Topological Operators in a 3D Spatial Query Language for Building Information Models*

André Borrmann and Ernst Rank

Computation in Engineering, Technische Universität München, München 80290, Germany

Abstract: The paper describes the definition and implementation of topological operators which form part of a 3D Spatial Query Language that has been developed to query Building Information Models using spatial conditions. The partial models resulting from spatial queries can be used as input for numerical simulations and as exclusively modifiable subsets in the context of collaborative planning activities, for example. Because human language is vague and ambiguous, the semantics of the provided 3D topological predicates are formally defined using an adapted version of the 9-intersection model originally developed by the GIS research community for 2D space. The paper further describes a novel approach for implementing 3D topological operators based on an octree representation of the geometric objects whose topological constellation is the subject of investigation.

Key words: Building Information Modeling, Spatial Query Language, Topology, 9-Intersection Model, Octree

Introduction

The computer-based modeling of buildings, also known as Building Information Modeling (BIM), has been an important topic in the AEC research community for more than 15 years now. Today we can observe how the technological concepts developed in the context of our research are exerting more and more influence on the AEC industry. This is mainly due to the advent of matured standards such as the Industry Foundation Classes (IFC) and reliable software tools implementing theses standards.

One of the most important components of a modern IT infrastructure supporting AEC processes is a central model management server, also known as a product model server, that centrally stores the Building Information Model and manages all access to it. Commercially available product model servers specialized in handling IFC data include the *Secom IFC Model Server*, the *Jotne EDMServer* and the *EuroStep Model Server*, for example.

To allow the user to extract parts of the full building model, the product model servers provide query languages which make it possible to formulate conditions that need to be fulfilled by the resulting set of building components. However, none of the existing query languages provides the option of including spatial conditions.

This can be explained by the conceptual structure of Building Information Models in use today: the IFC model, for example, is primarily designed from a semantic point of view, i.e. building components, their attributes and relationships are described in an abstract object-oriented way. Consequently, the geometry of a building component is not modeled explicitly, i.e. not by means of a boundary or CSG representation, but implicitly by using attributes with a geometric meaning. Since product model servers do not know the geomet-

Received: 0000-00-00; revised: 0000-00-00

* Supported by the German Research Foundation (DFG)

+ To whom correspondence should be addressed.

E-mail: borrmann@bv.tum.de; Tel: +49-89-289-25117

ric implications of semantic attributes, they are not able to interpret and process spatial information.

This has to be seen as a major deficiency, since spatial relations between building components play a significant role in most of the design and engineering tasks of the AEC domain. To fill this technological gap we have developed concepts and techniques for a 3D Spatial Query Language for Building Information Models. It makes it possible to select specific building components by means of spatial constraints.

Possible applications of the developed 3D Spatial Query Language for Building Information Models range from verifying construction rules to extracting partial models that fulfill particular spatial constraints. Such a partial model resulting from a spatial query may serve as input for a numerical simulation or analysis, or might be made exclusively accessible to certain participants in a collaborative scenario.

The proposed 3D Spatial Query Language relies on a spatial algebra that is formally defined by means of point set theory and point set topology [1][2]. Besides fully three-dimensional objects of type *Body*, the algebra also provides abstractions for spatial objects with reduced dimensionality, namely by the types *Point, Line* and *Surface*. This is necessary because building models often comprise dimensionally reduced entities, such as load points, power lines, plates, slabs etc. All types of spatial objects are subsumed by the super-type *SpatialObject*.

The spatial operators available for the spatial types are the most important part of the algebra. They comprise

- metric (distance, closerThan, fartherThan etc.),
- directional (above, below, northOf etc.) and
- topological (touch, within, contains etc.)

operators.

While the metric and directional operators are presented in [3] and [4] respectively, this paper discusses the definition and implementation of the topological operators.

1 Formal specifications of topological operators

Colloquial language is vague and ambiguous when used to describe topological relationships between spatial objects. Since an unequivocal definition is essential for using topological relationships as conditions in a spatial query language, it is necessary to formally specify their semantics.

Topological operators can be applied to use the topological relationship between two spatial objects within a query. Because they return a Boolean value they are also described as *topological predicates*. Topological predicates have two operands: the spatial objects for which the topological relationship shall be tested.

