Modeling and Managing Topology for 3-D Track Planning Applications

**Edgar Butwilowski, Andreas Thomsen, Martin Breunig,
Paul V. Kuper and Mulhim Al-Doori**

Abstract Topology plays a central role in the modelling and management of n-d geo-information. However, hitherto the standardization of n-d topological data models is still at its beginning. A general model based on oriented hierarchical d-Generalised Maps is proposed to handle topology in object-oriented geo-database management systems. The implementation of the approach is demonstrated as module in DB4GeO, our service-oriented 3-d geo-database architecture. A 3-d tracks planning application example in the city of Munich is presented demonstrating the usage of the n-d topological model. Finally, an outlook on our ongoing 3-d geoinfo research in the Dubai region is given.

Keywords n-d topology · Geo-database · Tracks planning · 3-D planning application

1 Introduction

Navigation through parts of the geometry in 3-d city and infrastructure models is an important issue for analysing such models. Thus topological data models should play a central role in 3-d geo-information applications. However, to our knowledge,

E. Butwilowski (✉) · A. Thomsen · M. Breunig · P.V. Kuper
Karlsruhe Institute of Technology, Karlsruhe, Germany
e-mail: edgar.butwilowski@kit.edu

A. Thomsen
e-mail: andreas.thomsen@kit.edu

M. Breunig
e-mail: martin.breunig@kit.edu

P.V. Kuper
e-mail: kuper@kit.edu

M. Al-Doori
American University in Dubai, Dubai, United Arab Emirates
e-mail: maldoori@aud.edu

M. Breunig et al. (eds.), *3D Geoinformation Science*,
Lecture Notes in Geoinformation and Cartography,
DOI 10.1007/978-3-319-12181-9_3

the combined representation of topology in different spatial dimensions has not been investigated systematically.

In n-d space, a multitude of different representations from simplicial complexes to boundary representations and Constructive Solid Geometry are used (Schaeben et al. 2003; Van Oosterom and Stoter 2010). In this paper, we investigate how a more general concept, namely oriented hierarchical G-Maps, can be used to handle the topology of digital spatial models in a more generic way. The approach is general enough to support 2- and 3-dimensional models, as well as 2-d manifolds in 3-d space. An application example from a tracks planning project is presented.

2 Concept for Modeling N-D Topology

Cellular complexes and in particular cellular partitions of d-dimensional manifolds (d-CPM) are able to represent the topology of an extensive class of spatial objects (Mallet 2002). Based in algebraic topology, they provide a more general, less rigid framework than simplicial complexes. By stepwise aggregation of cells, hierarchies of d-CPM model a succession of generalisations (Köthe 2000). The topology of d-CPM can be represented by d-dimensional Cell-Tuple Structures (Brisson 1993) respectively d-dimensional Generalised Maps (d-G-Maps) (Lienhardt 1991). These possess the combinatorial structure of abstract simplicial complexes, each d-cell-tuple comprising d + 1 cells of different dimension, and related to its neighbours by involution operations. Lévy (1999) has shown that 3-d G-Maps have comparable space and time requirements as the well-known DCEL and radial edge structures, but a much wider range of application, and allow for a more concise and robust code. Lévy introduces hierarchical G-Maps (HG-Maps) for the representation of nested structures. Fradin et al. (2002, 2006) combine G-Maps with semantic information to model architectural complexes in a hierarchy of multi-partitions. G-Maps have been used to represent the topology of land-use changes (Raza and Kainz 1999) and are currently applied in the geoscientific 3-d modeling software GOCAD (Mallet 1992). Circular Incident Edge Lists (CIEL) is another useful data structure for the modelling of geoscientific subsurface data that has been introduced by Lévy et al. (2001).

3 Managing N-D Topology: Implementing Topology in DB4GEO

DB4GeO (Bär 2007; Breunig et al. 2010, 2012), our service-based geo-database architecture, has been designed to support geometric and topological applications in 3-d space. The core geometry model of DB4GeO is based on the model of simplicial complexes. In the model of simplicial complexes (for their application in GIS research see Egenhofer et al. 1989), geo-objects in 2-d and 3-d space are represented by a structure of connected non overlapping simplices, i.e. as triangles and tetrahedra, respectively.

In DB4GeO's implementation, an aggregation of contiguous non-overlapping simplex elements <Simplex>3DElt of the same dimension forms a simplicial complex represented by instances of the classes <Simplex>Net3DComp. The spatial part of a 3-d object may consist of several disjoint simplicial complexes. Therefore, the spatial part of a 3-d object in DB4GeO is represented by the classes <Simplex>Net3D, which may have multiple components. In order to provide efficient access to elements of a 3-d object, an R*-tree or an Octree is used for each net component. It is built up during the construction of the net component. As expected each element is enclosed by a minimum bounding box represented by the DB4GeO class MBB3D.

3.1 Boundary Representation and Level of Detail in DB4GeO

Whereas the inherent structure of simplicial complexes serving to represent manifolds of dimension 0, …, 3 does not require a special treatment of topology, 3-d solids can also be represented as cellular complexes by their cell boundaries: Connected 3-d objects are decomposed into a number of connected cells, the faces, edges and nodes of which are in turn represented by simplicial complexes of lower dimension—triangle nets, polylines and point sets. The topological model of this BRep representation uses a variant of the HGMap structure proposed by Lévy (1999) (see below). In a way this cellular structure can be interpreted as a "generalization" of the non-manifold structure of its underlying components (Fig. 1).

In a similar way, simply connected groups of contiguous cells can be aggregated to a single cell by their common outer boundary, thus yielding a generalized representation of the solid. At each level, the topology of the cellular complex is described by a 3-GMap referencing geometrical boundary representation (Fig. 2). Fradin et al. (2006) provide a framework for the topology of such groups.

