and *body*) and four primitives (*node*, *arc*, *face* and *edge*). According to the conventions, *arcs* and *faces* cannot intersect. A *node* and an *arc* must be created instead. Singularities are permitted in such a way that *arcs* and *nodes* can exist inside *faces* or *bodies*. The role of the *edge* is dual, i.e. to define the border of a *face* (relationship *face-arc*) and establish an orientation for a *face*, which is needed to specify *left* and *right body*. The number of *arcs* constituting an *edge* is not restricted. *Arcs* must be straight lines and *faces* must be planar. The *surface* has one outer boundary and may have several non-nested boundaries, i.e. may have holes or islands. The *body* has one outer surface and can have several non-nested bodies or holes.

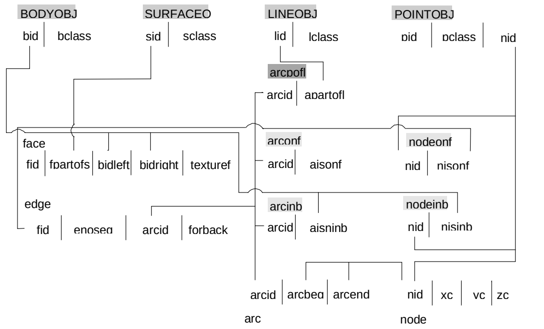

Figure 1: 3D Formal Data Structure (3DFDS): relational implementation, Rikkers et al, 1993

The fundament of 3D FDS is the concept for a single-valued map, i.e. *node*, *arc*, *face* or *edge* can appear in the description of only one geometric object of the same dimension (Molenaar 1989). The idea of the single-valued approach is to partition the space into non-overlapping objects and thus ensuring 1:1 relationships between the primitives and the objects of same dimensions, e.g. *surfaces* and *face*. Primitives of different dimensions, however, can overlap, e.g. relationships *node-on-face*, *arc-on-face*, *node-in-body* and *arc-in-body* are explicitly stored.

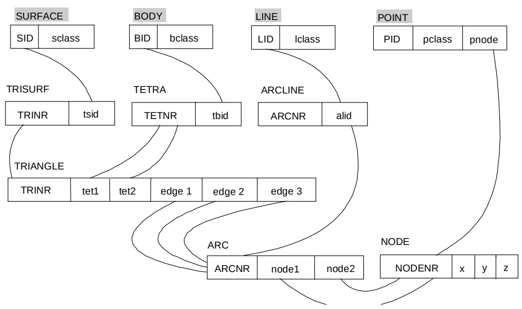

Figure 2: Tetrahedral Network (TEN): relational implementation for 3D, Pilouk 1996

3D FDS used by many to incorporate 3D objects. For example, Shibasaki and Shaobo, 1992 implement the model for maintenance and visualisation of 3D city models. De Hoop et al., 1993 investigate possible relationships (based on the 9-intersection model) for 3D FDS. The CC-modeller presented by Grün, and Wang, 1998, records 3D reconstructed objects in a schema similar to 3D FDS but extended to incorporate textures per face.

TEN: TEtrahedral Network (Figure 2) was introduced by (Pilouk 1996) to overcome some difficulties of 3D FDS in modelling objects with indiscernible boundaries (such as geological formations, pollution clouds, etc.). TEN follows simplex-oriented approach to represent 3D object from real world proposed by (Carlson 1987). Similarly to it, TEN has four primitives (*tetrahedron*, *triangle*, *arc* and *node*). In the relational implementation, the relationship *arc-node* is given by the ARC table; the TRIANGLE table contains the *tetrahedron-triangle-edge* link. A *body* is composed of *tetrahedrons*, a *surface* of *triangles*, a *line* of *arcs* and a *point* of *nodes*. The general rule for creating the model is based on the fact that each *node* is part of an *arc*, each *arc* is part of a *triangle* and each *triangle* is part of a *tetrahedron*. Singularities are not permitted. Since the model uses the simplex concept, TEN can be expected to cover the scope of possible topological relations in 3D space.

