9th 3DGeoInfo Conference 2014

Proceedings

Editors: Martin Breunig, Mulhim Al-Doori, Edgar Butwilowski,
Paul Vincent Kuper, Joachim Benner, Karl-Heinz Haefele

Published by the Conference Chairs of 3DGeoInfo 2014 in Karlsruhe, Germany.
Part of 3DGeoInfo conference proceedings series.

A multi-step transformation process for automatically generating indoor routing graphs from existing semantic 3D building models

Aftab Ahmed Khan [*], Andreas Donaubauer, Thomas H. Kolbe

Chair of Geoinformatics, Technische Universität München, Germany

`(aftab.khan,andreas.donaubauer,thomas.kolbe)@tum.de`

Abstract. IndoorGML, a draft standard of the Open Geospatial Consortium (OGC), defines an information model for indoor space based on the requirements of indoor navigation. IndoorGML allows to represent, manage, and store different infrastructures of the indoor environment in primal (volumetric and boundary geometries) and dual (graph model) spaces along with semantic information. Furthermore, it provides a sound mathematical framework to derive, use, and manage parallel and hierarchical graph structures (layers) based on the different contextual considerations for the purpose of indoor navigation and information services. IndoorGML is not tightly coupled with a specific type of semantic 3D building model. Instead, existing standards for semantic 3D building models from the Building Information Modeling (BIM) and Topography Information Modeling (TIM) domains, namely the Industry Foundation Classes (IFC) and the City Geography Markup Language (CityGML) can be used in combination with IndoorGML. IndoorGML provides a unique platform for existing 3D semantic building models to integrate, manage, and to extend their horizon of applications with the other indoor thematic context spaces, e.g., sensor space. Therefore, there is a need to investigate the potential of integrating these different semantic building models with the IndoorGML model. This investigation goes beyond the conversion from one schema to the other; it also includes the concept of automatically deriving correct navigation structures for the indoor navigation with different types of locomotion.

In this work, we describe a multi-step transformation process to automatically generate IndoorGML datasets from existing indoor building model data given in either IFC or CityGML LoD4. Moreover, we address semantic transformations, geometric transformations, topologic analyses, and spatial reasoning in order to derive navigation structures for the different types of locomotion. We tested our methods with a complex public building. In addition to the description of our conceptual work, this paper documents the lessons we learned from this test.

1 Introduction

CityGML and IFC are two well-known semantic models from the Topography Information Modelling (TIM) (GIS), and Building Information Modelling (BIM) domains.

Both semantic 3D building models represent and manage semantic, geometry, and topology information through different approaches, e.g., CityGML uses boundary representations to represent building geometry while IFC mainly uses volumetric and parametric approaches. In recent years, many researchers tried to integrate both models to take benefit from the respective other area of specialization. Most of these integrations or transformations aim at translating a dataset from one schema to the other (El-Mekawy 2012, Isikdag and Zlatanova 2009).

The new draft OGC indoor modelling standard, i.e., IndoorGML, facilitates the representation, storage, and management of primal (volumetric and boundary representation) and dual spaces (graph models) of different indoor thematic contextual spaces based on the requirements of indoor navigation (IndoorGML 2014a). In addition, for indoor navigation and information services, IndoorGML provides the opportunity to manage and integrate multiple as well as hierarchical graph models.

In order to use existing semantic 3D building models either modelled according to IFC or CityGML for the representation of topographic space in IndoorGML, the 3D building models need to be both abstracted to graph models and transformed into volumetric and boundary geometries including their semantic information. This transformation requires to take care of the correct topology apart from other transformations' requirements, such that the correct navigation structures can be derived. Therefore, unlike the traditional works to translate from one information model to the other, in our case, there is a need to investigate semantic transformations, geometric transformations, topology analysis and spatial reasoning with the objective to derive correct navigation structures for indoor navigation. As integral part of these transformations, there is a need to apply algorithms for creating subspaces of the topographic space, taking into account different locomotion types, namely walking, driving and flying.

In order to fulfill these requirements and in order to achieve a high level of automation in the transformation process, we designed a multi-step transformation process to automatically generate IndoorGML datasets from indoor building models either represented in IFC or CityGML LoD4. In the remainder of this paper, we describe this new multi-step approach. Rest of the paper is organized as follows; section 2 discusses related work, and section 3 presents the generic approach and transformation steps to achieve an IndoorGML model of the main building of Technische Universität München given in either IFC or CityGML. Furthermore, section 4 describes the requirements and results of computing subspaces for different locomotion types. In section 5 we draw conclusions regarding the transformation steps and deriving subspaces.

2 Related work

2.1 From IFC to CityGML LoD4

Many researchers address interoperability and interaction between IFC and CityGML models, which are two prominent semantic models in the thematic areas BIM and TIM (3D GIS) respectively. IFC is an international standard for AEC data exchange and representation. It is designed with the prime objective to represent building objects with geometrical and semantic information (BuildingSmart, 2014). On the other hand, CityGML is an OGC standard for the representation and exchange of 3D urban objects, including buildings (Kolbe, 2009). A number of publications and projects have focused on the integration of IFC and CityGML (Isikdag and Zlatanova 2009, De Laat and van Berlo 2011). Some researchers give attention to transformation of data from IFC to CityGML (De Laat and van Berlo 2011), whereas others focus on extending CityGML with regard to conceptual requirements for converting CityGML to IFC models (Nagel et al. 2009). There is also work on bidirectional transformation between CityGML and IFC using a unified building model (El-Mekawy et al. 2011). Most of the work on transformation of datasets from IFC to CityGML focuses on transformation of geometry and semantics from one representation to the other data model. However, in our case we are also interested in deriving detailed navigable graph structures according to the different locomotion types. Therefore, we focus on a detailed representation of a building model and use an elementary approach to convert 3D building models represented in IFC with semantic, topologic, and geometric information into CityGML and then to IndoorGML in order to achieve correct navigation structures (graphs).

2.2 From CityGML LoD4 or IFC to IndoorGML

CityGML is a well-known OGC standard to store, exchange, and represent urban objects. The main features of CityGML include multi-scale modeling, i.e., five Levels of Detail (LoDs) to represent a city from regional down to interior building level, modules that contain semantic modelling for different thematic areas, definition of classes and relations for the relevant topographic objects in cities. CityGML models objects with respect to their geometrical, topological, semantic, and appearance properties. Especially interesting for indoor navigation are CityGML LoD4 models since they represent interior structures of building, e.g., room, lamps, table, pillars, stairs, etc. with Opening, Room, Building Furniture, and Building Installation classes.

