Derivation of 3D Indoor Models by Grammars for Route Planning

Early Draft

Gerhard Gröger & Lutz Plümer, Bonn

Keywords: 3D city models, spatial grammars, CityGML, indoor route planning

Summary: This paper presents a method to generate three-dimensional indoor building models and corresponding route graphs which are suitable for indoor route planning. The concept of attributed grammars is adapted to generate the indoor model which meets the two essential preconditions for the automatic derivation of connectivity information for route graphs: consistency, which enables the identification of neighbouring spaces, and the representation of semantics. The first precondition is met by implementing a model that has been developed earlier and which provably assures consistency between geometry and topology. This gives a contrast to earlier approaches for generating buildings by grammars. Topology is represented by constraints generated by the grammar rules and is maintained by constraint reasoning methods. Semantic aspects being relevant for deriving the connectivity relation been spaces are represented according to *CityGML*. The generation of the indoor model by grammar rules requires only a small number of observations, and the derivation of the route graph from the indoor model is accomplished automatically.

Zusammenfassung: *Herleitung dreidimensionaler Innenraummodelle mit Grammatiken zur Planung von Wegen.* Dieser Artikel beschreibt ein Verfahren zur Herleitung dreidimensionaler Innenraummodelle für Gebäude und der zugehörigen routingfähigen Graphen, die für die Planung von Wegen in Gebäuden geeignet sind. Das Konzept der attributierten Grammatiken wird für die Erzeugung des Innenraummodells angepasst. Dieses Modell erfüllt die beiden entscheidenden Bedingungen zur Herleitung der Erreichbarkeitsinformationen für den routingfähigen Graphen: Konsistenz, die die Detektion benachbarter Räume ermöglicht, und die Repräsentation der Semantik. Die erstgenannte Bedingung ist durch die Nutzung eines bereits früher entwickelten Modells erfüllt, das die Konsistenz zwischen Geometrie und Topologie nachweisbar sicher stellt, im Gegensatz zu bisherigen Ansätzen zur Erzeugung von Gebäudemodellen mit Grammatiken. Die Topologie wird durch Constraints repräsentiert und durch Schlussfolgerungsmechanismen konsistent gehalten. Semantische Aspekte, die für die Herleitung der Erreichbarkeitsrelation zwischen Räumen relevant sind, werden in Anlehnung an *CityGML* repräsentiert. Die Erzeugung des Innenraummodells erfordert nur eine kleine Menge an Beobachtungen, und der routingfähige Graph wird automatisch aus dem Innenraummodell hergeleitet.

1 Introduction

Navigation and route planning are important applications in geo-information science. Whereas systems for outdoor navigation – car navigation, pedestrian navigation, etc. – are widely used nowadays, indoor navigation is still an active research area, e.g. PU & ZLATANOVA (2005) or BECKER ET AL. (2009). Indoor navigation is more difficult, mainly for two reasons: First, positioning as prerequisite for navigation is more difficult indoor, since positioning systems like GPS which are used outdoor very successfully do not work indoor. Second, the spatial data required for navigation and route planning is mostly not available for indoor scenarios (e.g. BECKER et al., 2009). This paper deals with the second problem.

For all route planning problems, indoor in rooms, floors and staircases as well as outdoor on streets, paths and places, a structure which represents connectivity and reachability explicitly as graph is an essential prerequisite. Such structures are called *route graphs*. In an outdoor environment, the edges of the graph typically represent street or path segments, and the nodes crossings or junctions where the segments meet. In an indoor environment, (parts of) rooms, hallways or staircases are modeled by nodes, while edges represent the reachability between the corresponding objects. In addition, route graphs store information relevant for route planning, like segment length, travel time for a segment, or constraints when a segment may be passed only in one direction, e.g. in a one-way street, or may not be passed by specific individuals, e.g. by handicapped persons, or by huge vehicles like trucks. Well known path finding methods like the algorithm of DIJKSTRA (1959) or its more efficient extensions, e.g. the A^* algorithm (NORVIG & RUSSELL, 2003), may be employed to derive optimal paths from route graphs.

In order to automatically derive route graphs from indoor building models, those models must meet two demands: First, *adjacency*, the immediate neighborhood between rooms, floors or stair cases, must be represented correctly, since adjacency is one precondition for the derivation of an edge of the route graph representing the reachability between rooms. The correctness of the adjacency relation is guaranteed only if the building model is geometrical-topologically consistent (GRÖGER & PLÜMER, 2010).

Whereas adjacency is a *necessary* condition for reachability between two rooms or between indoor and outdoor, it is not a *sufficient* one: Two adjacent rooms are in a reachability relation, only if an opening or a door connects both. Hence, the adequate representation of *semantics* of the components of the building model is crucial for the derivation of route graphs. A model representing the semantics of buildings and indoor components is *CityGML* (GRÖGER et al., 2008; KOLBE et al., 2008), which addresses the aspect of indoor modeling, but does not solve the problem of geometric-topological consistency. This is not the intention CityGML was developed for, since it claims to model spatial base data regardless of specific applications. Consistency as aspect of data quality (GUPTILL & MORRISON, 1995) is defined as ‘suitability for a specific purpose’ (ISO TC 211, 2002), hence it often can be defined only in the context of a specific application.

A model guaranteeing the geometric-topological consistency provably, particularly for buildings and their interior structures, is defined in GRÖGER & PLÜMER (2010), see also section 3. Methodically, this model is one essential base of the concepts presented in this paper.

The generation of 3D city and building models is also the intension of *spatial grammars*, particularly of *split grammars* (WONKA et al., 2003; MÜLLER et al., 2006). These constitute the second methodological base of the concepts presented in this paper. Split grammars model buildings by regular geometrical objects like cubes, but the problem of geometric-topological consistency is not addressed: these approaches do not claim to assure consistency. Penetrations of solids are not excluded explicitly, and in fact occur by rule applications.

The main contribution of this paper is a method to generate geometric-topological consistent indoor models, which allow for the derivation of route graphs. We show that the split grammars presented by WONKA ET AL. (2003) and MÜLLER ET AL. (2006) can be rephrased as attributed grammars and extend these approaches by providing an explicit representation of topology and by guaranteeing consistency. Our rules are a special case of the transaction rules in GRÖGER & PLÜMER (2009) and GRÖGER (2006).

For consistency, route planning is used as a benchmark: it poses a big challenge, since an existing path must be identified by the method, and since a derived path must in fact be passable. This paper links the topics geometry, topology and semantics: geometry and topology produce the route graph, which is the base for escape route planning. However, this structure is not sufficient, since semantics must be considered in addition.

The rest of this paper is organized as follows: In the second section, related approaches to indoor route planning as well as on the use of formal grammars for the derivation of 3D models are recapitulated. The third section introduces a geometric-topological 3D model for buildings, which is suited for the derivation of indoor models and for the representation of indoor reachability due to its consistency. The semantic aspects of indoor models are discussed in the next section by introducing CityGML, which enables the representation of objects classes that are relevant for indoor route planning. The fifth section presents a grammar which generates geometric-topologically consistent, semantic indoor models from a minimal set of observations. The derivation of route graphs is supported by that model. This paper ends with concluding remarks and a discussion of open questions and future work.