A formal definition of topological relationships is given as follows: Let X and Y be topological spaces. A mapping $f: X \rightarrow Y$ is continuous if for each open subset V of Y the set $f^{-1}(V)$ is an open subset of X. If f is a bijection and both f and f^{-1} are continuous, then f is called a *topological isomorphism*. Topological isomorphisms preserve neighborhood relationships between points during the mapping. Typical isomorphisms are translation, rotation and scaling as well as any combination of these transformations. Topological relationships are those relationships that are invariant under a topological isomorphism.

1.1 Related work

Topological relationships are among the most intensively investigated spatial relationships. The first substantial step towards a formalization of topological relationships was the development of the 4-Intersection Model (4-IM) [5][6]. It is based on a 4-tupel, which records whether the intersections between the *interior* and the *boundary* of the operands are empty or non-empty. Not all of the 16 theoretically conceivable constellations exist in reality. For 2D space, eight different actual relations have been identified and given designations from normal speech, namely *disjoint, touch, equals, inside, contains, covers, coveredBy* and *overlap*.

To resolve topological relationships between line elements more precisely, the 4-IM has been upgraded to the 9-IM by incorporating the exteriors of both operands [6]. The resulting nine intersection sets are recorded in a 3x3 matrix:

$$\mathbf{I} = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{pmatrix}$$

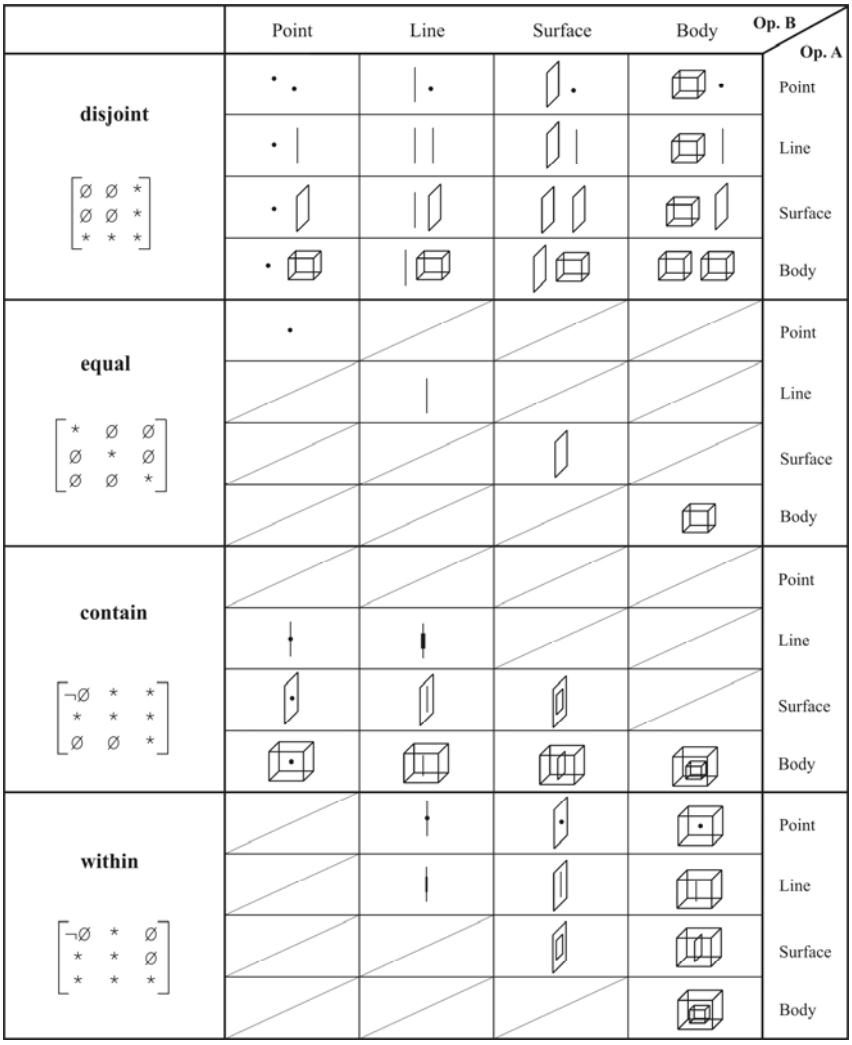

Fig. 1 The topological predicates provided by the Spatial Query Language (part 1).