While CityGML defines a detailed representation of the semantic, geometric, and topology information of indoor 3D building at LoD4, (Becker et al. 2009a, Becker et al. 2009b) address the requirements and key concepts related to indoor navigation in indoor space. A proposal was forwarded by Nagel et al. (2010) to have a new standard, i.e., IndoorGML, for indoor space representation based on these requirements and concepts. IndoorGML allows to represent and exchange indoor space information that is essential to develop and implement indoor navigation systems. IndoorGML represents geometric and semantic properties of indoor space but they differ in the space representation from CityGML and IFC. Normally, it is recommended to use IndoorGML in combination with other standards particularly for the representation of indoor subdivisions, where a subspace represented in a subgraph externally references a common indoor building model represented in any other standard, e.g., CityGML (IndoorGML 2014a, IndoorGML 2014b). Therefore, it is considered as a complementary standard to CityGML or IFC to support indoor navigation services.

In our case, we intend to subdivide the indoor space according to different locomotion types. Based on physical constraints of the different locomotion types the navigable spaces can differ. These different geometric navigable models representing navigable spaces for different locomotion types cannot be represented in a common data model using external reference feature of IndoorGML. Thus, we have to create the indoor subspace models of buildings in IndoorGML. The subspace models in IndoorGML will be sublayers of the main topographic layer (representing the building model), furthermore, to make these subspaces coherent with the main topographic layer we consider it important to convert the building model represented in CityGML to IndoorGML. In the following, we present a detailed transformation of each feature type of a public building represented in CityGML LoD4 into IndoorGML for the purpose of computing subspaces.

2.3 Driving routing graphs according to different locomotion types

A great deal of research has been carried out on deriving navigation structures (abstracted graph models) for different locomotion types from 3D building models. In contrast to our work, most papers focus on a single type of locomotion only, e.g., walking or driving. The network graphs extracted from floor plans of the buildings make these graphs only navigable for the locomotion types which are dependent on floor surfaces of the building. However, the process of indoor route planning for different types of locomotion depends on a network graph that has to be extracted from the 3D building model. Hence, these methods (Sahlemariam et al. 2008, Tsetsos et al. 2005, Stoffel et al. 2007, Dudas et al. 2009, Goetz and Zipf 2011, Lertlakkhanaku and Soyoung 2009, Lin et al. 2013, Steuer 2013) do not take into consideration the free space in indoor environment. 3D free space has the same importance as floor surfaces representing the navigable space for specific locomotion types. For example, for a flying vehicle the floor surface can be non-navigable but the free space is important to be navigable for its navigation. It shows that a navigable floor surface of a room does not define the whole room to be navigable for flying objects like UAVs. Therefore, there is a need to represent and extract the network graph from the free space and other parts of interior environment separately to decide about their navigability. Some researchers (Goetz and Zipf 2011, Dudas et al. 2009) consider users or user groups for their indoor navigation and they define a profile for each user by defining his/her physical capabilities and preferences. Furthermore, the network model extracted from the main topographic model of the building is filtered (subgraphed) based on the user's profile. In contrast to our work, these approaches do not represent the actual geometric navigable space for the user because the subgraph representing navigable space for the user is computed from a supergraph. In our work, we are interested in the computation of the actual geometric navigable spaces for the different locomotion types considering their physical constraints. The actual geometrical navigable space is reflected by subgraphs of the main topographic model using IndoorGML. In this work, we are considering the conceptual constraint model presented in (Khan and Kolbe, 2012) for each locomotion type. Most of the previous research papers give the same preference to physical and temporal constraints of locomotion type but in our case we consider physical constraints to be the base, taking precedence on temporal requirements. So, we define the different subspaces based on physical constraints of the locomotion type and generate the different graphs from subspaces automatically.

3 Generating IndoorGML datasets from semantic 3D building models

The general concept of generating IndoorGML datasets from different semantic 3D building models either represented in IFC or CityGML LoD4 and determining navigation structures according to the different locomotion types based on their specific navigating constraints is illustrated in Fig. 1.

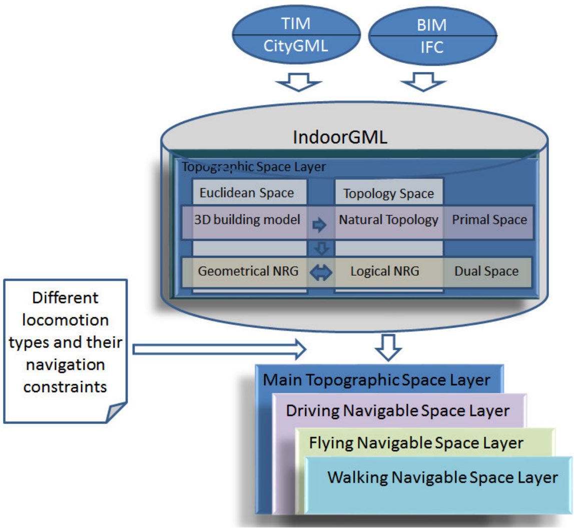

Fig. 1. Generating IndoorGML datasets from TIM and BIM sources and determining navigation structures according to the different locomotion types within IndoorGML.

The Multilayered Space-Event Model (MLSEM) is a framework defined by Becker et al. (2009a) which provides not only the method to abstract or to form graph geometries (Node Relation Graph (NRG) Lee 2004) from primal space (volumetric objects e.g. representing topographic space) but also defines a link between those graph models with other graph models representing different contextual thematic spaces of indoor environment for use in indoor applications, e.g., linking an indoor topographic layer with an another layer representing sensor covering area for route planning. IndoorGML, which is based on the MLSEM concepts, is not tightly coupled with a specific type of semantic 3D building model. Instead, existing standards for semantic 3D building models from the Building Information Modeling (BIM) and Topographic Information Modeling (TIM) domains, namely the Industry Foundation Classes (IFC) and the City Geography Markup Language (CityGML) can be used in combination with IndoorGML. In a simple case, transforming IFC or CityGML to IndoorGML just means to create references between nodes of the (manually created) Network Relation Graph (NRG) (Lee 2004) representing the topographic indoor space and the corresponding IFC *IFCSpace* or CityGML *Room* objects. As our intention is to automatically create subspaces of the indoor space described by the IFC or CityGML data and to automatically derive the NRG from these subspaces taking into account the constraints defined by different types of locomotion, the transformation process from IFC or CityGML to IndoorGML is a complex task.