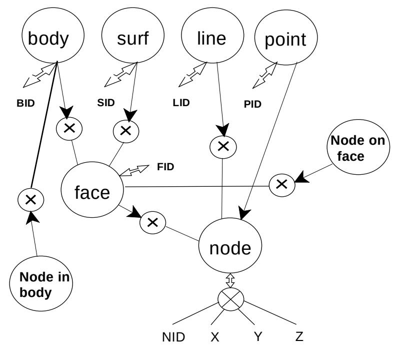

Figure 3: Simplified Spatial Model (SSM), Zlatanova, 2000

SSS: The Simplified Spatial Model (Figure 3) was designed to serve web-oriented applications with many visualisation queries (Zlatanova 2000). The basic objects are again four but the primitives used are only two, i.e. *node* and *face*. The motivation for omitting the arc of explicitly stored elements is that the uniqueness of the relationship arc/face in 3D space is lost, i.e. one *arc* can be part of more than two *faces*. 3D primitive is not maintained as well and convex *faces* represent the 3D objects. *Faces* must be planar. Intersecting primitives lead to new once. The singularities *node-in-face* and *face-in-body* are explicitly stored.

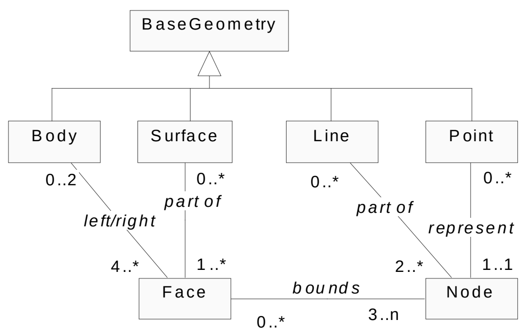

Figure 4: Urban Data Model (UDM), Coors, 2002

UDM: The Urban Data Model (Figure 4) represents the geometry of a *body* or a *surface* by planar convex *faces* (Coors, 2002). Each *face* is defined by a set of *nodes*. Two convex planar *faces* are adjacent if they share at least two *nodes*. The orientation of a *face* is stored implicitly. In the relational representation of the model, every *face* having more than three *nodes* is decomposed into triangles and the FACE table contains only three columns, i.e. the ID's of the three triangle *nodes*. The one-dimensional construction primitive (*arc*) is not supported as well. This primitive however can be implicitly defined by two successive *nodes*. Similarly to 3D FDS the relationships *face_body* is explicitly stored in the FACE table. The partition of the objects is indeed higher, all the surfaces have to be triangulated. Depending on the complexity of the surfaces (e.g. number of windows on a wall), this triangulation may lead to increase of the database. However, in case of simple façades (e.g. without windows), the constant number of columns in the face table compensates the increased number of element for maintenance. Singularities are relatively reduced, i.e. the relationships *node-on-face* and *arc-on-face* are resolved.

2.2 Object-oriented models

The models mentioned above are mapped in relational DBMS, which is often considered less appropriate for describing real-world objects. Abdul-Rahman (2000) utilises the FDS model (Molenaar, 1998) and implemented the 3D TIN based spatial objects in object-oriented environment (i.e. by using the commercial OO DBMS, that is the Persistent Object and Extended Technology, POET OO DBMS). The schema of the model looks as in Figure 5 where 3D objects (such as boreholes) are represented by a series of 3D TINs. The model works with four spatial primitives (node, line, surface, and solid).

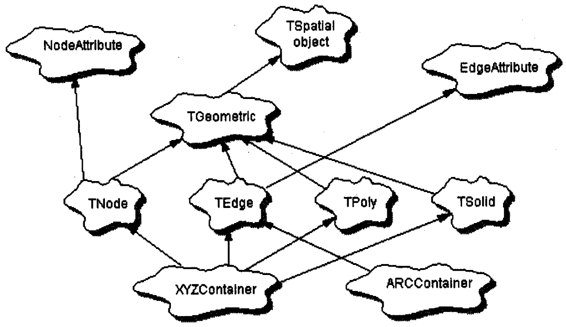

Figure 5: 3D TIN-based OO model

Simple topological relationships of the objects (TIN based) could be established such as point-line, point-surface, point-solid, line-surface, and line-solid. All the constructed classes of the model are then mapped according to the POET OO DMBS database schema.