Consequently, in DB4GeO a hierarchy of G-Maps grouping comprehensive subsets of simplicial complexes is used to model the topologies of different levels of

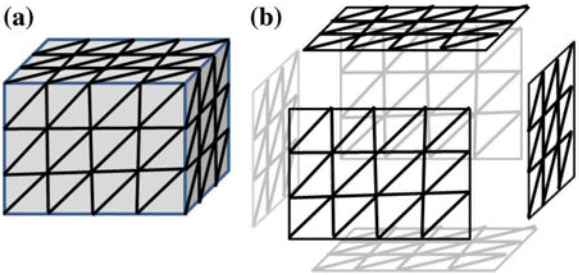


Fig. 1 Boundary representation of a solid cell. **a** 3-D solid, and **b** 2D triangle net faces

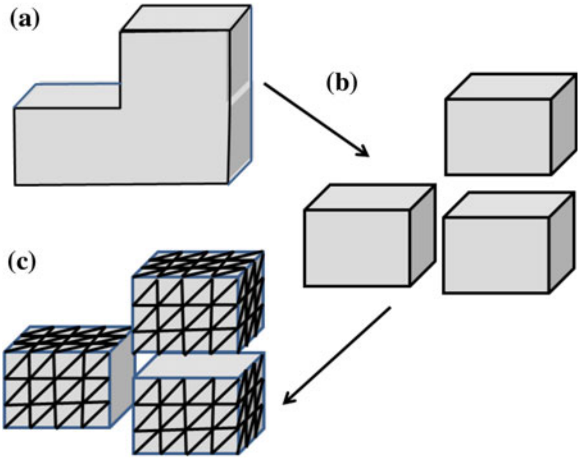


Fig. 2 Aggregation of solid cells. **a** Aggregated cell, **b** Individual 3-d cells, and **c** 2D triangle net cell boundaries

detail in n-d models. DB4GeO supports spatial operations on G-Maps, e.g. the division of a cell by a shortest path that respects the meshing of the underlying simplicial complex.

The topology module of DB4GeO allows to freely navigate along the simple geometries of a complex geometry mesh. Therefore we developed a class model and directly connected it with the DB4GeO geometry kernel API. In our model, the class CellTuple plays a central role for the whole architecture: the CellTuple class is designed as a *composition* of cells for each dimension d, $0 \leq d \leq 3$ (Fig. 3).

A reference to exactly one cell of each dimension, i.e. to a Node, an Edge, a Face, and a Solid, is included. These cell classes are introduced into our model to represent cells that are *not restricted* by the constraints of the *simplicial* classes of DB4GeO: the cells of the G-Maps package need not to be simplices, they rather can be of any shape. Additionally, a CellTuple includes references *alpha*0, *alpha*1, *alpha*2, and *alpha*3 to four other CellTuple objects. These are involution transitions between cell-tuples that are explicitly modelled and stored in attributes of the class

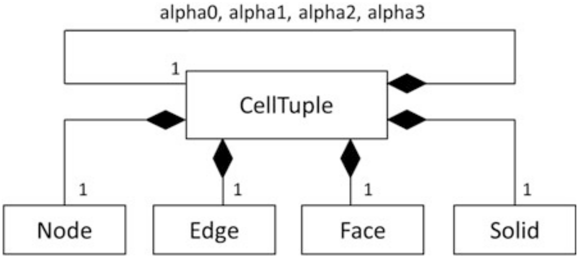


Fig. 3 CellTuple class designed as composition of cell classes

field. Each involution α_i leads to a cell-tuple that is equal to the given cell-tuple with the difference that the cell of dimension *i* is exchanged.

In summary, our CellTuple class has the following definition in Java:

```
public class CellTuple {
  long id;
  Node node;   // reference to 0-cell
  Edge edge;   // reference to 1-cell
  Face face;   // reference to 2-cell
  Solid solid; // reference to 3-cell
  CellTuple alpha0; // alpha 0 involution
  CellTuple alpha1; // alpha 1 involution
  CellTuple alpha2; // alpha 2 involution
  CellTuple alpha3; // alpha 3 involution
}
```

All these references of a CellTuple object are set during a construction process. In the construction process, a cell net is build-up that consists of a set of cells and a set of cell-tuples. In our toolbox, the cells and cell-tuples are created on the basis of a simplicial complex.

The API user first creates a simplicial complex manually or loads data from a file or a network resource into DB4GeO. Then cells (i.e. Node, Edge, Face, and Solid objects) are created on the basis of the geometric and topological primitives of the simplicial complex. The primitive types that are provided by DB4GeO and that are used in this process are Point3D, Segment3D, Triangle3D, and Tetrahedron3D. All these primitives are geometrically embedded by x, y, z coordinates of Point3D class.

The geometric embedding (coordinates) of the simplicial complex is adopted by the cell complex. In a first step, the topological and geometric configuration of the simplicial complex is simply copied onto the cell complex, i.e. a Point3D object becomes a Node object, a Segment3D object becomes an Edge object, a Triangle3D object becomes a Face object and a Tetrahedron3D object becomes a Solid object. After the cell objects have been created, the CellTuple objects are build-up with all their references to the already existing cell objects. Then all missing information that is needed by the cell-tuple structure and that is not already directly available in the simplicial complex—like missing object references—are derived through interpretation of the simplicial complex.

In a second step, the cell complex is reduced and simplified in order to create a basis for further cell complex manipulation (this will be discussed in a section below).

As presented in Breunig et al. (2013a, b), the references between CellTuple objects and Cell objects are bi-directional (Fig. 4).