The 9-IM can also be applied for combinations of spatial objects with different dimensionality [7]. A drawback of the 9-IM is that some topological configurations that are intuitively different result in the same 9-IM matrix while others that are intuitively identical are treated as being different. The first problem is partially solved by the Dimensionally Extended 9-Intersection Model (DE-9IM) which also records the dimensionality of the intersection set [8]. The DE-9IM forms the basis for the formal definitions of topological relationships in the OGC standard [9]. Here, *F* (false) is used in the matrices to denote an empty set, *T* (true) to denote an non-empty set, numbers may be used to define the dimensionality of the intersection set and, additionally, the wildcard (*) may be used at certain places in the matrix that are not relevant for the particular predicate, thereby solving the second of the aforementioned problems. Using this extended set of symbols, the OGC defines the predicates *contains, within, cross, disjoint, equals, intersect, touch* and *overlaps* for arbitrary combinations of (simple) point, line and polygon objects in 2D space.

For the 3D case there are a number of publications with definitions that either use the 9-IM [10][11] or the DE-9IM [12]. Unfortunately, they are unsuitable for application in the AEC domain, because they either rely on cellular decomposition of space or result in a very large number of topological predicates. We were therefore obliged to find our own definitions.

1.2 Definitions

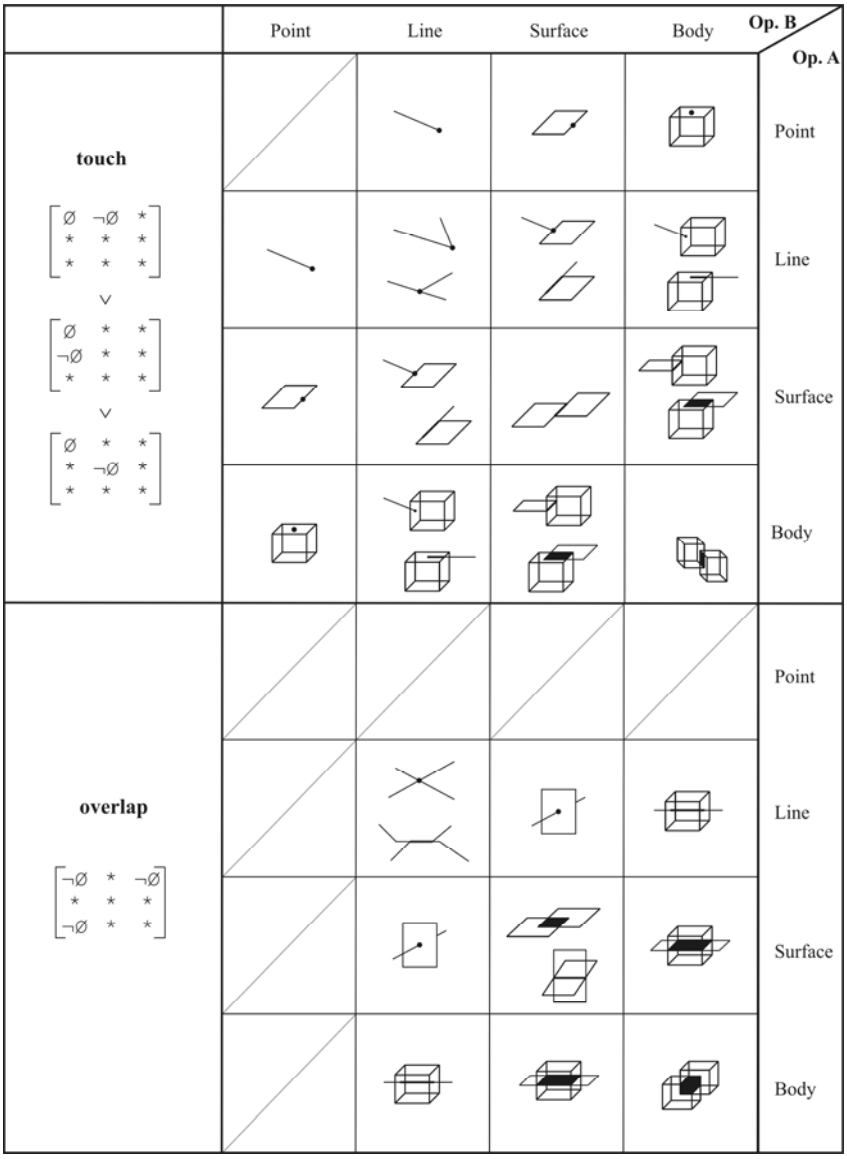

Fig. 2 The topological predicates provided by the Spatial Query Language (part 2).