Other solutions of structures explicitly maintaining objects are presented by (Pfund 2001), i.e. the Solid Object Management System (SOMAS) or the model of (de la Losa. and Cervelle 1999). Figure 6 and Figure 7 show the conceptual models. The structures, however, are not implemented in a DBMS.

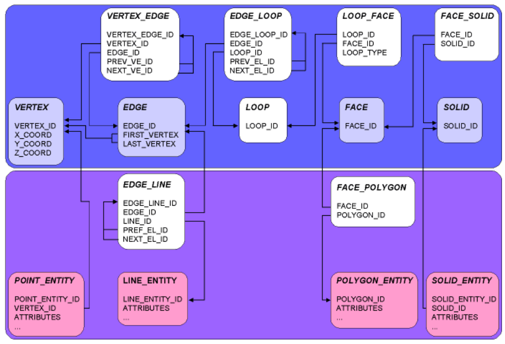

Figure 6: SOMAS, Pfund 2001

The authors of OO-model propose the order of the faces with respect to a common edge to be explicitly maintained in the model. Thus, the normal vector of each face is determined by the direction of the edge (and may not be always directed toward outside of the 3D object).

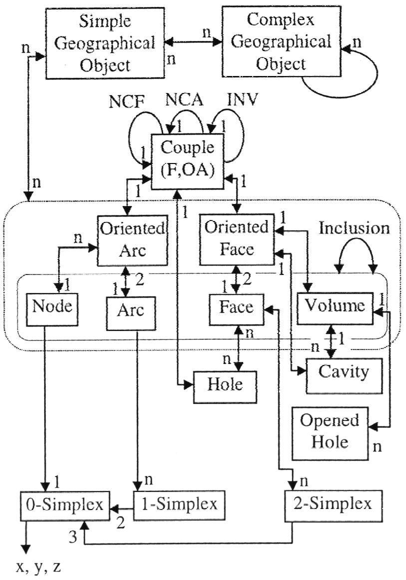

Figure 7: OO-model of de la Losa and Cervelle, 1999

Shi, Yang and Li (2002) developed an object-oriented data model for handling complex 3D objects in GIS (OO3D). First, the conceptual data model is developed based on the principle of object-oriented data modelling. This model is designed based on the following three basic geometric elements: node, segment and triangle. Accordingly, the abstract geometric objects are defined; these include points, lines, surfaces and volumes. Second, the corresponding 3D logical model is designed based on the defined abstract objects and the relationships between them. Third, a formal representation of the 3D spatial objects is described in detail. The model is applied in a 3D GIS developed – SpaceInfo.

2.3 3D structures with explicit representations of relationships

The spatial model introduced by (Brisson 1990) and extended by (Pigot 1992) is referred to as the tuple model. It defines cells and cell complexes upon the fundamental properties of a manifold. The k-cell complex is the union of all the k-dimensional and lower cells. Some later extensions (Mesgari 2000) of the model permit the existence of singularities, e.g. 0-cell inside 2-cell, 2-cell inside 2-cell (holes), 3-cell inside 3-cell (tunnels). Under these circumstances, any spatial object can be described as a set of tuples of 3-cell, 2-cell, 1-cell and 0-cell, i.e. the representation of cells is implicit. From construction point of view, the model permits cells with an arbitrary shape.

2.4 Comparison of different models

Clearly, advantages of a model in one of the aspects occur as disadvantages in another aspect. For example, the arbitrary number of nodes per face can be seen as advantage and disadvantage for different application. It is very convenient for modelling complex 3D objects (e.g. buildings) since an inappropriate partitioning (from user point of view) is not necessary and a 3D object can be represented by the faces on the boundary. However, the same freedom in face description may lead to problems in visualisation (the rendering engines handle only triangles). Furthermore, the operators for consistency check become very complex. Another example is the relationship face/body. It is very convenient for navigating through 3D objects, but in some cases (e.g. urban areas) may lead to storage of non-significant data (i.e. “open air” also has to be stored as a right body).