Each concrete cell class has a *getACellTupleOfCell()* method which returns an arbitrary cell-tuple of the cell (see *anyCellTuple* association in Fig. 4). A "cell-tuple of the cell" means a cell-tuple that vice versa contains the cell. With this model, it is easy to navigate from cell-tuple to cell and back at any point in the code. This is shown in the following example for the use case of iterating along the boundary of a face cell and printing all support points to standard out:

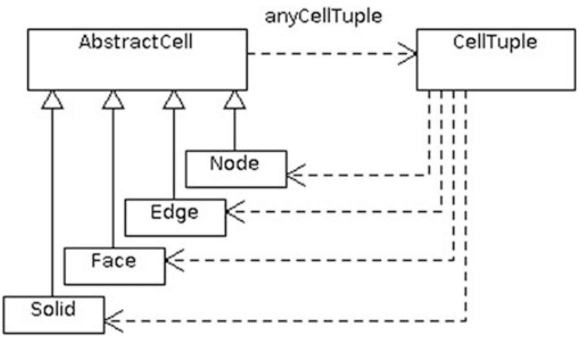
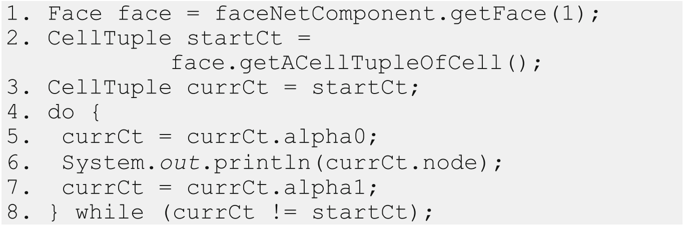


Fig. 4 Bi-directional references between CellTuple and Cell. *Source* (Breunig et al. 2013a, b)



```
Face face = faceNetComponent.getFace(1);
CellTuple startCt =
          face.getACellTupleOfCell();
CellTuple currCt = startCt;
do {
 currCt = currCt.alpha0;
 System.out.println(currCt.node);
 currCt = currCt.alpha1;
} while (currCt != startCt);
```

In this coding session, the API user first queries a face net component for a particular face with the identifier (ID) 1. Proceeding with the returned face, the user receives an arbitrary cell-tuple of the face (*startCt*). The cell-tuple is used as a starting point to initiate a 2-orbit $(\langle \cancel{\alpha_2} \rangle(startCt))$ "around" the face cell [an introduction to "orbits" can be found in Thomsen and Breunig (2007)]. In the do-while-loop, the α_0 and α_1 involutions are iterated rotatory each step. This is done by accessing the *alpha*0 and *alpha*1 field values of CellTuple class. A rotatory iteration of α_0 and α_1 involutions generates a 2-orbit, which navigates step-by-step along the nodes and edges of the face's boundary (Fig. 5).

The orbit stops and the program terminates when the cell-tuple of the current step (*currCt*) is the same as the start cell-tuple (which means the orbit is back to the beginning point).

Cells of the G-Maps package obey the requirements of Java Comparable interface (see Cell interface extending Comparable interface in Fig. 6).

This is helpful since it forces all realizing cell classes to implement a *compareTo ()* method. In our case, the method simply compares the IDs of two cells. Since the *compareTo()* method is realized, all cell objects can be stored in a Java *Set* or *Map* and retrieved efficiently by their ID, which is fast even at a high amount of cells.

The Cell interface specifies some methods that have to be realized by classes that claim to implement the interface. Some methods are specified in order to simplify the navigation on cell level. For example, the *getNeighbour…()* methods have to return all cells of dimension *j* that are neighboring a given cell of dimension *k*. For each dimension *j*, the Cell interface specifies an adapted method with a suitable cell type as the method's return type [e.g. *getNeighbourNodes()* or *getNeighbourEdges()*]. The methods have to return a list of all incident cells in the case that $j \neq k$ and a list of

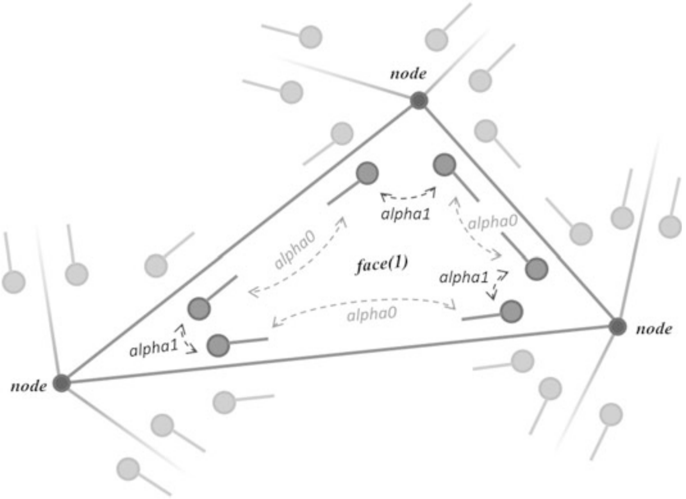


Fig. 5 Example of a 2-orbit (*inside a triangular cell*). The cell-tuples are represented as *darts*: a *dart* positioned at *node(i)* along *edge(j)* within *face(k)* represents the cell-tuple *(node(i), edge(j), face(k))*. The set of all cell-tuples *{(node(), edge(), face(*1*))}* linked by sequences of involutions *alpha*0 and *alpha*1 describes the triangular *face(*1*)*

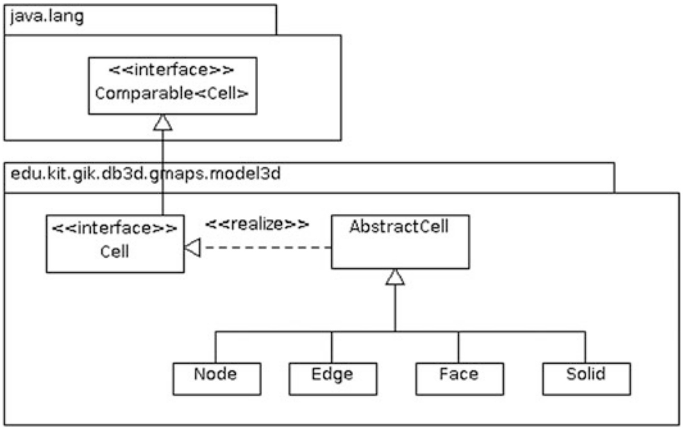


Fig. 6 Class model of cells overview

adjacent cells in the case that *j* = *k*. For example, if invoking the *getNeighbourEdges ()* method on a Face object, then the method returns all edges that are incident to the face cell. In contrast, invoking the *getNeighbourEdges()* method on an Edge object, returns all edges that are adjacent to the given edge. As convenience methods, we also introduce the *isNeighbourOf(cell:Cell)* methods that have to take the cell that is given as the method's parameter value and to check whether the cell is adjacent or incident to the cell that the method is invoked on. Internally, the methods have to determine the cell's dimension and to invoke the appropriate *getNeighbour…()* method on the basis of the cell's dimension.

