3D GEO-INFORMATION INDOORS: STRUCTURING FOR EVACUATION

M. Meijers[1], S. Zlatanova[1] and N. Pfeifer[2]

Delft University of Technology, the Netherlands
[1]OTB Research Institute, Section GIS-technology
b.m.meijers@student.tudelft.nl, s.zlatanova@otb.tudelft.nl
[2]Delft Institute of Earth Observation and Space systems
n.pfeifer@lr.tudelft.nl

Commission VI, WG VI/4

KEY WORDS: Three-dimensional, urban, data structures, databases, laser scanning, algorithms

ABSTRACT:

Interiors of buildings are often represented as two-dimensional spaces with attributes attached to them. Examples can be found everywhere, from architectural design plans to maps showing evacuation routes in emergency cases. Most of the navigation programs use primarily 2D plans for visualization and communication. Some exceptions are historical buildings and museums, which do offer navigation through textured, detailed 3D indoor models. However, these models are not structured with respect to the functionality of the building but largely with respect to the navigation/visualisation route.

Structuring the interior is not a straight forward task, as many concurrent decompositions of a building have to be accommodated (e.g., functional, according to building structure, accessibility-wise), which are very dependent on the application. Furthermore the semantic/terminology is still not unified and in many cases ambiguous. A very typical example are inner open-air courts, which are open-air but inside the building.

In this paper we present a semantic model of interior spaces. The structuring aims at facilitating calculation of evacuation routes and therefore particular building characteristics related to exit possibilities such as available doors and windows are considered as leading in the followed approach. The classification concept is used to build a graph for one of the buildings in the campus area. The geometry of the building and corresponding graph are organised in a geo-database management system (geo-DBMS)

1. INTRODUCTION

3D models of interiors have been created predominantly for visualisation purposes to guide users in buildings of historical (cathedrals, castles), architectural, cultural (museum) or other (airports) importance. These models are mostly not structured except for visualisation and navigation purposes. Usually the structure is built to speed up visualisation/navigation process.

In the last few years the attention in modelling interiors increases. Several tragic incidents have already suggested the current systems for evacuating people from large public and business buildings is not efficient (Cutter et al, 2003, Pu and Zlatanova, 2005). The research currently is on *intelligent buildings systems* to monitor and integrate data from a variety of sensors (temperature, movement or occupancy sensors, pressure pads, smoke or gas detectors, fire detectors) for detection, alarming and controlling other systems (heating, ventilation, air flow, lock/unlock doors, lighting, etc. (Reyes et al 2001). However, most of the existing systems are based on 2D environments, which reveal serious limitations when considering multi-level structures (Kwam and Lee, 2005, Zlatanova et al 2004). A building is represented as a combination of different types of spaces (rooms, compartments and different connection) but in 2D (Gillieron and Merminod, 2003).

In this paper we report a semantic model representing 3D structuring of interiors to be used for an intelligent computation of evacuation routes. The model consist of two levels (*polygon* and *section*), which take into account the possibilities to move through buildings. A semantic classification of polygons with respect to different characteristics is introduced as the first level (i.e. *polygon level*). Sections are specified at the second level (i.e. *section level*) on the basis of the polygon classification. Particular polygons and sections are accordingly mapped to nodes and edges of graph, which is used for routes calculations. The model is implemented in geo-DBMS using spatial data types and network possibilities of Oracle Spatial 10g.

The paper is organised in six sections. Section 2 presents the polygon and space partitioning. Section 3 discusses the process of graph creation. Section 4 gives a short overview on the network model of Oracle Spatial and presents the database implementation. Section 5 concentrates on the used data sets, implemented algorithms and the tests currently performed. Last section addresses future work and development.

2. SEMANTIC MODEL FOR EVACUATION FROM BUILDINGS

To be able to define meaningful routes to move within buildings, doors and other eventual exits, which can be used only in extreme cases, have to be explicitly modelled. Since the most common representation of these objects in 3D models is with polygons, we have use polygon as the smallest unit in our model. All the polygons in a building are given meaning with respect to their role in allowing people to walk around. Most of the polygons are just concrete walls and it is clear people cannot pass through them. However, there is a large variety of openings (with or without doors) that need elaborations.

The polygon classification presented here is based on four properties: *persistence*, *existence*, *access-granting* and *types of passing*:

1. The *persistency* of polygons reflects the possibility of a polygon to be temporarily removed (if needed). A very typical example of non-persistent polygon is a wall that can be folded and thus two spaces can be joined. This property of a polygon is further organized as a status operation 'being there' and 'not being there'.