In order to reduce complexity and to allow the existing semantic 3D building models to be represented both according to IFC and to CityGML, we divided this transformation task into multiple subtasks which are grouped into two main steps as shown in Fig. 2: in step 1, IFC data is semantically and geometrically transformed to CityGML LoD4 and the topology is analyzed; in step 2, CityGML LoD4 data is semantically, and geometrically transformed to IndoorGML. We investigated the transformation process from parametric representation to Boundary Representation (BRep) as required both by CityGML and IndoorGML. In the semantic transformation, we focused on transforming the maximum amount of the semantic information related with each indoor object following the schema rules of the IFC source and the CityGML target object. Whereas in topology analyses, we investigated the requirement to have correct topological relations of indoor building model's objects with their connected geometries, e.g., connected door and room geometries must correctly touch each other, there must be no overlap and they must determine boundary geometry. As IFC allows for a user to model a semantic 3D building in many different ways (Nagel et al. 2009), flexibility in the transformation to CityGML is required. We account for this requirement by using a standard spatial ETL tool, FME workbench in our case, for the implementation of step 1.

In the second step, the transformation from CityGML to IndoorGML has relatively fixed rules for the semantic and geometric transformation. Here, the focus of investigation was to transform boundary geometries from CityGML to volumetric space objects in IndoorGML including their semantic information, e.g., a multisurface room feature is translated into a room solid with its boundary geometries, i.e., interior wall surfaces, etc. Besides the transformation from CityGML to IndoorGML the third step of the overall transformation procedure deals with subspacing of topographic space and deriving the NRG for different locomotion types.

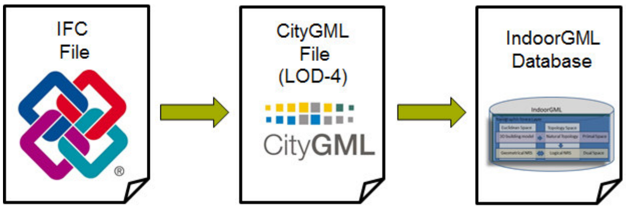

Fig. 2. 3D building model's transformation from IFC to CityGML LoD4 and then to IndoorGML.

3.1 Transformation from IFC to CityGML

The basic concepts and related work to transform from IFC to CityGML are discussed in detail in (Isikdag and Zlatanova 2009). In addition to these concepts, prior to the transformation step we need a detailed representation of the building model particularly for the identification of the navigable spaces, e.g., stairs, ramp, etc. and checking the topological relations between building elements so we can get navigation graph structures for different locomotion types.

The *IFCSpace* class defines all volumes and areas that are bounded by different building elements. For example, in Fig. 3 a room contains stairs. The whole space within that room including the stairs is represented as *IFCSpace*. As we need to compute the subspaces for different locomotion types and since for a specific type of locomotion the stairs are non-navigable (e.g., when using a wheelchair), whereas for another type of locomotion it is navigable (e.g., a walking person). Therefore, there is a need to represent the space above the stairs separately. Furthermore, all steps of the stairs may have different areas and properties. Therefore, each stair step has to be considered individually and, thus, the space above each stair step should have an individual representation (see Fig. 4). If a step is determined as non-navigable for some types of locomotion then the space above it will also be nonnavigable. The same approach is applied on each building element or area where its navigability is represented, e.g., free space above a ramp, free space within circular stairs, etc.

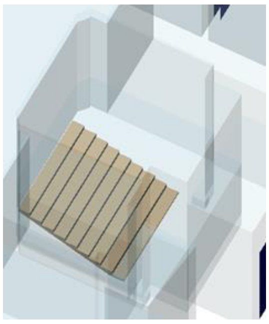

Fig. 3. *IFCSpace* representation of a room.

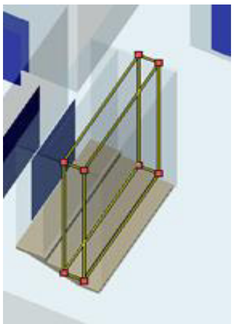

Fig. 4. Space representation above stairs

In the next step, all elements, e.g., *IFCStairs* and *IFCWall* objects, in the building model are checked whether they overlap the *IFCSpace*. If they overlap they are deduced from the *IFCSpace* to ensure that they have only a topological touch relation with the *IFCSpace*. *IFCOpeningElements*, which fill the void spaces in walls are checked for their topological relationships with the *IFCWallStandardCase* through the relation *IfcRelVoidsElements* and *IFCWallStandardCase* relation with the *IFCSpace* is checked through *IfcRelSpaceBoundary,* to be in touch relation and should not have an overlap or gap with the *IFCSpace*. Normally a door or window element fills an *IFCOpeningElement*, in this case we ignore the door or window geometry and consider the geometry of the *IFCOpeningElement* for the transformation because the former overlaps the later. Moreover, we provide simple conversion steps through which the transformation from IFC data into CityGML can be achieved. The conversion from IFC to CityGML is carried out in the following steps given in Table 1.

Table 1. Semantic mapping and transformation steps from IFC to CityGML dataset.

IFC Elements	Transformation details	CityGML Feature Types
IFCOpening-Element	Checking the relation *IFCRelFillsElement* of *IFCOpeningElement* with the *IFCDoor* or *IFCWindow* element; then the properties of *IFCDoor* or *IFCWindow* are attached to the respective *IFCOpeningElement. IFCOpeningElement* is converted into Door or Window *MultiSurface* geometries in CityGML as shown in Fig. 5.	Window MultiSurfaces/ Door MultiSurfaces
IFCSpace	*IFCSpace* geometry, which often is a parametric geometry in IFC is converted into boundary representation geometry and translated into a R*oom* feature (LoD4Solid) in CityGML as shown in Fig. 6.	Room
IFCSpace	*IFCSpace* is converted into *multiSurfaces*. Based on the height and relative altitude of *IFCSpace* the decision about each surface is taken, whether it is a *CeilingSurface* or a *FloorSurface*. If the height is between specific thresholds then it is tagged as *InteriorWallSurface*. Furthermore, *Window* and *Door* surfaces are deduced from *InteriorWallSurfaces* as shown in Fig. 7.	Floor-Surface, Ceiling-Surface, Interior-WallSurface
IFCWall	*IFCWall* is converted into *multisurfaces*. The *multisurfaces* are translated to *WallSurfaces* in CityGML, which represent the exterior shell of the building and have no connection to the *Room* feature type as shown in Fig. 8.	Wall-Surfaces
IFCStairs, IFCBeam, IFCColumn	The *IFCStairs*, *IFCBeam*, and *IFCColumn,* are translated into multisurface boundary geometries in CityGML. Moreover, IFC elements, which are within a specific room are transformed into *IntBuildingInstallation*. (Currently in our transformation process, IFC elements, e.g., *IFCBeam* and *IFCColumn* which extend over more than one room or crossing the boundary to the exterior are transformed into *BuildingInstallation* (in future we will rectify this drawback and will transformed into IntBuildingInstallation).	IntBuilding-Installation,