Similarly, we added *countNeighbour…()* convenience methods to the Cell interface that simply invoke the appropriate *getNeighbour…()* method and return the size of the result list. For example, the *countNeighbourNodes()* method of a Solid object returns the number of support points that the solid consists of, whereas the *countNeighbourSolids()* method of a Solid returns the number of its neighboring solids.

Additionally, the Cell interface demands a *getBoundary()* method that has to return the *d*-1-dimensional boundary cells of the given cell of dimension *d*. Internally, the *getBoundary()* method implementations simply have to invoke the appropriate *getNeighbour…()* method. For example, if invoked on an Edge object, the *getBoundary()* method internally invokes the *getNeighbourNodes()* method in order to receive *d*-1 boundary cells (nodes).

Of the presented methods of the Cell interface, only the *getBoundary()* method is individually realized for each concrete Cell class (Node, Edge, Face, and Solid). All other methods are implemented in an abstract cell class (AbstractCell), since their algorithms are equal for all different cell types. AbstractCell class is the superclass of all concrete cell classes.

The objective for the development of the GMaps package is to provide both advanced (fast and flexible) navigation capabilities inside the already available net structure of the simplicial complex data types of DB4GeO, as well as the possibility to define new cell types that are more general than simplices.

In order to integrate the already available functionality of the DB4GeO kernel with the model of general cell types, we implement the GMaps package as a *two level hierarchy*, where the general cell types are defined as the *object level* and the simplicial complex data types as the underlying *net level* (Fig. 7).

We then extend the CellTuple definition by two field attributes *higher* and *lower* of type CellTuple. These form the type-reflexive, bi-directional references between cell-tuples of lower level of detail (object level) and higher level of detail (net level).

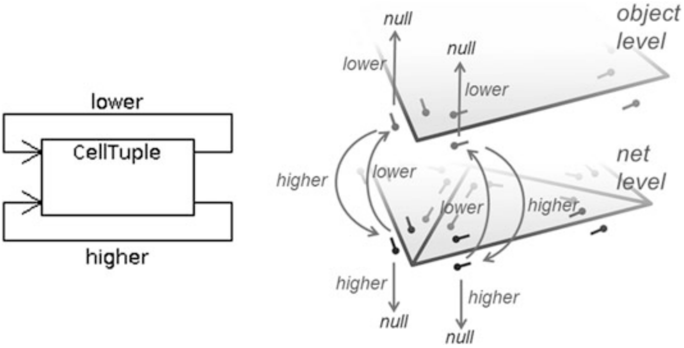


Fig. 7 Cell-tuples of object level and net level are bi-directionally linked by higher and lower references

In DB4GeO, the construction process for G-Maps structure performs in two steps: In a first step, all cell-tuples on net level are created on the basis of the underlying simplicial complex. Second, the object level is derived from the net level. In this step, only the cell-tuples of the components' boundaries are considered: for each cell-tuple at net level boundary, a sibling is created at object level. Each original net level cell-tuple (*ctNL*) is linked to its newly created sibling at object level (*ctOL*), in a way that *ctNL.lower* = *ctOL* and back-referenced (*ctOL.higher* = *ctNL*). If we already are on object level, then the return value of *lower* attribute is *null*. Vice versa, if we already are on net level, then the return value of *higher* attribute is also *null*. By this, we can always check whether we are on object level or on net level by checking *ct.lower* == *null* (object level) and *ct.higher* == *null* (net level). Every cell-tuple at object level has a valid reference to its representation at net level (i.e. *ctOL.higher* == *null* is never the case). Instead, cell-tuples at net level may or may not have a representation at object level, depending on their embedding. If they have no object level representation, i.e. if a certain detail is missing at the object level, then *ctNL.lower* == *null* is true.

The presented concept gives us free navigation capabilities between net level and object level and is flexible enough to be extended by any number of additional levels of detail.

The boundary representation (object level) of a geo-object can be used as a starting point to create additional cells at object level through suitable topological editing operations. The G-Maps package user can insert or delete nodes, edges, and faces. For example, by inserting an edge into a face, the face cell is split into two new face cells. In such case, the G-Maps API takes care of creating all new cell and cell-tuples and creating and maintaining all bi-directional multi-level references. The internal method for edge creation depends on the application purpose and can be exchanged. As a prototype, we implemented an edge creation mechanism on the basis of the shortest path between two nodes of the net level mesh, following triangle boundaries.

With the presented model, it is easy to navigate not only on top of meshed data like simplicial complexes but also to freely navigate between the object level and net level representation of a geo-object. In practice, this can be used for efficient algorithms in hybrid models that integrate e.g. TIN and B-Rep in one application.