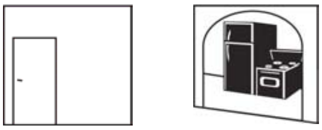

Figure 1: Persistent (door and wall, left) and virtual (virtual polygon closing the kitchen, right) polygons

2. On the basis of *existence* two types of polygons can be distinguished (Figure 1): *real* (e.g. walls) and *virtual* (e.g. virtual walls do not exist in reality but only in the model to close sections). This classification is critical for visualization process; virtual polygons should not be rendered.

3. We introduce two groups of polygons considering the *access-granting*. *Non-granting* polygons prohibit entering a section (e.g. wall). *Granting* polygons (Figure 2) allow three types of entering *full* (no conditions, these are the exits, e.g. polygons representing doors), *semi* (usually exists but have some restrictions, e.g. one needs to have a key) and *limited* (under normal circumstances not used as exist, e.g. emergency exist and windows).

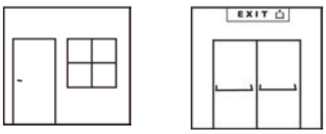

Figure 2: Granting polygons with full (door, left) and limited (window, left and exits, right) passing.

4. The *types of passing* can be *uni- and bi-directional* depending on the possible way of entering. For example many exits allow only one-way of entering. The consequence of such an entrance is that once inside a particular section the person may not get out using the same way.

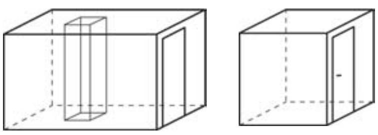

Figure 3: Non-accessible sections (columns inside a room, left) and end section (right)

The classification made for the section level is based on complete subdivision of space. The building is separated into relatively well-defined parts called *sections*. A section is defined as being the smallest amount of bounded space in a building that has a specific function (e.g. meeting room, stairs, etc.) with the following restrictions:

- Sections must be closed. If not closed in reality, virtual polygons must be introduced. Examples of such 'open' spaces are stairs, corridors, elevators, etc.
- Section must be distinct and may not overlap with any other section.

We distinguish between three types of sections: *end* (only one entrance/exit, Figure 3, right), *connector* (more than one entrance/exit) and *non- accessible* (no entrance/exit, Figure 3, left) sections. Connector sections are corridors, elevators, stairs, and sometimes rooms. Several sections may compose a *complex-of-section* (e.g. floor, Figure 4). A *building* is then an aggregation of sections and complexes of sections.

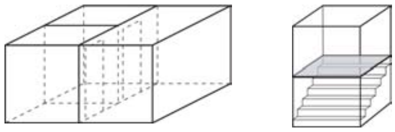

Figure 4: Complex of section (two end sections and one connector section (left) vertical complex of section (subdivided with a virtual polygon, right)

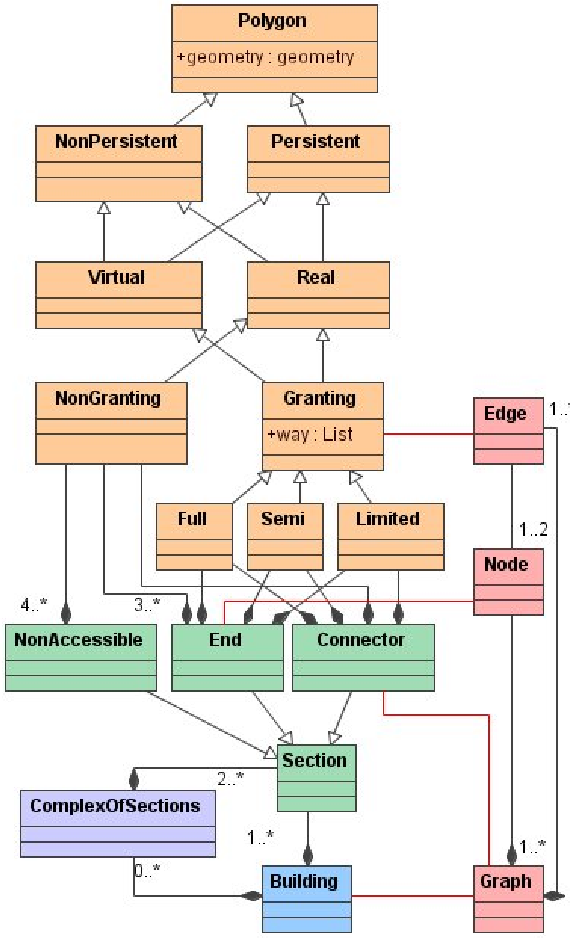

Figure 5: Semantic model for evacuation from buildings

UML diagram represents the relations between the different types of polygons, sections and complexes of sections (Figure 5).