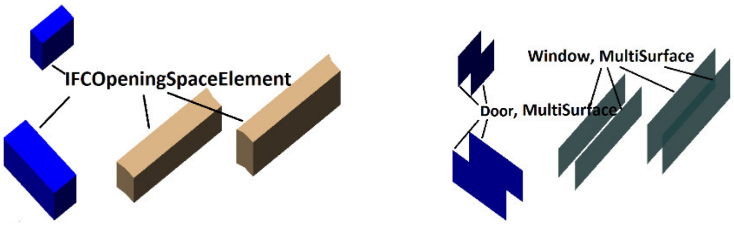

Fig. 5. Transformation of *IFCOpeningElement* to *Window or Door MultiSurfaces*.

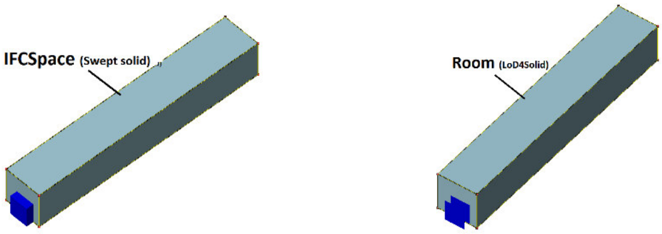

Fig. 6. Transformation of *IFCSpace* to *Room* feature type.

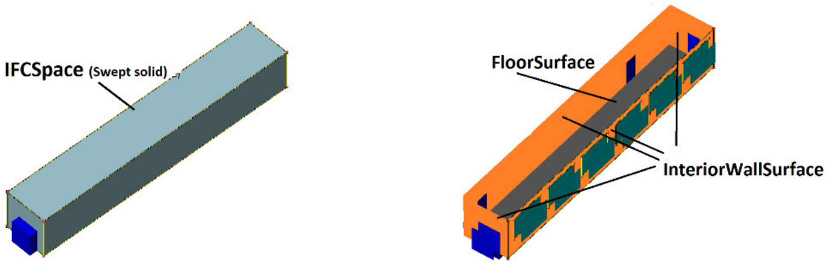

Fig. 7. Transformation from *IFCSpace* to *InteriorWallSurfaces, CeilingSurfaces* (not shown here), and *FloorSurfaces.*

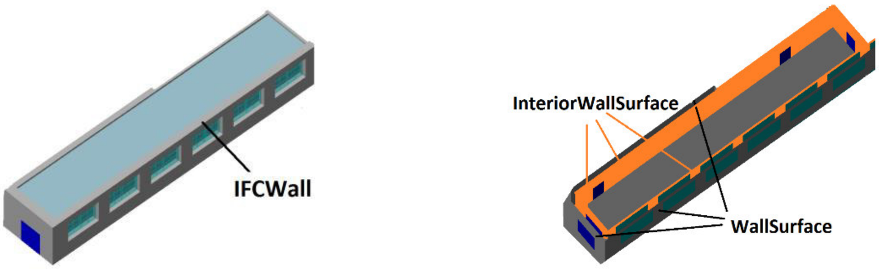

Fig. 8. Transformation from *IFCWall* to *WallSurface*.

IFCSlab objects in the IFC model are not converted to CityGML. They are only used to compute the ground surface and the roof surface of building in CityGML.

The transformation steps discussed in Table 1 were tested for different datasets and the results were found correct. The result of TUM main building's transformation from IFC dataset (shown in Fig. 9) to CityGML is shown in Fig.10.

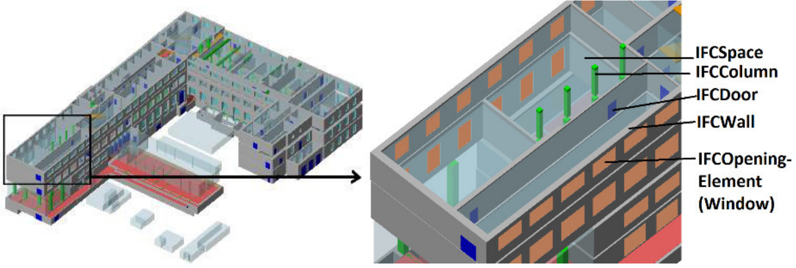

Fig. 9. (A) A part of the 3D model of TUM main building represented in IFC (left). (B). Detail view of a room and a corridor (right).

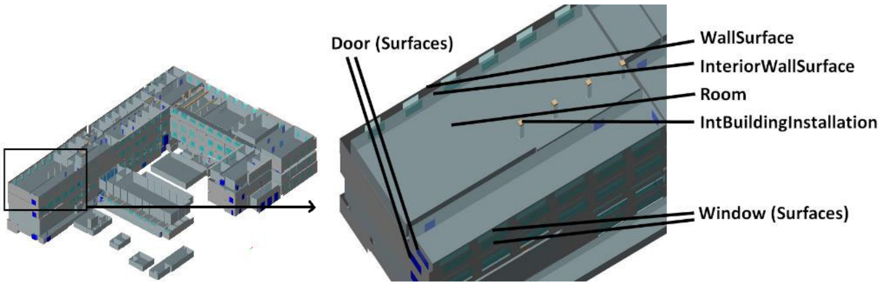

Fig. 10. A part of the 3D building model of TUM represented in CityGML after transformation (Ceiling Surfaces are removed for visualization purpose) (left). Detail view of a room and a corridor (right).