2 Related Work

The problem of representing and deriving route graphs which are suitable for planning routes in buildings has been in discussion for about two decades, e.g. CHALMET ET AL. (1982), HOPPE & TARDOS (1995), LEE (2001), or PU & ZLATANOVA (2005), but has only partially been solved. LEE & KWAN (2005) introduce a graph called *Combinatorial Data Model (CDM)*, whose nodes represent rooms, while the edges describe the reachability relation between rooms. This relation is defined by the concept of Poincaré Duality (HATCHER, 2001), which is based on the topological 'meets' relation provided by Egenhofer's 4-intersection model (EGENHOFER & HERRING, 1991). The embedding of the CDM in 3D space, called *Geometric Network Model (GNM)*, in addition enables the representation of metrical properties and multiple routes between two rooms. These approaches focus on the representation of graphs, but do not solve the problem of deriving those graphs from 3D models of the interior of buildings.

LEE & ZLATANOVA (2009) and MEIJERS ET AL. (2005) extend the GNM concept geometrically by considering paths composed of edges, which are derived by a median axis transformation from ground plans of rooms or storeys. Such a graph embedded in the 2D plane is derived for each floor of a building, and these plans are then linked by employing 2D overlay methods. These approaches define graphs suitable for route planning and its derivation from 2D floor plans, but do not provide any procedure how to derive such graphs from a 3D building model automatically and effectively. A pragmatic approach to tackle this problem (MEIJERS et al., 2005) is based on a procedure (VAN TREECK & RANK, 2004) to correct geometric-topological errors (e.g. mutually penetrating faces, gaps between faces or neighboring solids) by merging polygons, line segments or points, which have a distance smaller than a certain threshold. This method, which generalizes 2D methods used in commercial GIS like *ArcInfo*, however, lacks a method to check whether the resulting structure is consistent.

In procedures to derive route graphs from indoor models, semantics is considered only in a few approaches. MEIJERS ET AL. (2005) classify vertical polygons separating two rooms according to its reachability function. They differentiate between 'always passable', 'only with key passable', 'only in emergency case passable', 'bidirectional' or 'unidirectional passable'. ANAGNOSTOPOULOS ET AL. (2005) define an indoor ontology and extend the nodes and edges of route graphs by corresponding labels. A procedure to generate labeled graphs from 2D ground plans is drafted roughly.

LORENZ ET AL. (2006) propose the decomposition of rooms into *cells* to reflect that parts of a room differ with regard to reachability. Each cell is represented by a node in the route graph. This approach is restricted to a 2D representation of rooms and cells, and no procedure to automatically derive cells from rooms is given. BECKER ET AL. (2009) extend the GNM concept of LEE & KWAN (2005) as well as the cell decomposition by defining distinct models, called layers, for orthogonal indoor aspects with regard to escape route planning (room topography/topology, sensor transmission ranges). Each layer is represented separately, both as 3D model and corresponding graph. The spatial interaction of cells from different layers is represented explicitly by links between the corresponding nodes.

For deriving models of buildings, particularly for facades, outer hulls or interior structures, *formal grammars* (CHOMSKY, 1959) play a crucial role. In a spatial context, the special case of a *context free* or *type 2 grammar* is relevant. Such a grammar $G = (T, N, S, P)$ is composed of five components: T is the set of *terminal symbols*, N the set of *non terminal symbols* ($T \cap N = \emptyset$), $S \in N$ is the *start symbol* and P is the set of *rules*. The elements of P have the structure $A \rightarrow X$, where A is a non terminal and X is a sequence (a string) of terminal and non-terminal symbols. The *language* generated by the grammar is the set of strings which consists of terminal symbols only, and which is obtained by subsequent applications of rules – replacement of a non-terminal A by the right-hand-side X of the corresponding rule – starting with S. In an *attribute grammar* (ALBLAS & MELICHAR, 1991), a set of attributes is attached to the terminal as well as to the non terminal symbols. The value of an attribute is computed by a *semantic rule*, accompanying a grammar rule from P. If an attribute value attached to a (terminal or non-terminal) symbol on the right-hand-side of a grammar rule is computed by the values of the attributes on the left-hand-side, the attribute is called *inherited*. Vice versa, it is called *synthesized*, if a value attached to a left-hand-side symbol is derived from a value of a right-hand-symbol.

A grammar for the generation of spatial objects, called *shape grammar,* was developed by STINY & GIPS (1972) for architectural purposes. Such grammars operate on a purely geometrical level, not on a symbolical level of non-terminals, complicating the design of the rules. The *split grammars* presented by WONKA ET AL. (2003) can be interpreted as attribute grammars, which are used to generate building facades. The appearance of the facades and the design process are controlled by attributes attached to terminal or non terminal symbols. MÜLLER ET AL. (2006) extend this approach by generating the complete outer hull of buildings. Their grammar, which is called *CGA Shape*, contains rules which split a geometry shape, e.g. a polygon, into a sequence of polygons, according to a given ratio. An example for a split rule is given by

$$fac \rightarrow Subdiv(\text{'Y'}, 3.5, 0.3, 3, 3, 3)\ \{floor \mid ledge \mid floor \mid floor \mid floor\}$$

which splits the non terminal polygon *fac* into five parts along the y-axis, each having the length 3.5, 0.3, 3, 3, or 3, and the type *floor*, *ledge*, *floor*, *floor* or *floor*, respectively. To avoid occlusions of doors or windows by other parts of the building, a geometrical occlusion test is provided. Symmetry of the facades is achieved by introducing *snap lines*. The aim of the grammar of MÜLLER ET AL. (2006) is the generation of artificial spatial objects for movies or computer games, not the reconstruction of existing urban objects. The approach focuses on the generation of facades, not on interior building structures. The consistency of these models, however, is not intended by the grammar and achieved at most incidentally: there is no concept of sharing of common walls, and penetrations of objects are not prohibited. For the purpose of reconstruction, grammars are used by BRENNER & RIPPERDA (2006) and by MÜLLER ET AL. (2007) for facades, by DÖRSCHLAG ET AL. (2008) for buildings with detailed roof structures, and by SCHMITTWILKEN ET AL. (2007) for different types of stairs.

3 3D models being geometric-topologically consistent

An essential prerequisite for the automatic derivation of route graphs from interior building models is consistency, particularly with regard to the neighborhood between rooms, but also between the outer hull and the interior of the building. Two neighboring rooms (solids) are not allowed to penetrate mutually, and gaps separating both solids are prohibited as well. The first requirement may be met by sharing common faces, while the second is guaranteed by a complete covering of the building's interior by solids. Both requirements constitute a *three-dimensional tessellation* of the interior of a building.

A model which meets both requirements and therefore is geometrical-topologically consistent is defined in GRÖGER & PLÜMER (2010). It is based on the concept of *cell complexes* (HATCHER, 2001), which essentially requires that nodes, edges, faces and solids are either pair-wise disjoint or touch in their common boundaries. In the latter case, both touch in a common face, edge or node. To be more specific, when two solids touch, then they touch in a common face, if two faces touch, they touch in a common edge, and if two edges touch, they touch in a common node. A second condition is that 3D space is completely covered by solids, without any voids or hollow spaces not occupied by solids. Hence, two special solids are introduced: a solid representing the air space (which is bounded partially by the terrain and is not bounded from above) and one representing the earth's mass (which is bounded partially by the terrain and is not bounded from below). Hence, solids constitute a complete coverage of 3D space. An UML class diagram (BOOCH et al., 2005) of the model is depicted in Fig. 1.