Having the classification above, interesting interrelations can be observed. The type of polygons bounding a section has influence on (is related to) the type of section, i.e. only particular types of a polygon can be a part of some of the sections. For example *non-granting* polygons can be only *real* and close *non-assessable* sections only. Type of access and type of passing are critical polygon parameters for computing evacuation routes.

3. GRAPH DERIVATION

The semantic of the building is still not sufficient for route calculations due to lack of a connectivity information between the spaces. However the model allows for easy mapping to a graph representation. Graph model is often used to represent structure of buildings, e.g. as each room is represented by a node, and corridors are associated with edges (Gillieron and Merminod, 2004). Van Treeck and Rank, 2003 report an algorithm for an automatic graph derivation from a topologically structured data. The graph obtained however cannot be used for evacuation routes because edges are created through non-granting polygons.

In presented approach we construct the graph considering only *end section*, *granting polygon* and *connector section* (Figure 5) according to the following rules:

- an *end* section maps always to a node
- a *granting* polygon maps always to an edge.
- a *connector* section maps to a graph (i.e. a collection of nodes and edges has to be created).
- all remaining polygons and sections are not used to construct the graph.

Using this mapping, the nodes can also be seen as representative points for each end section (room), made explicit, e.g., by giving each node the 3D coordinates of a room center point. The edges can therefore also be seen as the way to move from one section into the adjacent section, describing the way people, or other mobile objects, move inside the building.

It is important to realise the role of the *connector* section. If the connector section is mapped to a node, people still may need to walk through walls (Figure 6a). Furthermore, when describing movements inside the building the trivial topological connectivity is not sufficient. People do not go first to the center of the corridor to reach the stairs (on both sides of the floor). Furthermore, evacuation routes may require following different paths, because sections of the building are not accessible or already very crowded. People behave different in different situations. While people may always take the shortest route in the case of an emergency, they tend to walk along walls in less stressful situations. Likewise, the number of people in a corridor will have influence on the path followed by the individuals. In Figure 6b the edges show also geometric content, their length denotes the distance for moving from one point (node) in the building to another. The length is not the only weight (a.k.a. label, property) that can be assigned to the edge, but the edges may also be associated with more general weight functions, e.g. corresponding to time or labour.

All the arguments given above clearly show why it is critical a connector section to be mapped into a graph. The following text discusses four methods to derive graphs for connector sections automatically. Some of these methods use knowledge on the room type, allowing considering application specific properties, others do not require additional knowledge, indicating that their usability for a certain application has to be ensured thereafter

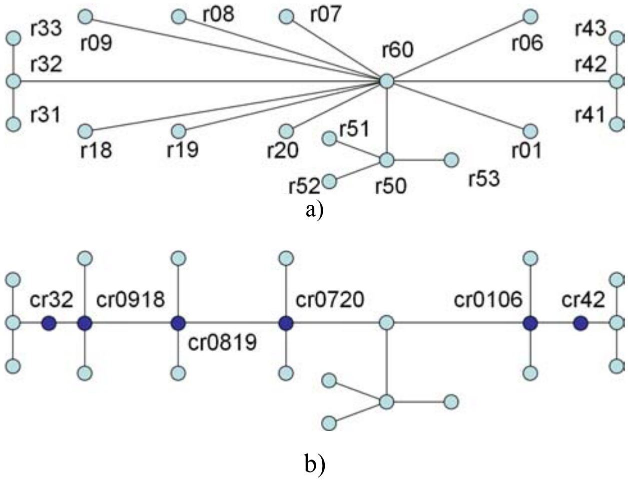

Figure 6: Graph models of a corridor in an office a) a graph where each section is a node and b) a graph where the corridor section (section r60) is represented also as a graph

The first method is quite appropriate for corridors, i.e. connector sections having for the ratio length/width a relatively large number. It uses an *adjusting line* to compute one centerline for the corridor. From every door (access granting plane) opening into the corridor a point in the corridor is created. This point corresponds to the point where people 'step into the corridor', and is computed by making one step (1m) from the door into the corridor. Starting from the door midpoint, the step distance in the direction orthogonal to the door plane, directed into the corridor, is used. This yields as many 'step into the corridor' points as doors and an adjusting line is fitted to these points. This line corresponds to the centerline of the corridor. The 'step into the corridor' points are projected orthogonally onto this line, yielding an ordered sequence of points. While this method works well in most corridor cases, it is not guaranteed to deliver centerlines which stay inside the corridor.