3.2 Transformation from CityGML to IndoorGML

IndoorGML is defined as an independent data model from the different approaches to building modelling, e.g., CityGML or IFC (IndoorGML 2014). Therefore, the main topographic space layer in IndoorGML can be represented using the input from a 3D building model that is represented either in CityGML or IFC or from any other information model describing the interior structure of a building. As discussed in section 2.3 we need to translate a 3D building model represented in CityGML into IndoorGML to obtain the main topographic space layer and to be able to compute subspaces according to different locomotion types. The details about CellSpace, CellBoundary, the structure model of each space layer, and the integration of multilayers can be found in (Becker et al. 2009a, IndoorGML 2014).The transformation mappings and steps to transform between the elements of CityGML LoD4 and IndoorGML are explicated in Table 2.

Table 2. Transformation mappings between conceptual CityGML and IndoorGML classes (continued on next page).

CityGML Feature Types	Transformation details	IndoorGML Elements
Room	*Room* geometry having *GM_Composite-Surface* and *GM_MultiSurface* is enforced to be a closed volume and translated into a Solid in IndoorGML	CellSpace
Door	*MultiSurfaces* representing a single Door are converted into a closed volume (Solid) in IndoorGML	CellSpace
Window	*MultiSurfaces* representing a single Window are converted into a closed volume (Solids)	CellSpace
Door as a Surface	A surface representing a Door is translated into a 3D boundary geometry in IndoorGML.	CellBoundary
Window as a Surface	A surface representing a Window is translated into a 3D boundary geometry in IndoorGML.	CellBoundary
Interior-WallSurface	An *InteriorWallSurface* representing the boundary surface of a room in CityGML is translated into a 3D boundary geometry (*CellBoundary*) of the incident room *CellSpace* in IndoorGML.	CellBoundary
FloorSurface	A *FloorSurface* representing the boundary surface of a room is converted into a 3D boundary geometry (*CellBoundary*) of the incident room CellSpace.	CellBoundary
Ceiling -Surface	A *CeilingSurface* representing the boundary surface of a room is converted into a 3D boundary geometry (*CellBoundary*) of the incident room CellSpace.	CellBoundary
Closure -Surfaces	Objects sealed using *ClosureSurfaces* are converted into a closed volume (Solid) in IndoorGML. Simultaneously, surfaces are converted into 3D boundary geometries of objects.	CellSpace and CellBoundary
BuildingFurniture, BuildingInstallation, IntBuildingInstallation	*BuildingFurniture, BuildingInstallation*, and *IntBuildingInstallation* represented by *MultiSurfaces* are converted into closed geometries (Solid) in IndoorGML.	CellSpace

WallSurface, *RoofSurface*, and *GroundSurface* objects are treated as outer *CellSpace* objects in IndoorGML and their geometries are not translated. Furthermore, each feature type in the CityGML LoD4 3D building model is translated into either a *CellSpace* or a *CellBoundary* geometry in IndoorGML with all the related attributes as described by table 2. Afterwards, the dual space geometries including state geometries (nodes) and transition geometries (edges) representing *CellSpaces* and *CellBoundary* in primal space respectively are computed to generate a space layer based on the MLSEM's method.

The steps defined in table 2 were implemented as a FME workspace and tested on of TUM main building model to translate from CityGML to IndoorGML. The resulting IndoorGML model is shown in Fig. 11.

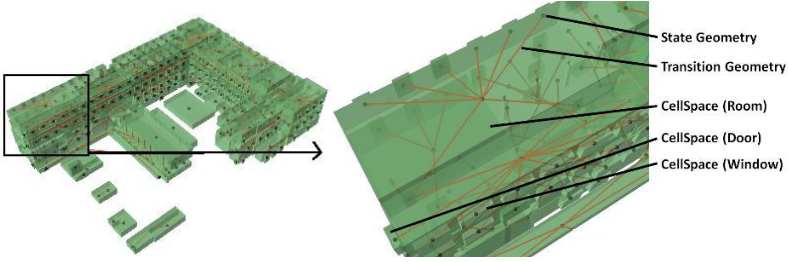

Fig. 11. Excerpt of the 3D model of TUM main building represented in IndoorGML (Space-Cell, State geometries, and Transition geometries) after transformation from CityGML model (left). Detail view of a room and a corridor (right).

4 Deriving routing graphs for different locomotion types

4.1 Subspacing approach

After having derived the IndoorGML building model either from IFC or CityGML, in next step, we compute the routing graphs for the different types of locomotion based on their specific navigating physical constraints. For each type of locomotion, i.e., flying, driving, and walking we consider an example based on its common usage in indoor environment. Those include Unmanned Aerial Vehicle (UAV), wheelchair, and a walking person respectively. The indoor navigation constraints of each locomotion type are based on the locomotion type's constraints model defined in Khan and Kolbe (2012).

In the field of robotics path planning, the mapping from work space to configuration space to determine a safe route for a rigid object resemble to a route for a point through the configuration space map. This approach has withdrawn the requirement for 2D or 3D collision detection and simplifies the path planning problem to finding a line that connects the start and target configurations avoiding the unsafe space. It also distinguishes the work space into three categories based on two solid objects which cannot overlap: obstacle configurations, in which objects will overlap; safe or free configurations, in which no overlap occurs and contact surface configurations, in which two or more objects touch each other (Lozano-Perez, 1983). This method is not specific to the robotics but also has been applied in the areas of construction, auto mechanics, etc. (Wise and Bowyer, 2000). Considering the simplicity, accuracy, and application of this approach in different fields we intend to compute the navigable spaces for the locomotion types through configuration space mappings. In a 3D environment we considered the generalized geometric models of a flying object, a walking person, and a wheelchair as 3D sphere and cylinders respectively along with their specific navigating physical constraints. The computation of the configuration space mapping was carried out based on Minkowaski’s sum method (Varadhan and Manocha 2006).

The decision to determine a specific element of the indoor space as navigable or non-navigable for the given locomotion type is taken by considering the physical navigating constraints of the locomotion type and spatial information (semantic, geometric, and topology information) of the element. The indoor space element, which is determined as non-navigable, will determine obstacle space around it to be deducted from the free space.