The major problem with TEN refers to the modelling stage. Since the space is completely subdivided into *tetrahedrons*, the interiors of objects (e.g. buildings), as well as the open space, are also decomposed into *tetrahedrons*. Such subdivision is rather inconvenient for 3D man-made objects. The author (Pilouk 1996) suggests these objects to be represented as 3D FDS features in TEN. However, the subdivision into *triangles* furnishes the data needed for display of graphic information in the most appropriate way. In this respect, TEN and UDM are perhaps the optimal models for visualisation of surfaces. Maintenance of *triangles* solves other modelling problems as holes or explicit storage of relationships (such as *arc-on-face* and *node-on-face*). An additional disadvantage for TEN is the much larger database compare to other representations, and the need for special processing of the *tetrahedrons* that are not needed for visualisation.

Due to the omission of *arcs*, data structures (SSS, UDM) can benefit form the significantly faster data traverse. However, navigating trough *surfaces* (e.g. “follow shortest path”) may become time-consuming. Representing *bodies* as set of *faces* (e.g. SSS) yields advantages for extracting the geometries of the objects, but navigation queries might be disturbed since the co-boundary relationships (i.e. *face-body*) is not explicitly maintained (i.e. it has to be derived).

The cell tuple data structure provides the largest spectrum of topological relations between cells and complex cells. Furthermore, the model promises an easy maintenance, due to the solid mathematical foundations. In the visualisation respect, the extraction of faces and points is a simple operation, due to the explicitly stored link between the cells. The data obtained from the tuple representation, however, lacks any indication regarding the order. Supplementary records are needed to establish the order (clockwise or anti-clockwise) of cells (note that the cyclic is ensured). Assuming a relational implementation, the entire tuple information is available in one relational table, which has advantages and disadvantages. On one hand, there is no need to perform JOIN operations to select any data. On the other hand, the size of the table grows tremendously, which slows down the speed of SELECT operations. For example, the records for a simple box occupy double space compared with 3D FDS.

One of the major advantages of object oriented 3D models is their capability in handling complex 3D objects, in comparison with the those models that are designed for mainly handling simple objects. This further improvement is particularly crucial for developing a cyber city for large cities where many complex objects exist, such as complex buildings.

Some of the OO models are designed with a compact characteristic, for example OO3D model has the basic elements -- node, segment and triangle. This design differs from the TEN and 3D FDS models (no Arc element as in the TEN and 3D FDS models). This design reduces data storage in the construction of spatial objects. However, due to the reason that the topologic relationships are not stored explicitly, the performance on some of the spatial analysis-related applications might not be as efficient other 3D models.

3 SPATIAL ANALISYS

3.1 Frameworks for representing spatial relationships

Three different approaches to encoding spatial relationships are discussed in the literature, i.e. metric, topology and order. The metric is a pure computational approach, based on the comparison of numerical values related to the location of the objects in the space. For example, the spatial relationship between a house and a parcel (e.g. inside, outside, to the south) can be clarified by a metric operation point-in-polygon performed for each point constituting the footprint of the building. The order establishes a preference based on the mathematical relation "<" (strict order) or "≤" (partial order), which allows an organisation of objects similar to a tree. For example, if a building is inside a parcel, the spatial relationship is represented as "building < parcel". The applicability to representing spatial relationships is investigated by (Kainz 1989) who argues that it has advantages in expressions of inside/outside relationships.

Topology allows the encoding of spatial relationships based on the neighbourhoods of objects regardless of the distance between them. The main property of topology, i.e. the invariance under topological transformations (i.e. rotation, scaling and translation) makes it appropriate for computer maintenance of spatial relationships. The following section discusses the general framework based on topology.