4 Tracks Planning Example

4.1 Handling Levels of Detail

The planning of a system of tunnels for underground railways is a complex task with different protagonists such as city planners, civil engineers, geologists etc. However, they share geometry and topology as central parts of their models. Our application example, an ongoing 3-d tracks planning project in the city of Munich, investigated by the research group "Computer-supported Cooperative Planning for

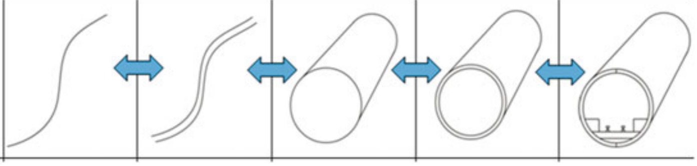
	Level 1	Level 2	Level 3	Level 4	Level 5
Subway Planning / Tunnel section	2D polyline	2D polyline (with width)	3D Tunnel geometry	precise 3D geometry (with inner shell, drainage, separating wall)	precise 3D geometry incl. rails /, traffic lanes, fan equipment, cables, lighting devices, traffic signs
Figure					

Fig. 8 Geometric objects on different LoD in tunnel planning (by André Borrmann, TU Munich)

3-d City and Building Models" (*3DTracks*) (Breunig et al. 2011), demonstrates how topology is used to navigate through 3-d planning models: In subway planning, completely different scales have to be considered—ranging from the scale of several kilometers for the general design of the subway alignment down to centimeter scale for the detailed planning of traffic nodes.

The use of multi-scale models is well established in the Geographical Information System (GIS) domain (van Oosterom and Schenkelaars 1995), and forms an integral part of the respective data exchange standards, such as CityGML for example Kolbe (2008). However, these approaches focus on static models and primarily aim at supporting the visualization of the models. Accordingly, such approaches are less suitable for multi-scale models used in planning processes where frequent modifications result in high dynamics. Currently there is yet no well-established methodology available to ensure and preserve consistency between such dynamic multi-scale models. In a first step, the number of levels-of-detail (LoD) as well as the corresponding geometrical representation on each LoD has to be defined (Fig. 8).

Contrary to the bottom-up definition of levels-of-detail by successive generalizations in GIS, in building planning LoDs are established top-down starting with a general outline and stepwise adding detail. On LoD1, the tunnel is represented by only its main axis (the *track line*), while on LoD5, it is represented by a precise 3-d model including all planning details. The representations on the diverse LoDs result from different detailing demands in the individual planning stages. In a second step, half-automatic transformations between the geometric representations at different LoDs should be provided.

In general, the storage and management of multiple representations of real world entities involve logical redundancies which have to be carefully handled in order to avoid inconsistencies. This applies in particular when model modifications are as frequent as in an ongoing planning process. In the context of the *3DTracks* project, a logical framework has been developed for the multi-representation of tunnel planning models at several LoDs. Two sub-projects of the *3Dtracks* research group at Munich University of Technology have developed a methodology and software system that aims at ensuring consistency across levels-of-detail during synchronous co-operative interactive tunnel planning (Borrmann et al. 2013; Borrmann and Jubierre 2013). A characteristic feature of their level-of-detail hierarchy is the interdependency of semantics, spatial properties and consistency constraints. Hence they employ the combination of a semantic model *SM* with an associated procedural (CSG) geometry model *PM*. The *SM* describes the functions of objects and the dependencies between different model parts, in particular the relationships between LoDs whereas in the *PM*, the progression from lower towards higher LoDs is realized by successively applying geometric operations adding more and more detail.

Borrmann and Jubierre (2013) define 5 levels-of-detail of a tunnel segment (Fig. 8): LoD1: track line, LoD2: tunnel volume as a compact solid, LoD3: tunnel wall, LoD4: tunnel wall with track bed, LoD5: Tunnel with full equipment. A number of constraints linking the objects ensure spatial consistency: the inner "hollow" volume of the tunnel must be completely contained within the volume defined by the outer hull, contiguous tunnel parts must exactly fit together etc. Because of such constraints and other dependency relations, the directed graph model of LoDs consists of a stratified tree augmented at each LoD by links between the nodes of different branches.

In the hybrid model of tunnel planning, the LoD hierarchy is established within the semantic model *SM* and carries over to the spatial procedural model *PM* by the application of a combination of spatial operations and spatial constraints. A closer look shows that semantic, topological and geometrical aspects cannot be strictly separated, even though each aspect follows its own rules: e.g. the semantics of a tunnel require, that contiguous tunnel parts fit together at all LoDs, hence their topological models must be connected, and the geometries of their front parts must match. Parts of the tunnel equipment at LoD5 like e.g. railway signals are topologically disconnected separate units the geometrical position of with however is determined by engineer standards.

For application purposes—like augmented reality visualization (cf. Urban et al. 2013) and integration with city models (cf. Steuer et al. 2013)—from the *PM* a corresponding explicit representation *EM* at different LoDs is derived by means of CAD software (Fig. 9). The transformation into an explicit representation *EM* of the model geometry does only rudimentarily preserve these logical dependencies. Instead we must try to reconstruct them by means of the references between semantic and procedural model parts and their explicit representation. From a DBMS point of view, these structures and dependencies are to be taken for granted and must be represented as such in the hybrid database.

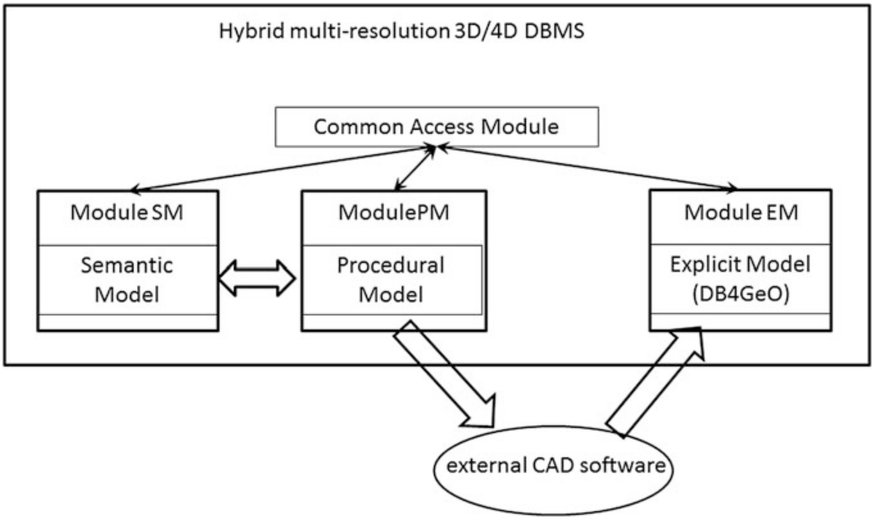


Fig. 9 Modifications must take place on the semantic *SM* and procedural *PM* representations. External CAD software (AutoDesk Inventor) is used to transform the updated version into a new explicit model. There is no feedback of modifications from *EM* to *PM* and *SM*