Because the *dimension* operator cannot be realized by means of the octree implementation technique presented in Section 2, we use the pure 9-Intersection Model instead of the dimensionally extended version here. In order to avoid an unmanageably large number of different topological predicates, we apply the clustering method, as proposed by [13], which allows to place wildcards (*) at those places in the 9-IM matrix that are not decisive for assigning a predicate to a certain constellation. An important pre-requisite for applying the 9-IM is the formal specification of the *interior*, the *boundary* and the *exterior* for each of the 4 spatial types. It has been provided in [1] and [2] and is not repeated here.

Figures 1 and 2 show the topological predicates provided by the Spatial Query Language, present the corresponding 9-IM matrices and illustrate their semantics for different combinations of types by means of pictograms. The given system of topological predicates fulfils the demands of completeness and mutual exclusiveness, i.e. we assign to any topological constellation exactly one of the predicates. This enables the introduction of an additional operator which returns the topological predicate for any given pair of spatial objects. This operator is called *whichTopoPredicate*.

Using the pure 9-IM without the dimension operator

does not allow for any distinction between an *overlap* and a *cross* situation, as proposed in [1]. Nor is it possible to realize the proposed refinements of *touch* (*meet* and *onBoundary*).

There are small differences from the definitions in [13] with respect to the clustering: The predicates *coverBy* and *cover* have not been adopted, because in the application domain considered here, it is normally irrelevant whether only the two operands' interiors overlap or also their boundaries. Accordingly, these two constellations are subsumed under *within* and *contains*, respectively. In addition, we use the designation *touch* instead of *meet* in order to gain a maximum compliance to the OGC standard.

2 Octree-based implementation

2.1 Octree representation

Our implementation technique is based on the octree representation of the spatial objects involved in the topological query. The octree is a space-dividing, hierarchical tree data structure for the discretized representation of 3D volumetric geometry [14]. Each node in the tree represents a cubic cell (an octant) and is either *black*, *white* or *gray*, symbolizing whether the octant lies completely inside, outside or on the boundary of the discretized object. Whereas *black* and *white* octants are branch nodes, and accordingly have no children, gray octants are interior nodes that always have eight children. The union of all child cells is equal to the volume of the parent cell, and the ratio of the child cell's edge length to that of its father is always 1:2. The equivalent of the octree in 2D is called quadtree.

In our implementation concept, each spatial object is represented by an individual octree. There are several different approaches for generating an octree out of the object's boundary representation, most of which are based on a recursive algorithm that starts at the root octant and refines those cells that lie on the boundary of the original geometry, i.e. which are colored gray. For our implementation we use a very efficient creation method developed by Mundani [15] that is based on processing the halfspaces formed by the object's bounding faces. The most important advantage of Mundani's approach for our purposes is that it automatically marks inner cells as black without performing a laborious filling algorithm. As described in the next sections, the existence of black cells is an important prerequisite for the applicability of many rules that make it possible to abort the recursive algorithm at an early refinement level in many situations.

2.2 The recursive algorithm

Due to the limited space available, we can only outline the algorithm here; a more detailed description will be presented in follow-up publications. Furthermore, we restrict the explanation to the algorithm implementing the *whichTopoPredicate* operator.

The recursive algorithm performs a synchronized breadth-first traversal of both octrees. On each level, pairs of octants are created with one octant originating from object A and one octant from object B, both representing the same sector of the 3D space. Each octant pair provides a color combination for the specific rules that can be applied. These rules may lead to filling a 9-IM matrix that is maintained by the algorithm to keep track of the knowledge gained about the topological constellation. Fig. 3 shows an example of a *positive* color combination rule. There are 12 positive and 9 negative rules altogether. A positive rule can be applied when a certain color combination occurs, and a negative rule if certain color combinations do not occur over an entire level. Positive rules lead to *empty set* entries in the matrix, negative rules to *non-empty set* entries.

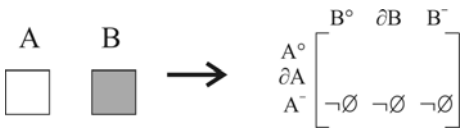

Fig. 3 An example of a positive rule.