The 9-intersection model (Egenhofer and Herrring 1990): The framework utilises the fundamental notions of general topology for topological primitives to investigate the interactions of the spatial objects. The topological primitives of a spatial object can be defined for each spatial model and hence the framework can be applied to any spatial model. The basic criterion to distinguish between different relations is the detection of empty and non-empty intersections between topological primitives. Depending on the number of the topological primitives considered, two intersection models were presented in the literature. The first idea is to investigate the intersection of interiors and boundaries of two objects. This results in $2^4=16$ relations between two objects. Apparently, many relations cannot be distinguished on the basis of only two topological primitives, therefore the evaluation of the exterior is adopted. The number of detectable relations between two objects increases to $2^9=512$. Eight relations are possible between 3D and 3D objects and they are given names, i.e. *disjoint*, *meet*, *contains*, *covers*, *inside*, *coveredBy*, *equal* and *overlap* (Figure 8). For example, if the boundaries of the two objects intersect but the interiors do not, then the conclusion is that the objects *meet*. Despite the criticism (i.e. not all the relations are possible in reality, the intersections are not further investigated, many object intersections are topologically equivalent), the framework provides a systematic, easy-to-implement way of detecting spatial relations

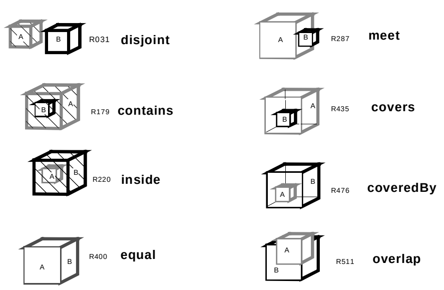

Figure 8: The 9-intersection model: possible relationships between 3D and 3D objects

The Dimensional model (DM): The DM is another framework utilising order of points, which is related to the study of affine space (a subspace of the topological space) and convex shapes. The formal definition of the model can be found in (Billen et al 2002). Here, we will use a simple example. If one looks at a triangle in R^2, the points of order 0 are the vertices, the points on the edge have order 1 and the points of order 2 are all the points that are "inside" the triangle. Applying this formalism, spatial objects can be described and their spatial relationships can be decoded. In the 3D Euclidean space (R^3), four types of dimensional elements are allowed, i.e. 0D, 1D, 2D and 3D elements. For example, a polygon has a 2D-element, a 1D-element and a 0D-element. The 2D-element coincides with the spatial object (i.e. the polygon). To represent the dimensional relationships between two objects, one has to consider all the dimensional elements of these elements. For example, the dimensional relationships between two simple spatial objects of dimension 2 (i.e. polygons A and B) can be defined in the following order: first, check the dimensional relationship between 2D element of A and all the dimensional elements of spatial object B; then, check the dimensional relationship between 1D element of A and all the dimensional elements of spatial object B, etc.

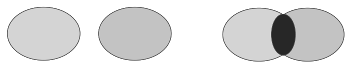

Figure 9: The Dimensional relationships: *non-existent*, *partial* and *total*

The dimensional relationship can be *partial*, *total* or *non-existent*, depending on the interaction between the interiors of the objects (Figure 9). The benefit of these frameworks is mostly in the flexibility while deciding on which dimensional elements are to be used. In general, a larger number of relationships can be distinguished compare to the 9-intersection model (see Figure 10).

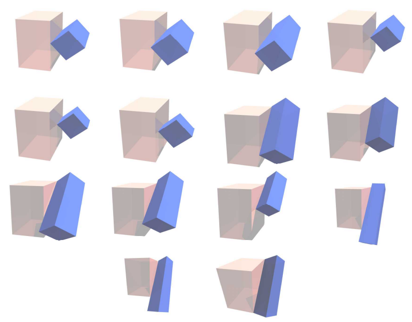

Figure 10 The Dimensional model: possible relationships between 3D and 3D objects

3.2 Spatial Operators

Having the data structure and the framework for representing relationships specified, the next step is defining the operations that have to be supplied by the system. The operations describe all the actions that can be performed on the data. First of all, operations to "build" consistent data structure and update it have to be investigated and developed. For example:

- operations to organise the data according to the data structure, i.e. operations for planarity, convexity and discontinuity as they are defined in the model.