4.2 A Hybrid Multi-representation 3D/4D DBMS for the 3DTracks Tunnel Planning Project

This leads to the concept of a hybrid, multi-representation 3D/4D database for the management of different temporal versions of 3-d geometry models with associated semantics and derived explicit representations at different levels of detail (Fig. 10). Its data storage comprises four different components—*SM* for model semantics and dependencies, *PM* for its procedural (CSG) geometry, *EM* for an explicit representation, *AM* for additional information (annotation, comments etc.). A common *access module* supports navigation across the whole database, and delegates elementary queries stemming from the query module, to the corresponding storage modules.

From the way the data are generated, it follows that there are strong dependencies between the different representations. In particular, the *EM* is completely dependent on *PM* and *SM,* and there is no feed-back from *EM* to *PM*. In consequence, all updates of the data set must first take place on *SM* and *PM*, and then be carried over to *EM* (Fig. 9). For the implementation of the *EM* component, an adapted version of DB4GeO is to be used. At each level of detail, the topology of the explicit representation of the model parts on the object level consists of cellular complexes in BRep referencing the 2-d simplicial complexes on the net level (Fig. 11).

For the navigation on the hybrid database, the access module uses a graph of the dependencies derived from the *SM* and *PM*, and enhanced by spatial information from the *EM* like e.g. bounding boxes and by thematic information from the *AM* like material, costs etc.

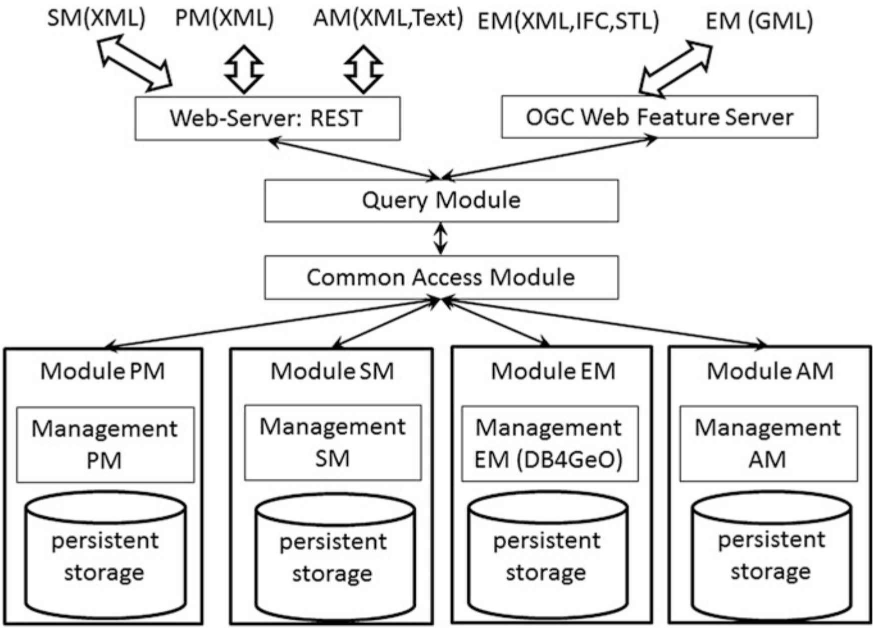


Fig. 10 Architecture of a hybrid spatio-temporal database. Each model representation is stored in a separate database module (*PM* procedural, *SM* semantic, *EM* explicit, *AM* annotation). Integrated access is provided by a common access module

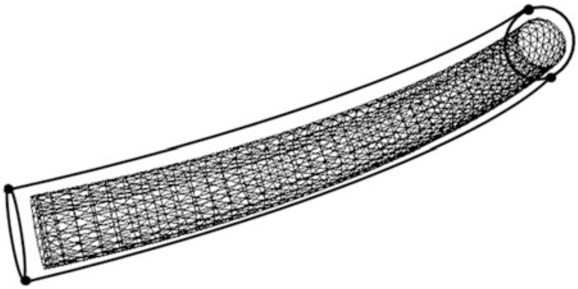


Fig. 11 Explicit representation of the topology of the outer hull of a tunnel part at LoD1. For the topology at the object level, a cellular complex of only two faces is sufficient referencing two triangle nets at net level

Whereas the hierarchy of the Levels of Detail is defined within the semantic model *SM* by references and cross-references between objects of *SM* and the objects of *PM* and *EM*, its logical structure carries over to these representations (Fig. 12).

To support the navigation across different LoDs, there are essentially two ways of handling dependencies between different representations of a spatial object.