4.2 Example scenario

Consider a 3D building model containing a corridor and a room that contains four columns. The representation of building elements in CityGML and corresponding representation in IndoorGML are presented in Fig. 12 (A) and Fig. 12 (B) respectively. The extraction of a network model from the building as main topographic space layer in IndoorGML is shown in Fig. 13 (A). Most of the methods compute the navigable subspace for the locomotion type using constraints of the indoor space at the graph level. For example, the navigable space for the wheelchair shown in Fig. 13 (B) is computed considering its capabilities and constraints of the indoor space from the network model shown in Fig. 13 (A). The decision of the navigability of each element of building, e.g., a door shown in Fig. 13 (A) is taken after considering its spatial properties, i.e., length and width. If the length and width of the door is greater than the length and width of the wheelchair, then the door is considered to be navigable.

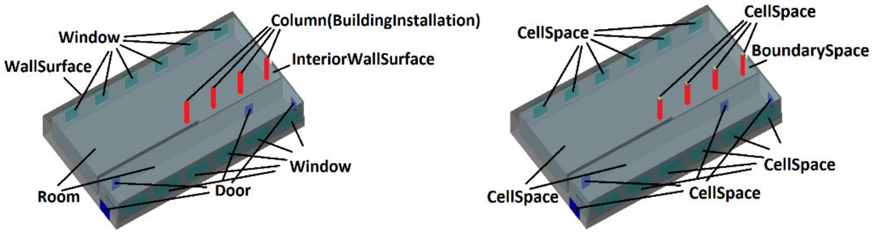

Fig. 12. (A) 3D building model in CityGML (left) (B) 3D building model in IndoorGML (right)

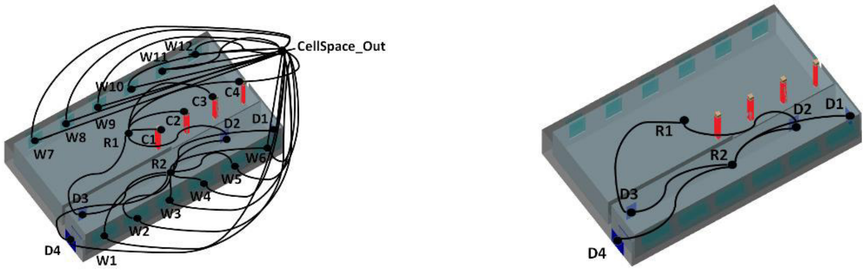

Fig. 13. (A) In IndoorGML, the main topographic layer of the 3D building model (left). (B) Navigable subspace computed based on network model according to wheelchair navigation (right).

The network model and the subspace building model representing the navigable space for the wheelchair shown in Fig 13 (B) is not enough precise for approximating the reasonable navigable space. Because, there are other locomotion types (e.g. flying) which may require precise or the detail geometric indoor navigable space so they avoid collision with the obstacles (e.g. column) located in the room. Therefore, we have to compute the actual navigable space after deducing the obstacle space (non-navigable space created by obstacles).

The physical constraints of the wheelchair are considered to determine the obstacles according to the constraints model defined in Khan and Kolbe (2012). For example, considering a wheelchair and its navigation constraints, a decision to determine navigability of the specific element is taken after considering all its properties. If there is a free space element from indoor space then we consider a collection of constraints for a wheelchair which need to be fulfilled to declare the free space navigable for the wheelchair. In this example, as a first step, we consider the *ScaleGeometryRelatedConstraint* of the locomotion type, according to this constraint it needs volume of 1 meter cubic or more free space to navigate. Furthermore, we have to consider more constraints as given in (Khan and Kolbe 2013), some of them are shown as example in Fig. 14 and they are combined through *complexlocomotionconstraint* operator “and”. So they all need to be fulfilled to determine free space navigable for the wheelchair. The next constraint is the *DirectionalGeometryRelatedConstraint,* which requires the wheelchair to have a surface to be held on or the free space must have a floor surface. Once that constraint is fulfilled, the free space element is checked for *NotConsiderConstraint,* whether the indoor element is “Window” in this case as it is not window so it become irrelevant to be fulfilled, otherwise if it is window then it is determined as non-navigable. Then in the next step, the *TopologicalGeometryRelatedConstraint* is considered which emphasizes that the free space must fulfill the requirement to be navigated “within” with the geometry of locomotion type. If the free space has enough space to contain locomotion type within then that free space element is navigable otherwise it determines as non-navigable. In further realization of constraints of the wheelchair on free space element of indoor space, the *CapacityContraints* are considered that include *CrossThrough and PassOn,* in this case, the wheelchair is evaluated if it has the capacity to cross through free space and pass on floor surface of the free space, the free space is computed as navigable otherwise non-navigable.

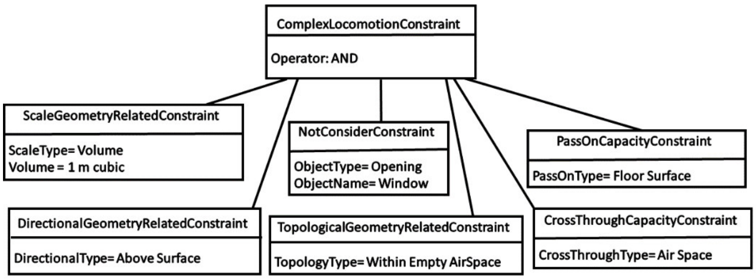

Fig. 14. An example of a complex locomotion constraint for computing navigable subspace for a wheelchair formed by aggregating subconstraints.

In this example, after considering constraints from Fig. 14 we determined the free space and door spaces as navigable for the wheelchair. Furthermore, considering other constraints from constraints model of the locomotion type we declare columns, windows, and walls of the room as non-navigable. The non-navigable spaces (e.g. columns, windows, and walls) will determine obstacle spaces based on Minkowski's sum as shown in Fig. 15 (A) in pink color. The actual navigable space is determined after deducing the obstacle space as shown in Fig. 15 (A) in green color. Furthermore, the route graph for the wheelchair is formed using the IndoorML method (Poincare duality) from the actual navigable space as shown in Fig. 15 (B).

Fig.15. (A). Actual Navigable spaces after deducting obstacle spaces according to the wheelchair shown in green color (left). (B). Network model of the corresponding navigable space (right).

The difference between navigable space that is computed for a wheelchair through graph based approaches and free or safe navigable space which is computed through configuration space (Lozano-Perez 1983) approach can be observed in Fig. 13 (B) and Fig. 15 (B) respectively. The navigable space computed using configuration space is more precise particularly giving geometric details of non-navigable space around the columns which is not possible to represent through graph based approaches.