- operators for consistency check: validation of the objects (e.g. polygon closed, body closed), *node-on-line, node-on-face, node-in-body, line-on-face, line-in-body, intersection of lines, face-on-face, intersection of faces, face-in-body.*
- 3D overlay, which is based on the same operation for consistency check and 3D editing.
- operation for 3D editing: add, delete and update of cells.

Apart from these constructing operators, GISs have to perform a number of specialised operations such as selection, navigation and specialisation. (Molenaar 1998) specifies the GIS query as a selection operation with three components: data type specification, conditions and operations that have to be performed on the data. The selection then can be performed on semantics, geometry or topology. For example, "select the buildings (data type) higher than 15m (condition) and show their ID (operation)". Sophisticated operations on data may diminish the boundary between query and analysis. Theoretically, the original operation and the further processing can be encapsulated in a new operation. Many classifications of the operations can be found in the literature (Aronoff 1995, Goodchild 1987). In general, the operations can be subdivided in three large groups with respect to the geometric and semantic characteristics and the spatial relationships. Most interesting are the operations related to the geometry and the spatial relationships. One classification of these operations follows:

- metric operations are selection operations based on shape and size of objects and further computations e.g. compute distance, volume, area, length, centre of gravity, intersect;
- position operations are selection operations based on position (no further processing), e.g. objects in a certain area;
- proximity operations are selection operations based on geometric characteristics and the creation of new object, e.g. buffer, convex hull, union of objects;
- relationship operations are selection operations based on spatial relationships (no further processing), e.g. neighbouring operations, overlay;
- network operations are selection operations based on spatial relationships and geometries and further processing (with different level of complexity), e.g. route planning
- visibility operations are a selection based on geometric characteristics and further processing, e.g. sign of view
- semantic operations are selections based on semantic characteristics
- mixed operations, i.e. selections on the basis of geometric and semantic characteristics.

Apparently, the operations related to the spatial relationships of the objects are highly influenced by the data structure. As mentioned in the previous section some of the structures may appear better suited to perform certain queries than others. Moreover, it should not be forgotten that spatial analysis could be performed on geometric models as well. Many relational DBMS offer support of spatial object (in geometric model) and supply a number of spatial operations (validation, point-in-polygon, objects-within-distance, area, length, etc.). However, the operations (although some of them accept 3D faces) make use of only X, Y coordinates.

4 CONCLUSIONS

In this paper, we have given a short overview on topological models implemented in relational or OO-s and discussed two frameworks for detecting spatial relationships between objects. Bearing in mind the discussion on advantages and disadvantages of the different models we have to conclude that selecting an appropriate structure is a complex process related to the application (objects of interest, resolution, required spatial analysis, etc.). A model that is good for 3D spatial analysis may exhibit insufficient performance for 3D visualisation and navigation. Moreover, the implementation (relational or object-oriented) of the model has also has impact on the performance.

Following the current trends for integrated maintenance of spatial and non-spatial data, many DBMSs have already provided support of spatial objects. According to the abstract OpenGIS specifications (Open GIS consortium Inc. 1999), the spatial objects are to be maintained in the database with their geometric and topologic representations, as conversion operations have to ensure the consistency between the two models. This does not necessarily mean that one 3D topological model has to be accepted by all vendors for implementation. As discussed above different models may be appropriate for different tasks. Oosterom et al 2002 propose maintaining multiple topological models in one database by describing the objects, rules and constraints of each model in a metadata table. Such an approach will ensure maximal efficiency and effectiveness in providing the user with a large number of operations. Metric, position operations as area or volume computations will be presented on the geometric model, while relationship operations such as “meet”, “overlap” (basic for more complex spatial analysis) will be performed on the topological model.

References

Abdul-Rahman, A., 2000, The design and implementation of two and three-dimensional triangular irregular network (TIN) based GIS, PhD thesis, University of Glasgow, United Kingdom, 250 p.

Aronoff, S., 1995, Geographic Information Systems: A Management Perspective, WDL publications, Ottawa, Canada, 293 p.