The model described so far defines consistency from a mathematical point of view. However, it does not provide a method to effectively and efficiently check whether a data set meets these requirements: It is not effective, since the complete covering by solids is not immediately checkable, and not efficient since the pair-wise test for penetrations or intersections is very expensive from a computational point of view. In GRÖGER (2006) and GRÖGER & PLÜMER (2010), this problem is solved by introducing so-called *axioms*, which are provably equivalent to the mathematical model described above, while being effectively and efficiently implementable. The pair-wise checking of penetrations and intersections is restricted locally to faces and edges in the boundary of a solid, hence avoiding the pair-wise consideration of all faces and edges.

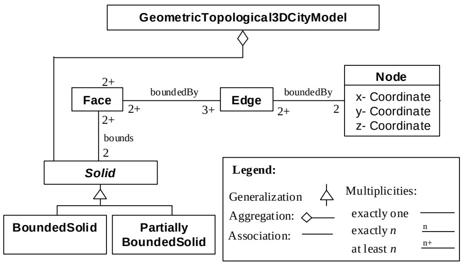


Fig. 1: UML-Diagram of the geometric-topological 3D model of GRÖGER & PLÜMER (2010)

4 Representation of semantics

The geometrical-topological model presented in the last section is the base for defining semantic objects (often called *features*) which are relevant for modeling indoor structures and for route planning. A semantically very rich model for cities and urban objects is *CityGML* (GRÖGER et al., 2008; KOLBE et al., 2005, 2008), which is based on the *Geography Markup Language GML 3* (LAKE et al., 2004). GML is designed and widely used for data exchange in spatial data infrastructures.

CityGML defines specific levels of detail (LoD), which differ with regard to semantic and geometric resolution. The most detailed LoD4 enables the modeling of the outer hull and of interior structures of buildings. The corresponding UML diagram is given in Fig. 2. A building is defined geometrically by its outer hull (*Solid*), external constructions like balconies and dormers (*BuildingInstallations*), and *Rooms*, which may contain immovable installations (*BuildingInstallations*) like stairs or pillars, or mobile objects (*BuildingFurnitures*) like desks or chairs. Rooms are represented geometrically by solids, and are bounded by thematic surfaces (*InteriorWallSurface*, *CeilingSurface*, *FloorSurface*), which may contain *Openings*. i.e. *Doors* and *Windows*.

Hence, CityGML meets the basic semantic requirements of indoor modeling for route planning. For special cases, e.g. planning of escape routes, more detailed semantic specifications of features are needed, e. g. the representation of fire protection classifications of doors, or of the side to which a door may be opened. These requirements can be included by the object-oriented concept of *Application Domain Extensions*, which is provided by CityGML to extend the model in a seamless and consistent way.

In the UML diagram, the spatial representation of features is given by geometric objects viz. *Solids, MultiSurfaces*, or *Geometries.* Additionally, these objects may be defined by corresponding topological primitives provided by GML - nodes, edges, faces, and topological solids. These data types cover the ones of the model described in the last section. Hence, CityGML is suitable for representing the geometric-topological model described in section 3 and will be used to represent and provide the building model derived by the grammar.

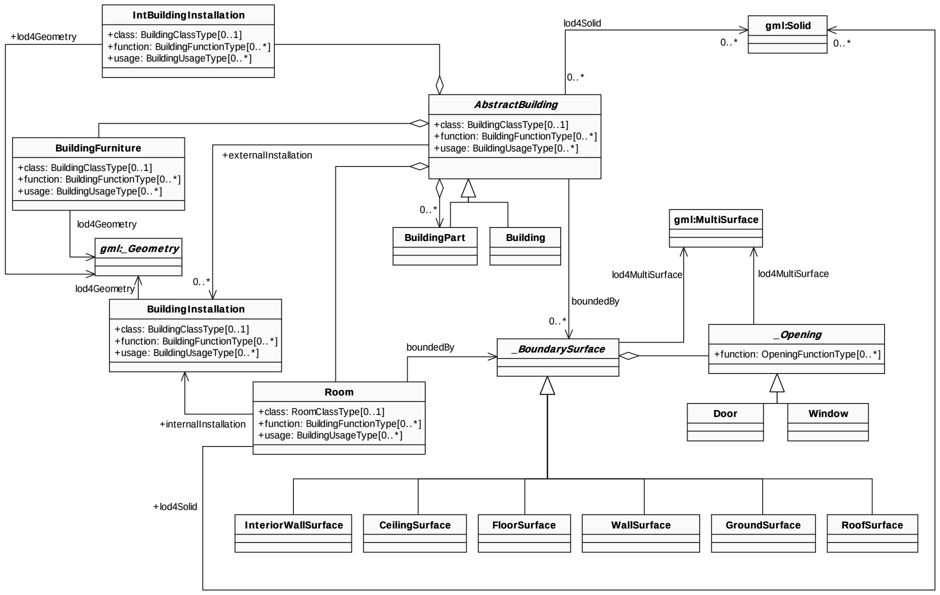


Fig. 2: Modeling of buildings and its internal structures in *CityGML* (GRÖGER et al., 2008). The UML diagram depicts the most detailed Level-of-Detail 4.

To summarize the related approaches presented so far, a model representing geometry and topology in a consistent way is available, and semantics is provided by CityGML. What is missing is a method to derive consistent, semantic building models, which are a suitable base to derive route graphs for indoor route planning, and a method to automatically derive such graphs from indor models. Approaches to cope with both tasks will be provided in the following sections.

5 A grammar for the generation of interior building models

In this section, an attribute grammar for the construction of building models with interior structures is presented. This grammar generates models which firstly are geometrical-topologically consistent (c.f. section 3), which secondly include the semantics which are mandatory to compute routes (c.f. section 4) and which thirdly require only a few observations to be generated, starting with the exterior hull of a building. In contrast to the grammar presented by MÜLLER ET AL. (2006), our approach guarantees consistency.

When grammar rules are applied in order to construct buildings by splitting objects, as in, e.g. MÜLLER ET AL. (2006), the context of rule applications and the ’knowledge’ is local, restricted to the object which is split. In contrast, concepts of consistency and reachability are more global, and therefore require information about objects which are close spatially, but ‘remote’ from a grammar derivation point on view. When a box *B* is split, for example, the inserted separating face is prohibited to interact with a door which has been inserted in a wall of a box adjacent to *B* by a previous rule application. This interaction would yield an error with regard to reachability and consistency. We solve this problem of combining local rules and global concepts of topology and reachability by introducing a global *constraint store*, which contains constraints providing the knowledge to prevent such error cases where a door and a wall interact in an inconsistent way. These constraints are generated by rule applications and explicitly formalize the concepts of adjacency, reachability and semantics.

5.1 *Representation of Geometry and Topology by Constraints*

The restriction of the geometry to boxes enables a simple representation of the model presented in section 3. In section 5.5, we will discuss how irregularly shaped buildings and rooms can be generated by aggregating boxes. We assume that the sides of the box are parallel to the axes of the coordinate systems; an affine transformation can be applied afterwards to transform the model to its location in geographic or geodetic space. Such a box can geometrically be represented uniquely by two corner points $P_1 \in \mathbb{R}^3$ and $P_2 \in \mathbb{R}^3$, such that the absolute values of the three components of the difference $P_1 - P_2$ are equal to the three side lengths the box.