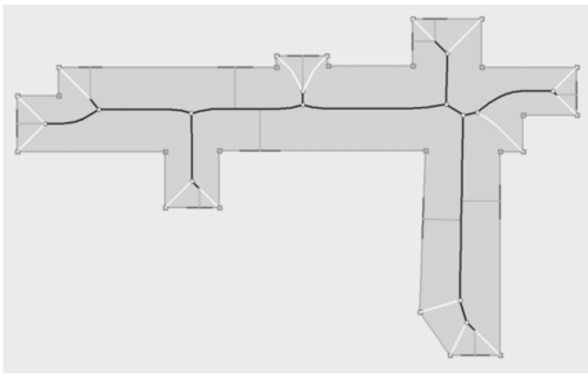

Figure 7 A corridor and its medial axis (black and white lines). Doors are drawn as bold lines, and the way from each door to the corridor center line in light gray. The medial axis was computed with the applet on http://www.lupinho.de/gishur/html/Skeleton.html.

Convex connector sections can be processed using a method based on *convex hull*. It can provide centerlines that always do remain inside the section. The convex hull is computed from the 'step into the corridor' points. The two points of the hull furthest apart form the centerline. All other points are projected onto the centerline and sorted as above. Alternatively, the *full graph*, connecting each node with each other node can be used. This yields edges that do not cross walls for convex sections only. The full graph includes also the shortest route from every door to every other door. Similar results will be obtained using a method based on the *Euclidean Minimum Spanning Tree*. This graph is the tree with shortest total edge length and every 'step into the corridor point' is connected to its closest 'step into the corridor point'.

The third method to be mentioned here provides correct results for any shape of the connector section can be based on the *medial axis*. The concept of the medial axis corresponds closely to the centerline of a corridor. The medial axis of a polygon is the set of all centers of circles i) touching the polygon in at least two points and ii) being completely contained inside the polygon. This definition does not require the polygon to be simply connected (i.e., it may have inner holes) and can also be applied to shapes which do not have straight but curved edges. It is also a 'continuous version' of the Vornoi diagram, where the generators are not only the polygon vertices, but every point on the polygon outline. First, the medial axis from the connector section is obtained (Figure 7). Then the graph is obtained by projecting the 'step into the corridor' points onto the medial axis. The 'step into the corridor' points are in this application in the middle of the doors, while the projection direction remains orthogonal to the door plane (Figure 7). Because of the definition of the medial axis, this is guaranteed to yield a point on the medial axis. This method has the advantage that it also works on concave shaped connector sections and for sections with columns inside. The medial axis 'takes care of this'. Note that the edges of this graph are in general curved.

The last alternative, which is not applicable in the context of persons trying to have short walking distances, is to maintain only *one node per connector section*. This is useful for checking accessibility via different rooms. This can also be useful when giving horizontal edges the weight zero, and other a weight depending on their slope or height difference. Shortest path computations then provide the route with the overall smallest slope or height difference.

4. GEO-DBMS STORAGE

Use of Geo-DBMSs in 3D modelling of buildings brings a lot of advantages. Besides the standard advantages of DBMS with respect to multi-user control, automatic locks during database transactions, advanced protocol mechanisms to prevent the loss of data, data security, data integrity and operations that comfortably retrieve, insert and update data, geo-DBMS offers efficient management of large spatial data sets (often encountered in 3D modelling).

Furthermore in geo-DBMS different models can be used, i.e. *geometry*, *topology*, and *graph* to maintain different representations for the . While the geometric structure provides direct access to the coordinates of individual objects, the topological structure encapsulates information about their spatial relationships. A geometry model (more or less compliant with OGC specifications) has been implemented in all mainstream DBMSs (e.g. Oracle Spatial, Informix, Ingress, PostGIS, MySQL). Although the implemented spatial data types are 2D, 3D objects still can be stored (Stoter and Zlatanova, 2003). Although topological implementation specifications are under development, topological structures are already available (Laser-Scan Radius and Oracle spatial 10g). A graph model is currently offered only by Oracle spatial.

The management of the semantic model as described in Section 2 and Section 3 requires utilisation of (at least) two types of structures: geometric model and graph model. While the geometric model will be used for extracting coordinates of the 3D polygons, which are needed for 3D visualisation, the graph model will be used to perform the algorithms for evacuation calculation. The semantic classification will provide further indications, which nodes and edges will be used, giving the dynamically changing situation in a building (accessibility of stairs, number of people, time to reach the exits, etc.