Billen R., S. Zlatanova, P. Mathonet and F. Boniver, 2002, The Dimensional Model: a framework to distinguish spatial relationships, in: Advances in Spatial Data handling, D.Richardson, P.van Oosterom (Eds.), Springer, pp. 285-298

Brisson, E., 1990, Representation of d-Dimensional Geometric Objects, PhD Thesis, University of Washington, USA, 165 p.

Carlson, E., 1987, Three dimensional conceptual modelling of subsurface structures, Technical Papers of ASPRS/ACSM Annual Convention, Baltimore, Vol. 4 (Cartography), pp. 188-200

Coors, V., 2002, 3D GIS in Networking environments, CEUS (to be published), 17 p.

Egenhofer, M. J. and J. R. Herring, 1990, A mathematical framework for the definition of topological relationships, in: Proceedings of Fourth International Symposium on SDH, Zurich, Switzerland, pp. 803-813

Goodchild, M., 1987, A spatial analytical perspective on geographical information systems, International Journal of GIS, Vol.1, No.4, pp.327-334

Grün, A and X. Wang, 1998, CC-modeller: a topology generator for 3D city models, ISPRS Journal, Vol 53, No 5., pp. 286, 295

de Hoop, S, L. van de Meij and M. Molenaar, 1993, Topological relations in 3D vector maps, in: Proceedings of the 4th EGIS, Genoa, Italy, pp.448-455

Kainz, W., 1989, Order, topology and metric in GIS, ASPRS_ACSM Annual Convention, Vol. 4, Baltimore, pp. 154-160

de la Losa, A. and B. Cervelle, 1999, 3D topological modelling and visualisation for 3D GIS, Computer& Graphics, Vol.23

Mesgary, S. M., 2000, Topological cell-tuple structures for three-dimensional spatial data, ITC dissertation number 74, ITC, The Netherlands, 200 p.

Molenaar, M., 1998, An Introduction to the theory of spatial objects modelling, Taylor&Francis, London

Molenaar, M., 1990, A formal data structure for 3D vector maps, in: Proceedings of EGIS'90, Vol. 2, Amsterdam, The Netherlands, pp. 770-781

Oosterom. P.J.M. van, J.E. Stoter, S. Zlatanova, W.C. Quak, 2002, The balance between Geometry and Topology, Advances in Spatial Data Handling, 10th International Symposium on Spatial Data Handling, D.Richardson and P.van Oosterom (Eds.), Springer-Verlag, Berlin, pp. 209-224

Open GIS Consortium, Inc., 1999, The OpenGIS abstract specification, topic 1: Feature geometry. Technical Report Version 4 (99-101.doc), OGC, URL: http://www.opengis.org/techno/abstract.htm

Pfund, M., 2001, Topologic data structure for a 3D GIS, Proceedings of ISPRS, Vol.34, Part 2W2, 23-25 May, Bangkok, Thailand, pp. 233-237

Pigot, S., 1995, A topological model for a 3-dimensional Spatial Information System, PhD thesis, University of Tasmania, Australia

Pilouk, M., 1996, Integrated modelling for 3D GIS, PhD thesis, ITC, The Netherlands

Rikkers R., M. Molenaar and J. Stuiver, 1993, A Query Oriented Implementation of a 3D Topologic Data structure, EGIS'93, Vol.2, Genoa, Italy, pp. 1411-1420

Shi, W.Z., B.S Yang and Q. Q. Li, 2002, An Object-Oriented Data Model for Complex Objects in Three-Dimensional Geographic Information Systems. International Journal of Geographic Information Science. (in press)

Shibasaki, R. and H. Shaobo, 1992, A digital urban space model - a three dimensional modelling technique of urban space in a GIS environment, ISPRS, Vol. XXIX, Part B4, Commission IV Washington, D.C., USA, pp. 257-264

Zlatanova, S., 2000, 3D GIS for Urban development, PhD thesis, ITC, The Netherlands, 222 p.

Zlatanova, S., A.A. Rahman and M.Pilouk, 2002, 3D GIS: current status and perspectives, , in *Proceedings of the Joint Conference on Geo-spatial theory, Processing and Applications*, 8-12 July, Ottawa, Canada, 6p. CDROM