Topology is represented explicitly by *constraints* stating relationships between faces. We consider the following three types of constraints:

- *Equality constraints*
  When two boxes share a common face, or when a wall surface shares a face with a box, the two corresponding faces F_1 and F_2 are identical, i.e. they denote the same face. This is stated by an equality constraint

  equals(F_1, F_2).
- *Aggregation constraints*
  When a box is split, the bounding faces are split accordingly. The relation of the original face (say, F) to the two resulting disaggregated faces (namely, F_1 and F_2) which have been split is stated explicitly by a ternary aggregation constraint

  aggr(F, F_1, F_2).

  The meaning of that constraint is that face F is the aggregation of faces F_1 and F_2, or vice versa, that the interiors of F_1 and F_2 are disjoint and that the union of F_1 and F_2 yields F.
- *Containment constraints*
  When an opening (face F_1) is inserted into a wall (face F_2), this relation is explicitly represented by a containment constraint

  inside(F_1, F_2).

  The face F_1 is contained in F_2, i.e. the point set representing F_1 is a subset of the point set representing F_2. Note that the boundaries of F_1 and F_2 do not have to be disjoint.

All constraints are generated by the applications of grammar rules and are collected in the constraint store. By applying geometrical constraint solving and reasoning methods to this set of constraints and the box geometry, topology can be derived. This kind of topological reasoning is similar to the approach presented in EGENHOFER (1991).

The constraints considered in this paper differ from the binary relations of the 4-intersection model (EGENHOFER & FRANZOSA, 1991) and its 3D extensions (ZLATANOVA, 2000): whereas the *equals* relation is the same, our *inside* denotes Egenhofer's relation *inside* as well as *coveredBy*. The aggregation constraint goes beyond the relations of the 4-intersection model, since it states that one face is equal to the disjoint union of two other faces.

5.2 *Symbols and Attributes*

The components of the grammar $G = (N, T, S, P)$ for the construction of building models are defined as follows: The set N of non-terminal symbols[1] is *{S, Box, Rectangle}*, where S is the start symbol, and the set T of terminal symbols is *{box, rectangle, internalDoor, externalDoor, window}*. The non-terminal symbol *Box*, which represents either a building, a storey, a part of a storey or a room, has the following attributes:

- The geometry of a *Box* is given by two corner points $P_1, P_2 \in \mathbb{R}^3$, as introduced above.
- The six rectangular faces bounding the *Box* are specified implicitly by six attributes $F_1, \ldots, F_6$ of type *rectangle*. Each F_i is characterized by two Boolean values *OuterFace* resp. *VerticalSepa-*

---

[1] As usual, non-terminal symbols start with an upper-case letter and terminals with a lower-case letter.

ratingFace, which are *true* if and only if the rectangle is in the outer hull of the buildings resp. is a vertical rectangle separating rooms. Both values are used to denote potential positions for windows and (internal and external) doors. By a naming schema (c.f. the presentation of rule R1), the six faces $F_1,..., F_6$ are uniquely assigned to the six bounding faces of a box. Hence, the geometry of each of the F_i can be derived anytime from the corresponding box geometry.

- The thematic classification of a box is denoted by a *Type* attribute. Possible values are {*Building, BuildingPart, Storey, StoreyPart, Room, Hall, Staircase*}.

The attributes of a terminal *box* symbol are identical to the attributes of non terminal *Box*. A non-terminal *Rectangle* represents a thematic surface, e.g. a wall or a roof surface. Its attributes are a *Type* with potential values {*Wall*, *Ceiling*, *Floor*} and a geometry *Geom* which represents the extent of the surface by two points $P_1, P_2 \in \mathbb{R}^3$. Again, the attributes of the terminal *rectangle* symbol are the same. The terminal symbols *internalDoor, externalDoor* and *window* have only a *Geom* attribute which is defined accordingly.

5.3 *Rules*

The grammar rules are enhanced by parameters, whose values are instantiated when the rule is applied. These values typically emerge from observations and are denoted in brackets, following the left hand non-terminal symbol. According to the concept of *guarded horn clauses* (UEDA, 1985), a rule may have a precondition or *guard*, which prevents rule applications yielding an inconsistent state. In general, a rule can only be applied if the guard, which is a Boolean expression, yields true. This concept is also used in the *CGA shape grammar* of MÜLLER ET AL. (2006).

Now the set *P* of grammar rules which consists of the rules R1 to R8 is introduced:

R1: $Box\ (Direction, Ratio, Type_1, Type_2): Pre \rightarrow Box^1\ \ Box^2$

This rule splits a box, denoted by the non-terminal *Box*, in two boxes, namely Box^1 and Box^2, if the guard *Pre* holds true; this will be explained later. The three non-terminals *Box* denote the same symbol but differ with regard to the attribute values; hence they are differentiated by superscript indices. *Direction*, *Ratio*, $Type_1$ and $Type_2$ are parameters. The direction of the split, either along the x-, the y- or the z-axis, is denoted by a parameter, as well as the split *Ratio* in the range $0 < Ratio < 1$. The thematic classifications of the resulting boxes are given by two parameters $Type_1$ und $Type_2$. In the following, the case *Direction* = ‘x’ is considered only; the other cases are defined accordingly. The geometrical aspect of rule R1 is implemented by a semantic rule S_1^1, which uses a procedure *split* to compute the corner points of both new boxes Box^1 and Box^2:

$\mathbf{S_1^1}$: $(Box^1, Box^2) = split(Box, Direction, Ratio)$

The result of this rule is the computation of the corner points of the new boxes. Hence, P_1 and P_2 are inherited attributes. The calculations are carried out by the following constraints, which are added to the constraint store (again, the case *Direction = ’x’* is considered):

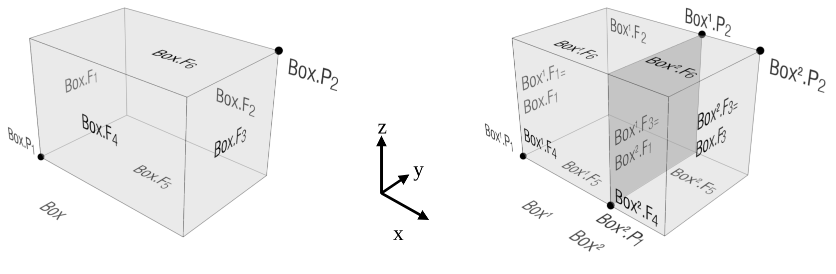
- $Box^1.P_1 = Box.P_1$
- $Box^2.P_2 = Box.P_2$
- $Box^1.P_2 = (Box.P_1.x + |Box.P_2.x - Box.P_1.x| * Ratio, Box.P_2.y, Box.P_2.z)$
- $Box^2.P_1 = (Box.P_1.x + |Box.P_2.x - Box.P_1.x| * Ratio, Box.P_1.y, Box.P_1.z)$

A second semantic rule accompanying the grammar rule R1 generates the following seven constraints and adds them to the constraint store:

$\mathbf{C_1^1}$: $equals(Box^1.F_1, Box.F_1)$
$\mathbf{C_1^2}$: $equals(Box^2.F_3, Box.F_3)$
$\mathbf{C_1^3}$: $equals(Box^1.F_3 , Box^2.F_1)$
$\mathbf{C_1^4}$: $aggr(Box.F_{2,} Box^1.F_2, Box^2.F_2)$
$\mathbf{C_1^5}$: $aggr(Box.F_5, Box^1.F_5, Box^2.F_5)$
$\mathbf{C_1^6}$: $aggr(Box.F_6, Box^1.F_6, Box^2.F_6)$

C_1^7: $aggr(Box.F_4, Box^1.F_4, Box^2.F_4)$

These equations completely specify the topological relations between the faces bounding the origin *Box* and the faces bounding the boxes Box^1 and Box^2. Fig. 3 illustrates these relations between faces by giving an example of the application of rule R1. If the common face $Box^1.F_3$ is vertical, the attribute *VerticalSeparatingFace* is set to *true.*



a) b)

Fig. 3: Example for the application of rule R1. a) box before and b) after rule application with values *Direction* = *‚x'* and *Ratio = 0,7*. The rectangle shared by Box^1 and Box^2 is depicted in grey color.