The rules are derived from the semantics of the colors. A *white* octant, for example, is part of the *exterior* of an operand, and a *black* octant is part of its interior. If a *white* octant of the first operand occurs at the same place as a *black* octant of the second operand, it follows that the intersection between the *exterior* and the *interior* of the operands is non-empty.

The 9-IM matrix is successively filled by applying these rules for all octant pairs. Each time a new entry is made, the matrix is compared with the matrices of the formal definitions (Section 1.2). If it completely complies with one of these matrices, the recursion is

aborted and the algorithm returns the respective predicate. If there is any divergence between the filled matrix and the matrix of a predicate, the respective predicate is precluded. If no unequivocal decision is possible for any of the predicates, a further refinement is necessary, i.e. octant pairs of the next level are created.

If the algorithm reaches the maximum level, which is defined by the user in advance, and no decision has been made, the so-called predicate hierarchy (Fig. 4.) needs to be applied, i.e. the highest non-disproved predicate is returned. This approach may lead to an "incorrect" predicate being returned, which can be seen as a disadvantage of our algorithm. It may also be interpreted as a way of viewing topological relationships fuzzily: the user is able to define the resolution relevant for the appraisal of topological relationships in his particular application.

Fig. 4 The predicate hierarchy.

3 Summary

This paper has presented formal definitions of topological predicates which have been made available in a 3D Spatial Query Language for Building Information Models and has given an overview on the octree-based implementation technique developed by our group. A detailed discussion of the algorithms will be presented in follow-up publications, as well as an explanation of the integration of the topological operators in the object-relational query language SQL:1999.

References

[1] Borrmann A, van Treeck C, Rank E. Towards a 3D Spatial Query Language for Building Information Models. In: Proc. of the Joint Int. Conf. for Computing and Decision Making in Civil and Building Engineering, Montreal, Canada, 2006.

[2] Borrmann A. Computerunterstützung verteilt-kooperativer Bauplanung durch Integration interaktiver Simulationen und räumlicher Datenbanken [Dissertation]. Technische Universität München, 2007.

[3] Borrmann A, Schraufstetter S, Rank E. Implementing metric operators of a Spatial Query Language for 3D Building Models: Octree and B-Rep approaches. *Computing in Civil Engineering*, submitted.

[4] Borrmann A, Rank E. Specification and implementation of directional operators in a 3D Spatial Query Language for Building Information Models. *Advanced Engineering Informatics*, submitted.

[5] Egenhofer M, Herring J. A Mathematical Framework for the Definition of Topological Relationships. In: Proc. of the 4th Int. Symp. on Spatial Data Handling, 1990.

[6] Egenhofer M. Reasoning about Binary Topological Relations. In: Proc. of the 2nd Symp. on Advances in Spatial Databases, 1991.

[6] Egenhofer M, Franzosa R. Point-Set Topological Spatial Relations. *Int. Journal of Geographical Information Systems,* 1991, **5**(2): 161-174.

[7] Egenhofer M, Herring J. Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases. Technical Report. Department of Surveying Engineering, University of Maine, 1992.

[8] Clementini E, Di Felice P. A Comparison of Methods for Representing Topological Relationships. *Information Sciences - Applications,* 1995, **3**(3): 149-178.

[9] OpenGIS Consortium (OGC). OGC Abstract Specification. 1999.

[10] van Oosterom P, Vertegaal W, van Hekken M, Vijlbrief T. Integrated 3D modelling within a GIS. In: Proc. of the Workshop on Advanced Geographic Data Modelling, 1994.

[11] Zlatanova S. 3D GIS for Urban Development [Dissertation]. Institute for Aerospace Survey and Earth Sciences, Enschede, The Netherlands, 2000.

[12] Wei G, Ping Z, Jun C. Topological data modelling for 3D GIS. Proc. of ISPRS Commission IV Symp. on GIS, 1998.

[13] Schneider M, Behr T. Topological relationships between complex spatial objects. *ACM Transactions on Database Systems,* 2006, **31**(1): 39-81.

[14] Meagher D. Geometric modeling using octree encoding. *IEEE Computer Graphics and Image Processing,* 1982, **19**(2): 129-147.

[15] Mundani R, Bungartz H, Rank E, Romberg R, Niggl A. Efficient Algorithms for Octree-Based Geometric Modelling. *Proc. of the 9th Int. Conf. on Civil and Structural Engineering Computing,* 2003.