4.3 Discussion

We demonstrate in section 4.2 why it is important to derive geometric subspaces for the different locomotion types in contrast to many other approaches (Meijers et al. 2005, Stoffel et al. 2007, Dudas et al. 2009, Lertlakkhanaku and Soyoung 2009, Petrenko et al. 2014) where subspaces are computed only on a graph model level. The navigable space that is computed through graph based approaches, in essence, used only some geometric position (centroid of the object) and connection information between spatial objects (topological graph). The semantic information (e.g. types of spaces, and properties of building components) and the actual geometry of the object have not been considered yet. In contrast, the subspacing we carry out through the configuration space approach uses fully geometric and semantic information from a semantic 3D building model. In addition, if there are obstacles within an indoor space (e.g. column), the methods based on the graphs will fail or be not precise enough for approximating the reasonable navigable space, which may limit the path planning in many route planning applications.

From the brief discussion and comparison above it is apparent that it is necessary to compute the accurate subspaces at the geometric level for the given locomotion type and to extract the network models from the navigable space.

4.4 Implementation

We tested our methods on the complex TUM main building model, the semantic 3D model of which is available as an IFC dataset as shown in Fig. 9. The IFC dataset is translated into a CityGML dataset and further into IndoorGML as showed in Fig. 10 and Fig. 11 respectively. To compute the subspaces according to different locomotion types the obstacles for the given locomotion type are determined based on their constraint models. After determining the obstacles the unsafe regions around each obstacle are computed based on Minkowski sum (Barki et al. 2009), e.g., for the wheelchair the obstacle space is shown in Fig.16. The network graphs are extracted from the actual navigable spaces (calculated after deducing the obstacle spaces) to represent the subspaces for the given type of locomotion. Fig. 17 and Fig. 18 show the resulting network models for the navigable spaces for UAVs and wheelchairs respectively. The results of the subspacing of the main topographic space according to different locomotion types show that they significantly differ from each other.

The process of transformation is implemented using the ETL tool FME Workbench (FME 2014) and the implementation of subspacing by a Java program in combination with Oracle spatial DBMS and Esri ArcObjects (Oracle 2014, ArcObjects 2014). The automation and proof of concept on error-prone real-world data show that our steps of transformation are simple to implement and easy to adjust to deal with the flexibility of IFC input data.

Furthermore, our experience shows that normally the dataset, e.g., IFC 3D building models, provided by Construction Engineering and Design community have several topological issues, e.g., geometries are overlapping or have unnecessary gap to directly generate CityGML or IndoorGML models. Therefore, before using those models, we recommend to perform topology checks on the building elements as we did as part of the transformation process (see section 3.1).

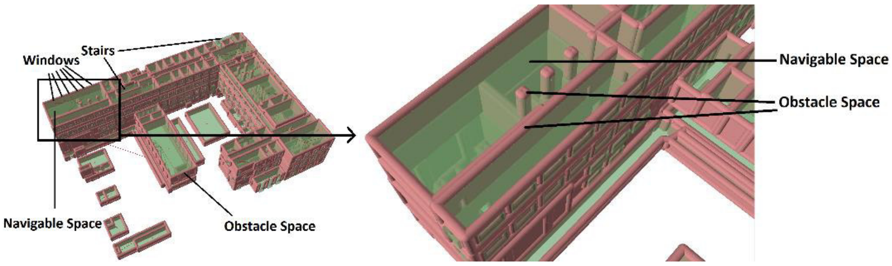

Fig. 16. Obstacle spaces (Unsafe regions) around obstacles (in pink) and navigable spaces for the wheelchair (in green) (left). Detail view of a room and a corridor (right).

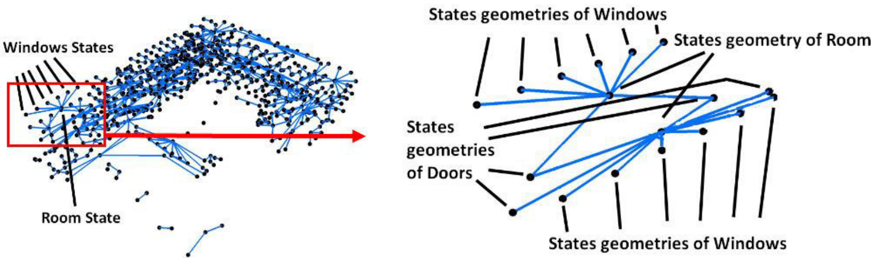

Fig. 17. The dual space of the UAV navigable subspace layer (left). Detail view of a room and a corridor (right).

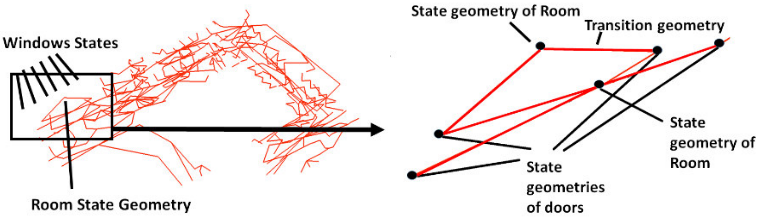

Fig. 18. The dual space of the wheelchair navigable subspace layer (state geometries of windows and stairs spaces are missing (non-navigable)) (left). Detail view of a room and a corridor (right).

5 Conclusions

We investigated a multi-step transformation process and demonstrated that IndoorGML datasets can be automatically derived from existing semantic 3D building models structured according to IFC or CityGML to support indoor navigation for different types of locomotion.

Dividing the overall transformation process into the two steps “IFC to CityGML LoD4” and “CityGML LoD4 to IndoorGML” has major advantages as follows: First, the workflow allows the source data to be structured either according to IFC or CityGML. Second, the transformation procedure from IFC to CityGML can be kept quite flexible accounting for the high degree of flexibility offered by IFC for structuring building models whereas the CityGML-to-IndoorGML transformation has fixed and simpler transformation rules. Furthermore, we presented the subspacing approach and demonstrated it for a public building using IndoorGML taking into account different locomotion types. The subspaces are computed using the real 3D geometry based on configuration space method and then network models are extracted. The subspaces created at the geometric level are more precise and consider semantic and geometric information of 3D building model making our approach different from other approaches.

The detailed representation of the 3D building model’s elements (e.g. detail representation of stairs free space) and their topology checking support to extract correct and detailed graphs for indoor navigation. Overall, the automation of the transformation process and the subspacing to support different types of locomotion for the indoor navigation for a public building show that our methods simplify the process and help to avoid manual errors and demonstrated the feasibility of the approach.