The precondition for splitting a box by rule R1 is that the inserted face which separates Box^1 from Box^2 does not split or affect an existing opening, viz. an internal or external door, or a window. This is prevented by the guard *Pre* of R1, which operates as follows: First, all openings in face $Box.F_4$ or in face $Box.F_2$ – the faces affected by the face to be inserted – are determined. Note that an opening is related to a wall by the relation *inside* (c.f. the discussion of rule R4 below), and that a wall is related to the corresponding face of a box by an *equals*-constraint (c.f. rule R3 below). The procedure *collectOpenings,* the pseudo code of which is given now, returns a list of all openings in a face *F*:

```
Procedure List(Openings)  collectOpenings(rectangle F)
    //First case: F has not been split ('*' denotes an arbitrary value)
    if there is no constraint aggr(F,*,*)
            then return all openings o with equals(F, W) and inside(o, W)
            else //second case: F has been split: recursive calls for both faces
                select F', F'' with aggr(F,F',F'')
                return collectOpenings(F') ∪ collectOpenings(F'')
```

For the set of openings collected by the procedure, it has to be assured that all are disjoint to the face $Box^1.F_3$ resp. $Box^2.F_1$ inserted by rule R1. Equivalently, all openings have to have the relation *inside* to either $Box^1.F_4$, $Box^1.F_2$, $Box^2.F_4$ or $Box^2.F_2$ (c.f. Fig. 3). This topological relation can be checked by implementing a corresponding predicate *inside* and then by analyzing the constraint store.

Now all prerequisites for the implementation of the precondition *Pre* of rule R1 have been introduced. It is as follows (again in pseudo code):

```
Pre:  List(Openings) LO = collectOpenings($Box.F_2$) ∪ collectOpenings($Box.F_4$);
      if for all openings o in LO either inside(o, $Box^1.F_4$) or inside(o, $Box^1.F_2$)
              or inside(o, $Box^2.F_4$) or inside(o, $Box^2.F_2$) holds
                  then return true;
                  else return false;
```

Note that a non-terminal symbol that has been replaced by a rule application and does not occur on the right-hand side of the rule is not deleted as in standard grammars, but is labeled as nonreplaceable. Particularly, all attributes are still available, since they are required to reconstruct topology and geometry.

The next rule is applied to a non-terminal *Box* symbol if it is not intended to split it any further or to derive any wall symbol from its boundary; it becomes a terminal *box* symbol. The rule is applicable only if the type of the Box is allowed in terminal boxes, i.e. if the type is *Room, Hall* or *Staircase*. This is assured by a precondition:

R2: $Box: Box.Type \in \{Room, Hall, Staircase\} \rightarrow box$

The attribute values of *box* are obtained by copying the values of *Box.*

The third rule takes one of the six faces bounding a box and designates it explicitly as semantic surface object, viz. a wall, a ceiling or a ground surface, according to CityGML. One purpose of the rule is to construct a rectangle in which an opening, i.e. a door or a window, can be inserted afterwards. The geometry of the rectangle is given by a variable *Index* ($1 \le Index \le 6$), which refers to the numbering of boxes' faces (see Fig. 3).

R3: $Box\ (Index, Type): Type \in \{Wall, Ceiling, Floor\} \rightarrow Box\ \ Rectangle$

The following constraints are added to the constraint store by rule R3:

C_3^1: $equals(Rectangle.Geom, Box.F_{\mathrm{Index}})$

and the attribute *Rectangle.Type* is set to the value of *Type.*

The next rule inserts a rectangular opening into a rectangle derived by rule R3. The geometry of an opening is specified by two points $P_1, P_2 \in \mathbb{R}^3$:

R4: $Rectangle\ (P_1, P_2): Pre \rightarrow Rectangle\ \ interiorDoor$

The geometry of the terminal *interiorDoor* is derived from the opposite corner points P_1 and P_2. Four preconditions, the conjunction of which is denoted by *Pre,* have to be fulfilled before R4 can be applied: 1) The geometry of the *interiorDoor* has to be disjoint to other openings, 2) the geometry of the *interiorDoor* has to be disjoint to the boundary of other faces, 3) the *Rectangle* has to be vertical (*Rectangle.VerticalSeparatingFace = true)*, and 4) both points P_1 and P_2 have to be inside the *Rectangle*. The first precondition is similar to the one introduced in R1 and can be implemented accordingly by using the procedure *collectOpenings*. The second one again is similar to the one of R1 which avoids conflicts between walls and openings.

The rule R4 adds a single constraint to the store, which states that the opening is inside the wall face:

C_4^1: $inside(interiorDoor.Geom, Rectangle.Geom)$

In a similar fashion, rules for inserting exterior doors and windows are provided, which differ only with regard to the third precondition: it is replaced by *Rectangle.OuterFace = true*. Exterior doors are inserted by the rule

R5: $Rectangle\ (P_1, P_2): Pre \rightarrow Rectangle\ \ exteriorDoor$

whereas the windows are covered by rule R6, where again the third precondition is *Rectangle.OuterFace = true*:

R6: $Rectangle\ (P_1, P_2): Pre \rightarrow Rectangle\ \ window$

The following rule replaces a non terminal rectangle by a terminal one, copying all attributes:

R7: $Rectangle \rightarrow rectangle$

The following constraint is added to the store:

C_7^1: $equals(Rectangle.Geom, rectangle.Geom)$

Finally, a rule generates an initial *Box* from the start symbol *S*, by providing a 3D point *P*. The box is defined by two opposite points, the origin $(0, 0, 0)^t$ and *P*, as described above:

R8: $S\ (P) \rightarrow Box$

The attributes of *Box* are initialized accordingly, and the values $Box.F_i.OuterFace = true$ and $Box.F_i.VerticalSeparatingFace = false$ are set for all faces F_i, $1 \le i \le 6$. The attribute *Box.Type* is set to the value '*Building*'.

To generate an indoor model by applying the rules R1 to R8, only a small set of observations is required. The outer hull of a building is represented by three values (length, width, height), and for each splitting step only the split direction and one value are necessary: the storey height or the room length or depth. For each door or window, only two points denoting its position relative to the wall are necessary. For the operational formalization and processing of the grammar rules, the Java-based *XGep* tool (SCHMITTWILKEN et al., 2009), which has been developed at the Institute for Geodesy and Geoinformation, is suitable. The sequence of grammar rule calls, including the required parameter values, can be represented in an XML structure which is immediately processed by XGep.