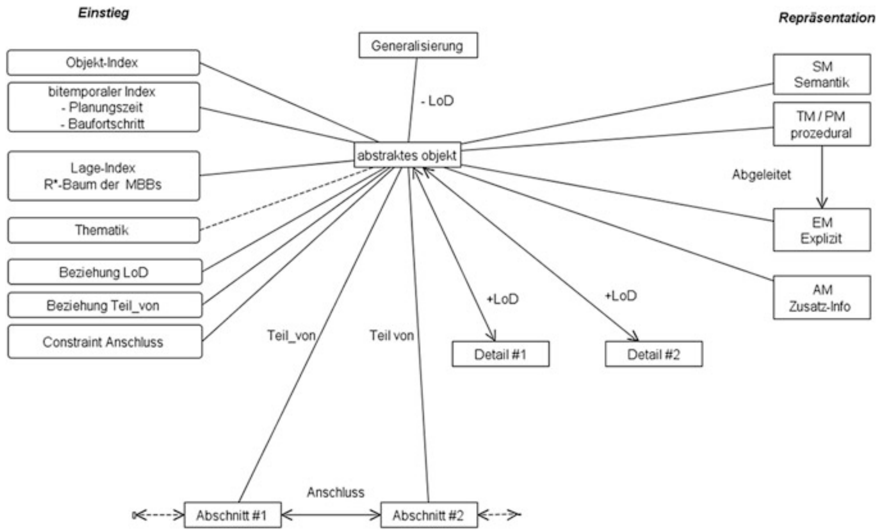


Fig. 12 Multi-representation of a spatial object in the hybrid database. An abstract object references semantic (*SM*), procedural (*PM*), explicit (*EM*) representation and annotation (*AM*), at different levels of detail. Relationships and constraints describe dependencies between objects. Access is provided by different spatiotemporal and thematic indices

On the one hand, a hierarchy of LoDs can be established ad hoc by applying a combination of well defined generalization resp. specialization operations to an explicit spatial model. An important question is whether such operations are reversible, and whether they are Eulerian—i.e. preserve essential topological properties (Mäntylä 1988; Thomsen and Breunig 2007). A particular example of reversible generalization of triangle nets is the *progressive meshes* method (Hoppe 1996). On the other hand, representations at different LoDs can be produced separately, and afterwards linked together by explicit references (Haunert and Sester 2005; Anders and Bobrich 2004). The two approaches can be combined. In DB4GeO, a close relationship between topologies at different LoDs is provided by the explicit linking of cell-tuples, following Lévy (1999) and Fradin et al. (2002). If we try to apply this approach to the *EM* component of the hybrid DBMS, however, the question arises whether the two LoD hierarchies of *SM/PM* and *EM* can be made compatible (Fig. 13). If this were the case, navigation across LoDs on the explicit part of the DBMS might be enhanced by topological and geometrical operations, instead at each step passing by cross-references to the semantic model and back again. However, as the two hierarchies are established independently, it seems to be impossible to achieve compatibility, because of the unidirectional dependency of *EM* on *SM/PM*. The only generalization operation that can be performed independently on the EM is a change of the mesh densities of the boundaries elements, as this parameter is not controlled by the semantic and the procedural model. Mesh density, however, does not appear in the definition of

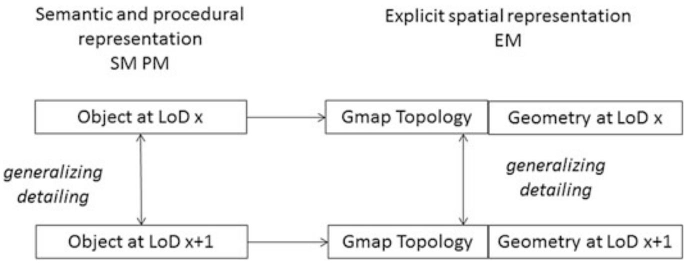


Fig. 13 Combination of semantic and spatial LoD hierarchies

LoDs mentioned above. Therefore, for the navigation on LoDs, the access module uses the dependencies from the *SM* and *PM*, combined with references to the objects of the *EM*. It has yet to be investigated if any gain in efficiency can be obtained by establishing links between LoDs within *EM* that are completely controlled by the dependencies within the semantic and procedural models.

5 Conclusion and Outlook

In this paper we have investigated an approach for the modelling and management of n-d topology in geo-database management systems. As an implementation example we demonstrated topology classes and their relationships to geometry classes of DB4GeO, our service-oriented geo-database architecture. Finally we applied the introduced topology approach to topological objects of a 3-d tracks planning example in the city of Munich, Germany.

In our future research work we plan to use new application examples in the Dubai region that is generating a strong requirement for geospatial services for above- and sub-surface models for applications such as 3-d city modelling, 3-d tracks planning, oil exploration, etc.

Problems to be solved in the region are water sources seriously threatened by pollution, agriculture intensification, decline of urban settlements and transport networks, soil sealing and fragmentation of landscape. A 3-d model covering a vast area of the Emirates, including the city itself and neighbouring territories to north, west and south, will be able to create new kinds of database queries helping to understand and analyse urban and sub-surface planning.

Apart from conventional 3-d applications there is also an increasing demand of mobile systems. For instance in May 2013 the vice-president and prime minister of the UAE and ruler of Dubai, Sheikh Mohammed bin Rashid Al Maktoum, announced the “Mobile Government” initiative supporting the development of mobile systems. In this context mobile GIS are of prime importance e.g. the GIS department at Dubai Municipality is currently working on mobile applications like Dubai Cadastral, Dubai Map und Dubai Utilities Map.

Until now most of the mobile GIS applications are handling 2-d data following the Simple Feature guidelines. But the applications capable of handling 3-d geodata e.g. sub-surface or city models are on the rise. The area of augmented reality, especially in the context of Smart Cities has enormous potential.

At the Karlsruhe Institute of Technology (KIT) an Android based augmented reality viewer was developed in 2013 which is capable of presenting 3-d building models (Breunig et al. 2013a, b). For the further development in this research area still many challenges need to be solved. Precise location management, limited processing power and limited memory of the mobile devices are just some of the problems which must be resolved.

The long-term goal is to contribute to the enhanced planning, designing and development processes being undertaken as part of the Dubai Vision for the next two decades.

Acknowledgments This work is funded by the German Research Foundation (DFG) within the Research group 1546 "Computer-Aided Collaborative Subway Track Planning in Multi-Scale 3D City and Building Models", grant no. BR 2128/14-2.