6 References

1. ArcObject (2014), ESRI ArcObjects: http://resources.esri.com/help/9.3/arcgisengine/java/doc/b0a96bd8-fc78-4573-9a70-e108cf6a4580.htm, Accessed 14 June 2014
2. Barki H, Denis F, Dupont F (2009) Contributing vertices-based Minkowski sum computation of convex polyhedral. Computer-Aided Design 41(7):525-538
3. Becker T, Nagel C, Kolbe T H (2009) (a) 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, p 61-77
4. Becker T, Nagel C, Kolbe T H (2009) (b) Supporting contexts for indoor navigation using a multilayered space model. In: Tenth international conference on mobile data management: systems, services and middleware, Taipei, May 2009. IEEE, p 680-685
5. BuildingSMART: http://www.buildingsmart.org/. Accessed 13 June 2014
6. De Laat R, van Berlo L (2011) Integration of BIM and GIS: The development of the CityGML GeoBIM extension. In: Kolbe T H, König G, Nagel C (eds) Advances in 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, p 211-225
7. Dudas P M, Ghafourian M, Karimi H A (2009) ONALIN: ontology and algorithm for indoor routing. In: Proceedings of the Tenth International Conference on Mobile Data Management: Systems, Services, and Middleware (MDM09), Taipei, May 2009. IEEE, p 720-825
8. El-Mekawy M, Ostman A B, Hijazi I (2012) An evaluation of ifc-citygml unidirectional conversion. International Journal of Advanced Computer Science and Applications 3(5):159-171
9. El-Mekawy M, Östman A, Shahzad K (2011) Towards interoperating CityGML and IFC building models: a unified model based approach. In: Kolbe T H, König G, Nagel C (eds) Advances in 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, p 73-93
10. FME (2014), Safe softwares: www.safe.com, Accessed 13 June 2014
11. M, Zipf A (2011). Formal definition of a user-adaptive and length-optimal routing graph for complex indoor environments. Geo-Spatial Information Science 14(2):119-128
12. IndoorGML (2014)(a): Open Geospatial Consortium (OGC) IndoorGML draft. OpenGIS specification. OGC's document no. OGC 14-005r1, Version. v.0.9.0.
13. IndoorGML (2014)(b): IndoorGML: www.indoorgml.net, Accessed 13 June 2014
14. Isikdag U, Zlatanova S (2009) Towards defining a framework for automatic generation of buildings in CityGML using building Information Models. In: Lee J, Zlatanova S (eds) 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, Springer Berlin Heidelberg, p 79-96
15. Khan A A, Kolbe T H (2012) Constraints and their role in subspacing for the locomotion types in indoor navigation. In: Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sydney, Nov 2012. IEEE, p. 1-12
16. Khan A A, Kolbe T H (2013) Subspacing based on connected opening spaces and for different locomotion types using geometric and graph based representation in Multilayered Space-Event Model. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS), Vol II-2/W1p, p. 173-185

17. Kolbe T H (2009) Representing and exchanging 3D city models with CityGML. In: Lee J, Zlatanova S (eds) 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, Springer Berlin Heidelberg, p 15-31
18. Lee J (2004) A spatial access-oriented implementation of a 3-D GIS topological data model for urban entities. GeoInformatica 8 (3):237-264
19. Lertlakkhanaku J, Soyoung B (2009). GongPath Development of BIM based Indoor Pedestrian Navigation System. In: Proceedings of the Fifth International Joint Conference on INC, IMS and IDC, Seoul, Aug 2009. IEEE, p 382-388
20. Lin Y H, Liu Y S, Gao G, Han X G, Lai C Y, Gu M (2013) The IFC-based path planning for 3D indoor spaces. Advanced Engineering Informatics 27(2):189-205
21. Lozano-Perez T (1983) Spatial planning: A configuration space approach. Transactions on Computers IEEE 100(2):108-120
22. Meijers M, Zlatanova S, Pfeifer N (2005) 3D Geo-information indoors: structuring for evacuation. In: Proceedings of the 1st International ISPRS/EuroSDR/DGPF-Workshop on Next Generation 3D City Models (EuroSDRBonn), Bonn, Germany, 21–22 June 2005, p. 6
23. Nagel C, Stadler A, Kolbe T H (2009) Conceptual requirements for the automatic reconstruction of building information models from uninterpreted 3D models. In: ISPRS Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS), Vol XXXVIII-3-4/C3
24. Nagel C, Becker T, Kaden R, Li K, Lee J, Kolbe T H (2010) Requirements and space-event modeling for indoor navigation. Tech. Rep. OGC 10-191r1, Open Geospatial Consortium, Discussion Paper.
25. Oracle database 11g: www.oracle.com, Accessed 14 June 2014
26. Petrenko A, Sizo A, Qian W, Knowles A D, Tavassolian A, Stanley K, Bell S (2014) Exploring mobility indoors: an application of sensor-based and GIS systems. Transactions in GIS 18:351–369, doi: 10.1111/tgis.12102
27. Steuer H (2013) High precision 3D indoor routing on reduced visibility graphs. In: Krisp, Jukka M. (ed) Progress in Location-Based Services, Lecture Notes in Geoinformation and Cartography Springer, Berlin, p 265-275
28. Stoffel E P, Lorenz B, Ohlbach H J (2007) Towards a semantic spatial model for pedestrian indoor navigation. In: Rolland C, Trujillo J, Yu E, Zimlanyi E, (eds) Advances in Conceptual Modelling-Foundiations and Applications, Lecture Notes in Computer Science, vol 4802, Springer, Berlin, p 328-337
29. Sahlemariam Y, Ahn S, Ko H (2008) Context based pathfinder for personalized indoor navigation. Master thesis. Korea Institute of Science and Technology.
30. Tsetsos V, Anagnostopoulos C, Kikiras P, Hasiotis P, Hadjiefthymiades S (2005) A human-centered semantic navigation system for indoor environments. In: Proceedings of the International Conference on Pervasive Services *ICPS'05*, July 2005. IEEE, p 146-155
31. Varadhan G, Manocha D (2006) Accurate Minkowski sum approximation of polyhedral models. Graphical Models 68(4):343-355
32. Wise K D, Bowyer A (2000) A survey of global configuration-space mapping techniques for a single robot in a static environment. The International Journal of Robotics Research 19:762