If an indoor model, e.g. a CAD (Computer Aided Design) model, is already available, a conversion tool can be developed that generates the XML structure, which is processed by the XGep tool and initiates the sequence of rule calls. If the structure of the indoor model corresponds to the model generated by the grammar (box-like structure), this conversion can be done automatically. Hence, a CAD model can be converted in a model the consistency of which is guaranteed. Likewise, 2D floor plans can be used to activate corresponding grammar rule calls for splitting storey boxes, yielding an indoor model for the corresponding storey.

5.4 *Examples*

The stepwise generation of a building with interior structures by applying rules R1 to R8 is exemplified in Fig. 4. From the start symbol *S* and parameter value *P* = (10, 14, 6), a *Box* with corresponding lengths of its sides and six faces is derived (a) by rule R8. Applying R1 (*Direction = ‚y’*, *Ratio = 0.15, Type*$_1$=*‘Staircase’*, *Type*$_2$=*‘BuildingPart’*) to *Box*, the structure in figure b) results, which consists of *Box*^{1} and *Box*^{2} with corresponding attribute values. A face (depicted in dark color) is shared by both boxes. Now *Box*^{2} is split, by application of rule R1 with *Direction* = ‘z’, *Ratio* = 0.5 and *Type*$_1$ = *Type*$_2$ = ‘*Storey*’ (Fig. c). Afterwards, the upper storey is split in x-axis direction (Fig. d), with *Type*$_1$=‘*Floor*’ and *Type*$_2$ = ‘*StoreyPart*’. The *Box* with Type ‘*StoreyPart’* again is split in *y*-direction by Rule R1 (*Type*$_1$ = ‘*Room’* and *Type*$_2$ = ‘*StoreyPart’*), yielding the structure in Fig. 4e).

Afterwards, the resulting ‘*StoreyPart*’-*Box* is split by R1 in two *Rooms* (f). Now, two non-terminal *Rectangles*, namely *Rectangle*^{1} and *Rectangle*^{2}, are generated by applying rule R3 to the *Box* with type *'Floor'*, first with index 4 and second with index 3, according to the naming schema (cf. Fig. 3). In both applications of R3, the *Type* parameter is set to ‘*Wall*’. By applying rule R4 once to *Rectangle*^{1} and three times to *Rectangle*^{2}, four internal doors are inserted (Fig. f). Note that topology, i.e. the adjacency between the *'floor' Box* and the *'Room' Box*, for example, is represented explicitly by constraints, particularly be equations. Hence, the reachability relation between rooms can easily be derived by considering these adjacencies and semantics, i.e. a *Door* in a *Rectangle*.

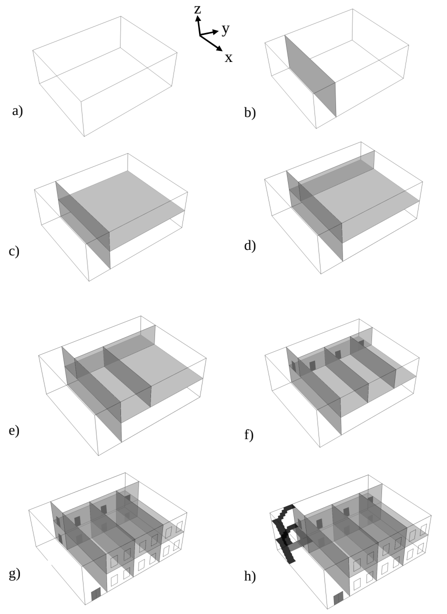


Fig. 4: Stepwise generation of a building with interior structures by applications of grammar rules. The stairs in h) can be obtained by additional grammar rules provided by SCHMITTWILKEN ET AL. (2007).

The lower storey is generated in a similar fashion, and windows are inserted by applications of rule R6, in the same way as doors have been inserted by rule R4. Finally, rule R5 inserts an exterior door (Fig. g). The insertion of stairs, as depicted in Fig. h), can be achieved by the grammar presented by SCHMITTWILKEN ET AL. (2007). Fig. 5 depicts the derivation tree of the rule applications, which corresponds to the generation of the building in the example.

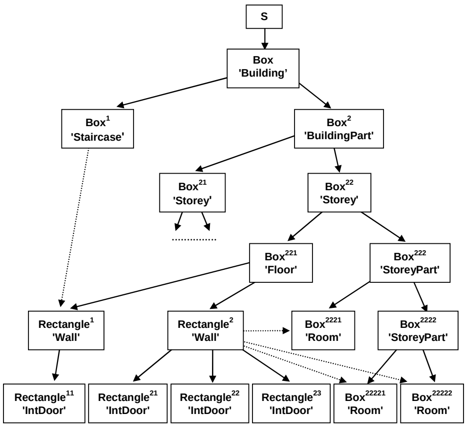


Fig. 5: Derivation tree for the generation of the building in Fig. 4. Bold arrows: rule applications; dotted arrows: relations derived from the constraints. The step of replacing a non terminal by a corresponding terminal is omitted.

A more detailed view on the constraints is given in Fig. 6, which depicts a slightly modified clipping of the previous example. A door *D* is to be inserted in a wall *W* by rule R6. The scene is represented by the constraints (only the relevant ones are given):

C^1: $equals(Box^1.F_3, Box^2.F_1)$

C^2: $aggr(Box^1.F_3, Box^{21}.F_1, Box^{22}.F_1)$

C^3: $equals(Box^{21}.F_1, W.Geom)$

From C^1, C^2 and C^3 it can be deduced that *W* is part of the boundary of Box^1 as well as of Box^{21} (C^3 only), hence the door *D* can be inserted, which connects both rooms represented by Box^1 and Box^{21}, and the constraint

C^4: $inside(D.Geom, W.Geom)$

is added to the store.

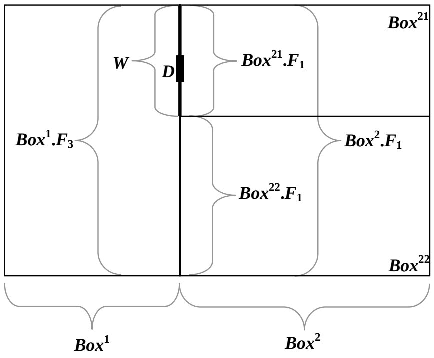


Fig. 6: Indoor scene where a door *D* is inserted in a wall *W* (view from above). From the constraints, it can be deduced that *W* and hence *D* are in the boundary of both Box^1 and Box^{21}.

5.5 *Extension to Extruded Simple Polygons*

The concepts introduced so far aim at generating box-shaped rooms and interior structures from box-shaped building hulls. However, in reality buildings and rooms often have a more irregular, typically L-, T- or X-shaped, structure. We now sketch how to generalize our data types as well as our rules to cope with such irregular shapes. Nearly all buildings and interior structures can be represented by polygonal footprints with right angles, where the floors are horizontal and the walls are vertical. Such structures are called *prisms* in our paper. Our prisms are special cases of general prisms, since all side surfaces are vertical and all angles are 90°.

For adapting our concept to prisms, only a few local changes of the rules are required. Since each prism can be generated by successively splitting boxes in both directions, we only need to extend our approach by an aggregation concept for boxes that defines which boxes belong to the same feature (building, storey, room, ...). This aggregation can be represented by additional constraints. Depending on those constraints it can be determined whether a side of a box is a wall or just a separation between boxes of the same object; in that case, it is labeled as 'invisible'. Fig. 7 gives an example: We start with three boxes forming a L-shaped building with a protrusion. By splitting boxes (application of a modified rule R1) a polygonal footprint is generated. $Room^1$, for example, is generated by splitting Box^1 two times in different directions. The dotted lines indicate that different boxes belong to the same feature (room). The detailed design of rules for generating complex prism structures will be the topic of a subsequent paper. A further extension of the approach to extruded polygons with arbitrary angles can be achieved by additional rules which modify walls locally by applying rotations; the (local) topology has to be updated consistently.