References

Anders K-H, Bobrich J(2004) MRDB approach for automatic incremental update. In: ICA workshop on generalisation and multiple representation, Leicester

Bär W (2007) Management of geoscientific 3D data in mobile database systems. In: German. Ph. D. thesis, University of Osnabrück, Germany, 166 p

Borrmann A, Jubierre JR (2013) A multi-scale tunnel product model providing coherent geometry and semantics. In: Proceedings of the 2013 ASCE international workshop on computing in civil engineering, Los Angeles

Borrmann A, Flurl M, Jubierre J, Mundani R-P, Rank E (2013, submitted) Synchronous collaborative tunnel design based on multi-scale infrastructure models. Submitted to advanced engineering informatics. Elsevier,Amsterdam

Breunig M, Schilberg B, Thomsen A, Kuper PV, Jahn M, Butwilowski E (2010) DB4GeO, a 3D/4D geodatabase and its application for the analysis of landslides. In: Geographic information and cartography for risk and crisis management. Lecture notes in geoinformation and cartography. Springer, Berlin, pp 83–102

Breunig M, Hinz S, Menninghaus M, Vögtle Th, Rank E, Schilcher M, Borrmann A, Mundani R-P, Ji Y, Donaubauer A, Steuer H (2011) Towards computer-aided collaborative subway track planning in multi-scale 3d city and building models. In: Proceedings 3D geoInfo 2011, Wuhan, 9 p

Breunig M, Butwilowski E, Golovko D, Kuper PV, Menninghaus M, Thomsen A (2012) Advancing DB4GeO. In: Proceedings 3D Geoinfo 2012, Quebec, Canada. Lecture notes in geoinformation and cartography. Springer, pp 193–210

Breunig M, Butwilowski E, Kuper PV, Golovko D, Thomsen A (2013a) Topological and geometric data handling for time-dependent geo-objects realized in DB4GEO. In: Timpf S, Laube P (eds) Advances in spatial data handling. Advances in geographic information science. Springer, Berlin, pp 1–13

Breunig M, Kuper PV, Dittrich A, Wild P, Butwilowski E, Al-Doori M (2013b) Design, implementation and applications of 3d web-services in DB4GEO. ISPRS Ann photogrammetry Remote Sens Spat Inf sci II-2/W1:87–94. doi:10.5194/isprsannals-II-2-W1-87-2013

Brisson E (1993) Representing geometric structures in d dimensions: topology and order. Discrete Comput Geom 9:387–426

Egenhofer MJ, Frank AU, Jackson JP (1989) A topological data model for spatial databases. In: Proceedings ACM SIGMOD, LNCS 409. Springer, Berlin

Fradin D, Meneveaux D, Lienhardt P (2002) Partition de l'espace et hiérarchie de cartes généralisées. In: AFIG 2002, Lyon, décembre 2002, 12 p

Fradin D, Meneveaux D, Lienhardt P (2006) A hierarchical topology-based model for handling complex indoor scenes. Comput Graph Forum 25(2):149–162

Haunert J-H, Sester M (2005) Propagating updates between linked datasets of different scales. In: Proceedings of XXII international cartographic conference, A Coruña, Spain, 11–16 July

Hoppe H (1996) Progressive meshes. In: ACM SIGGRAPH 1996, pp 99–108

Kolbe TH (2008) Representing and exchanging 3D city models with cityGML. In: Lee J, Zlatanova S (eds) Proceedings of the 3rd international workshop on 3D geo-information, Seoul, Korea. Lecture notes in geoinformation and cartography, Chap. 2, Springer, Berlin

Köthe U (2000) Generische Programmierung für die Bildverarbeitung. Dissertation, FB Informatik, Universität Hamburg, 274 p

Lévy B (1999) Topologie Algorithmique: Combinatoire et Plongement. Ph.D. thesis, INPL Nancy, 202 p

Lévy B, Caumon G, Conreaux S, Cavin X (2001) Circular incident edge lists: a data structure for rendering complex unstructured grids. In: Proceedings of visualization 2001, (VIS'01). IEEE, San Diego, pp 191–557

Lienhardt P (1991) Topological models for boundary representation: a comparison with n-dimensional generalized maps. Comput Aided Des 23(1):59–82

Mallet JL (1992) GOCAD: a computer aided design programme for geological applications. In: Turner AK (ed) Three-dimensional modelling with geoscientific information systems, NATO ASI 354. Kluwer Academic Publishers, Dordrecht, pp 123–142

Mallet JL (2002) Geomodelling. Oxford University Press, Oxford, 599 p

Mäntylä, M. 1988. An Introduction to Solid Modeling. Computer Science Press, 401 p

Raza A, Kainz W (1999) An object-oriented approach for modelling urban land-use changes. ACM-GIS 1999:20–25

Schaeben H, Apel M, vd Boogart G, Kroner U (2003) GIS 2D, 3D, 4D, nD. Informatik Spektrum 26(3):173–179

Steuer H, Donaubauer A, Kolbe TH, Flurl M, Mundani R-P, Rank E (2013) Planning inner-city-railway-tracks: dynamic integration of geospatial web services in a collaborative multi-scale geometry modelling environment. In: Proceedings of 20th international workshop: intelligent computing in engineering, Vienna, 1–3 July

Thomsen A, Breunig M (2007) Some Remarks on Topological Abstraction in Multi Representation Databases. In: Popovich VV, Schrenk M, Korolenko KV (eds) Proceedings of 3rd international workshop on information fusion and geographical information systems IF&GIS-07, St. Petersburg. Lecture notes in geoinformation and cartography. Springer, Berlin, pp 234–51

Urban S, Leitloff J, Wursthorn S, Hinz S (2013) Self-localization of a multi-fisheye camera based augmented reality system in textureless 3D building models. ISPRS Ann Photogramm Remote Sens Spatial Inf Sci II-3/W2:43–48

Van Oosterom P, Schenkelaars V (1995) The development of an interactive multi-scale GIS. Int J Geogr Inf Sys 9(5):489–507

Van Oosterom P, Stoter JE (2010) 5D data modelling: full integration of 2D/3D space, time and scale dimensions. In: GIScience 2010, pp 310–324