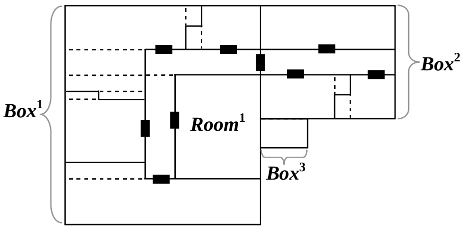


Fig. 7: Irregularly shaped footprint of a L-shaped building, which has been generated by splitting boxes and merging boxes belonging to the same room (separated by dotted lines).

5.6 *Derivation of Route Graphs*

Based on the building model and the constraints generated by the grammar, the method which derives a route graph *RG = (Nd, Ed)* automatically that represents the reachability inside a building is straightforward. In that graph, *Nd* is a set of nodes and *Ed* a set of undirected edges. In pseudo code notation, the method to derive the route graph is as follows:

```
Input: A derivation tree generated by the grammar
Output: A route graph $RG = (Nd, Ed)$ representing reachability
1. $Nd = \emptyset$; $Ed = \emptyset$:
2. For each terminal room, hall or staircase generated by the grammar, a node is added to $Nd$.
3. For each terminal internalDoor:
    a. get the related wall rectangle terminal $w$ (using constraint inside)
    b. get the two room/hall/staircase terminals $b_1$ and $b_2$ related to $w$, by analyzing the constraint store
    c. add an edge relating the nodes corresponding to $b_1$ and $b_2$ to the set $Ed$
4. Add an additional node representing the space outside the building to $Nd$, at least one for each externalDoor, and add corresponding edges to $Ed$. The nodes may be connected to the outdoor network.
```

To extend the method to planning of escape routes, additional edges and nodes have to be considered. Escape routes using windows near ground can be incorporated, since those windows can be detected by using the three-dimensional geometry provided by the model. Those cases can be treated similar to exterior doors, by providing an additional node outside the building which is connected with the corresponding node representing the room. This will be the topic of a subsequent paper.

6 Conclusions

In this paper, we have introduced a grammar which generates buildings and their interior structures. It requires only a small set of observations. In contrast to earlier grammar approaches, e.g. MÜLLER ET AL. (2006), our grammar generates buildings which are geometric-topologically consistent. This is due to the explicit representation of topology: the adjacency between solids, viz. rooms, floors of staircases, is represented explicitly by constraints stating the identity of shared faces, and that adjacency representation is maintained when solids or faces are split. Hence, the topology can be generated on demand. Consistency between geometry and topology is assured by using the model introduced by GRÖGER & PLÜMER (2010) where consistency is guaranteed by formal proofs. Once the correct topology is given, the derivation of a route graph which represents the reachability relation inside and outside the building and is suitable for indoor navigation is straightforward. This is due to the correct topology and the representation of semantics as provided by *CityGML*. The semantic information is generated by the grammar, simultaneously to the derivation of geometry and topology. An concept to deal with more complex building hulls and interior structures has been presented; its implementation requires only local modifications and extensions of the grammar rules. This will be elaborated in detail as the next step.

A further step will be the addition of rules which allow for the generation of more features inside and outside buildings, like balconies, pillars and furniture. Stairs can be incorporated by importing the grammar of SCHMITTWILKEN ET AL. (2007), and a more precise geometrical description of edges by using skeleton algorithms, as proposed by LEE & ZLATANOVA (2009) or MEIJERS ET AL. (2005).

An interesting and relevant extension of the concept will be the incorporation of *escape routes* as a special case of indoor route planning. The problem of automatically deriving route graphs suitable for finding optimal escape routes in the case of an emergency is much more difficult than ordinary indoor route planning. In an emergency case, the options to leave a building are much more multifaceted: Emergency exits or fire escapes which are usually closed may be used, or balconies or windows at or near ground level. Even balconies or windows far beyond ground level are suitable, when fire ladders or other rescue facilities, e.g. extension masts or rescue nets, are provided. In addition, persons may be evacuated from roofs by helicopters. Hence, the interface between indoor and outdoor is crucial for the derivation of escape routes. A more sophisticated notion of neighborhood between reachable spaces is required to cope with such cases. This extension requires in addition the enrichment of the generated model, since the roof type, for example, must be considered as well as the terrain surrounding a building.

Acknowledgements

The authors appreciate the helpful and detailed comments given by the anonymous reviewers. We thank Jan Prinz for proof-reading and improving the English language and style of this text and Gerrit Kowatsch for assistance in preparing the illiustrations.

References

ALBLAS, H. & MELICHAR, B. (eds.), 1991: Attribute Grammars, Applications and Systems. – In: Proc. Int. Summer School SAGA, Prague, Czechoslovakia, June 4 - 13, Lecture notes in computer science 545, Springer, Berlin, Heidelberg.

ANAGNOSTOPOULOS, C., TSETSOS, V., KIKIRAS, P. & HADJIEFTHYMIADES, S., 2005: OntoNav: A semantic indoor navigation system. – In: Proc. of the 1st Workshop on Semantics in Mobile Environments (SME'05) in conjunction with the 6th Int. Conf. on Mobile Data Management (MDM'05), May 9, Ayia Napa, Cyprus.

BECKER, T., NAGEL, C. & KOLBE, T. H., 2009: A Multilayered Space-Event Model for Navigation in Indoor Spaces. – In: LEE, J. & ZLATANOVA, S. (eds.), 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography, Springer, Berlin, Heidelberg, 61–78.

BOOCH, G., RUMBAUGH, J. & JACOBSON, I., 2005: Unified Modeling Language User Guide. – Addison-Wesley Longman.

BRENNER, C. & RIPPERDA, N., 2006: Extraction of Facades Using RJMCMC and Constraint Equations. – In: FÖRSTNER, W. & STEFFEN, R. (eds.), Proc. of Symp. of ISPRS Commission III Photogrammetric Computer Vision. PCV’06, September 20-22, Bonn, Germany, 165–170.

CHALMET, L. G., FRANCIS, R. L. & SAUNDERS, P. B., 1982: Network Models for Building Evacuation. – Management Science **28**(1):86–105.

CHOMSKY, N., 1959: On Certain Formal Properties of Grammars. – Information and Control **2:**137–167.

DIJKSTRA, E. W., 1959: A note on two problems in connexion with graphs. – Numerische Mathematik **1:**269–271.

DÖRSCHLAG, D., GRÖGER, G. & PLÜMER, L., 2008: Über die schrittweise Erstellung und Verfeinerung von Modellhypothesen für Gebäude (in German). – Photogrammetrie, Fernerkundung, Geoinformation PFG(3):157–164.

EGENHOFER, M. J., 1991: Reasoning about binary topological relations. – In: GÜNTHER, O. & SCHEK, H. J. (eds.), Advances in Spatial Databases. 2nd Symp, SSD '91 Zurich, Switzerland, August 28-30, 1991 Proceedings, Lecture notes in computer science, vol. 525, Springer, Berlin, Heidelberg, 143-160.

EGENHOFER, M. J. & FRANZOSA, R. D., 1991: Point-Set Topological Spatial Relations. – International Journal of Geographical Information Science **5**(2):161–174.

EGENHOFER, M. J. & HERRING, J. R., 1991: Categorizing binary topological relationships between regions, lines and points in geographic databases. – Technical Report, Department of Surveying Engineering, University of Maine, Orono, ME.

GRÖGER, G., 2006: Konsistente Modellierung virtueller Städte und Regionen (in German). – Habilitation Thesis, University of Bonn.

GRÖGER, G., KOLBE, T. H., CZERWINSKI, A. & NAGEL, C., 2008: OpenGIS City Geography Markup Language (CityGML) Encoding Standard. – Version 1.0.0. OGC Doc. No. 08-007r1.

GRÖGER, G. & PLÜMER, L., 2009: Updating 3D City Models – how to preserve geometric-topological consistency. – In: AGRAWAL, D. et al. (eds.), Proc. of the 17th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2009), Seattle, Washington, November 4.-6., 2009, ACM Press, New York, 536–539.

GRÖGER, G. & PLÜMER, L., 2010: How to Achieve Consistency for 3D City Models. – Geoinformatica (Online First) DOI: 10.1007/s10707-009-0091-6.

GUPTILL, S. C. & MORRISON, J. L., 1995: Elements of Spatial Data Quality. – Elsevier, Oxford.

HATCHER, A., 2001: Algebraic topology. – Cambridge University Press, Cambridge.

HOPPE, B. & TARDOS, E., 1995: The Quickest Transshipment Problem. – In: Proc. 6. ACM-SIAM Symp. on Discrete Algorithms (SODA), January 22-24, San Francisco, California, 422–441.

ISO TC 211 , 2002: ISO 10113:2002: Geographic information - Quality principles. – International Organization for Standardization.

KOLBE, T. H., GRÖGER, G. & PLÜMER, L., 2005: CityGML - Interoperable Access to 3D City Models. – In: VAN OOSTEROM, P. et al. (eds.), Geo-information for Disaster Management. Proc. of the 1st International Symposium on Geo-information for Disaster Management', Delft, The Netherlands, March 21-23, Springer, Berlin, Heidelberg, 883–899.

KOLBE, T. H., GRÖGER, G. & PLÜMER, L., 2008: CityGML - 3D City Models for Emergency Response. – In: ZLATANOVA, S. & LI, J. (eds.), Geospatial information technology for emergency response, International Society for Photogrammetry and Remote Sensing book series, vol. 6, Taylor & Francis, London, 257–274.

LAKE, R., BURGGRAF, D., TRNINIC, M. & RAE, L., 2004: Geography Mark-Up Language: Foundation for the Geo Web. – John Wiley & Sons, Chichester.

LEE, 2001: A 3D Data Model for Representing Topological Relationships between Entities in Build Environments. – Ph.D. Dissertation, Department of Geography, Ohio State University.

LEE, J. & KWAN, M. P., 2005: A combinatorial data model for representing topological relations among 3D geographical features in micro-spatial environments. – International Journal of Geographical Information Science **19**(10):1039–1056.

LEE, J. & ZLATANOVA, S., 2009: A 3D data model and topological analyses for emergency response in urban areas. – In: ZLATANOVA, S. et al. (eds.), Geospatial information technology for emergency response, ISPRS book series, Taylor & Francis, London, 143–168.

LORENZ, B., OHLBACH, H. J. & STOFFEL, E. P., 2006: A Hybrid Spatial Model for Representing Indoor Environments. – In: CARSWELL, J. D. & TEZUKA, T. (eds.), Web and Wireless Geographical Information Systems. Proc. of the 6th Int. Symp. W2GIS 2006, Hong Kong, China, December 4 - 5, 2006, Lecture notes in computer science, vol. 4295, Springer, Berlin, Heidelberg, 102–112.

MEIJERS, M., ZLATANOVA, S. & PFEIFER, N., 2005: 3D geoinformation indoors: structuring for evacuation. – In: GRÖGER, G. & KOLBE, T. H. (eds.), Proc. of the 1st Int. ISPRS/EuroSDR/DGPF-Workshop on Next Generation 3D City Models, Bonn, June 21-22, EuroSDR, 11–16.

MÜLLER, P., WONKA, P., HAEGLER, S., ULMER, A. & VAN GOOL, L., 2006: Procedural Modeling of Buildings. – In: Proceedings of ACM SIGGRAPH 2006 / ACM Transactions on Graphics (TOG) ACM Press, 614–623.

MÜLLER, P., ZENG, G., WONKA, P. & VAN GOOL, L., 2007: Image-based Procedural Modeling of Facades. – In: Proceedings of ACM SIGGRAPH 2007 / ACM Transactions on Graphics, ACM Press, New York, NY, 85–93.

NORVIG, P. & RUSSELL, S., 2003: Artificial Intelligence - A Modern Approach. – Pearson Education, Inc., New Jersey.

PU, S. & ZLATANOVA, S., 2005: Evacuation route calculation of inner buildings. – In: VAN OOSTEROM, P. et al. (eds.), Geo-information for Disaster Management. Proc. of the 1st International Symposium on Geoinformation for Disaster Management, Delft, The Netherlands, March 21-23, Springer, Berlin, Heidelberg, 1143–1161.

SCHMITTWILKEN, J., DÖRSCHLAG, D. & PLÜMER, L., 2009: Attribute Grammar for 3D City Models. – In: KREK, A. et al. (eds.), Urban and Regional Data Management. UDMS Annual 2009, CRC Press, Taylor & Francis Group, 49–58.

SCHMITTWILKEN, J., SAATKAMP, J., FÖRSTNER, W., KOLBE, T. H. & PLÜMER, L., 2007: A Semantic Model of Stairs in Building Collars. – Photogrammetrie, Fernerkundung, Geoinformation PFG(6):415–428.

STINY, G. & GIPS, J., 1972: Shape Grammars and the Generative Specification of Painting and Sculpture. – In: FREIMAN, C. V. & GRIFFITH, J. E. (eds.), Proc. of IFIP Congress 71, Vol 2 - Applications, Ljubljana Yugoslavia, August 23-28, 1971, 1460–1465.

UEDA, K., 1985: Guarded Horn Clauses. – In: Logic Programming '85, Lecture notes in computer science, vol. 221, Springer, Berlin, Heidelberg, 168–179.

VAN TREECK, C. & RANK, E., 2004: Analysis of building structure and topology based on Graph Theory. – In: Proc. 10th Int. Conf. on Computing in Civil and Building Engineering, Weimar, Germany.

WONKA, P., WIMMER, M. & WILLIAM, F. S., 2003: Instant Architecture. – ACM Transactions on Graphics **22**(4):669–677.

ZLATANOVA, S., 2000: 3D GIS for urban development. – ITC dissertation no. 69, International Institute for Aerospace Survey and Earth Sciences, Enschede, Netherlands.

Address of the Authors:

Priv.-Doz. Dr. rer. nat. GERHARD GRÖGER and Prof. Dr. rer. nat. LUTZ PLÜMER, University of Bonn, Institute for Geodesy and Geoinformation, Meckenheimer Allee 172, D-53115 Bonn, Tel.: +49-228-731764, +49-228-731750, Fax: +49-228-731753, e-mail: groeger@igg.uni-bonn.de, pluemer@igg.uni-bonn.de