Oracle Spatial 10g maintains a combination of geometry model and graph model, i.e. the Network Data Model, which can become a quite appropriate structure for evacuation route calculation. Within this model three kinds of Networks can be created: a graph network (no geometry), a geometry network (has geometry) and a Linear Referencing System (LRS) geometry network (has point-line geometry with measure value). All of these networks can be hierarchical. The network consist of four tables:

- Node table (Table 1). Stores the nodes, columns of this table include id, name, type, partition, cost, and etc.
- Link table (Table 2). Stores links between nodes, columns of this table include link id, name, type, start node, end node, cost (can be used to represent length), and etc.
- Path node table stores sequence of nodes.
- Path link table maintains sequence of links.

Table 1 Columns of Node table in the Network model

Name	Example
NODE_ID	3
NODE_NAME	' Room 1'
NODE_TYPE	'Room'
ACTIVE	'Y'
PARTITION_ID	2
GEOMETRY	MDSYS.SDO_GEOMETRY (2003, NULL, NULL, SDO_ELEM_INFO_ARRAY (1, 1003, 3), SDO_ORDINATE_ARRAY (100,200, 500,700))

Table 2 Columns of Link table in the Network model

Name	Example
LINK_ID	2
LINK_NAME	'Corridor 1'
START_NODE_ID	3
END_NODE_ID	5
LINK_TYPE	'Corridor'
ACTIVE	'Y'
LINK_LEVEL	1
GEOMETRY	MDSYS.SDO_GEOMETRY (2003, NULL, NULL, SDO_ELEM_INFO_ARRAY (1, 1003, 3), SDO_ORDINATE_ARRAY (500,2100, 550,3500))
COST	6

Path table and path link table are optional, but pre-calculated paths (routes) can be stored there. Networks can be created by either PL/SQL or Java API, but can only be analyzed by Java API. A shortest path function is already available in the Java API, but only the costs (length) of links are considered. Based on the existing well-structured network model, we can easily implement user-defined java functions that take other problems that affect the evacuation route calculation into consideration

The proposed semantic model is mapped in 10 relational tables using the geometry data types. The graph organisation is still under development.

5. EXAMPLES AND TESTS

For demonstrating and testing the feasibility of the presented approach the graph of an office building will be analyzed. The building used is the Aerospace Faculty of TU Delft. Figure 8 shows the graph where each node corresponds to one section and each edge to an access granting plane. The building has 13 floors, where only the ground floor and the first floor have a different layout. Three elevators close to the middle of the building and two staircases, one at either end, allow vertical movement, the upper ten floors all have one large corridor for accessing the offices, elevators, and the staircases. The building has two entrances, visible as leaf notes on the bottom. Some simplifications were performed to keep the presentation of the model concise.

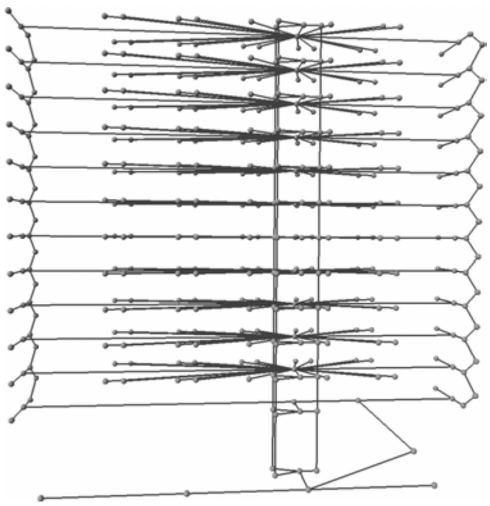

Figure 8: Graph of an office building with 13 floors.

The building graph has 362 nodes and 408 edges. Each corridor node of the floors 2 to 12, was replaced by a graph as described previously, using the *adjusting line* algorithm (as described in Section 3). This leads to a new graph with 593 nodes and 639 edges (not shown).

As a first experiment each edge was given its Euclidean length as weight. For simulating an emergency situation, the elevator nodes and edges have been removed. Dijkstra's algorithm was applied to compute the shortest way from each section to the open space. As there are two exits these two exits have been joined into another 'outdoor node' with edges of weight zero. This node is the root for the computation of shortest paths in Dijkstra's algorithm. The results can be seen in Figure 9, the virtual outdoor node and the edges incident to it are shown in grey. Naturally, this graph is a tree. Its use in indoor navigation is that it tells for each node which of the neighbouring nodes is closest to the exit without using the elevators. In Figure 8 it can be seen that there is only one stairway connecting the ground floor to the first floor, and it is lying closer to the staircases on the right side of the building. For most sections the shortest way out leads therefore over the right staircase.

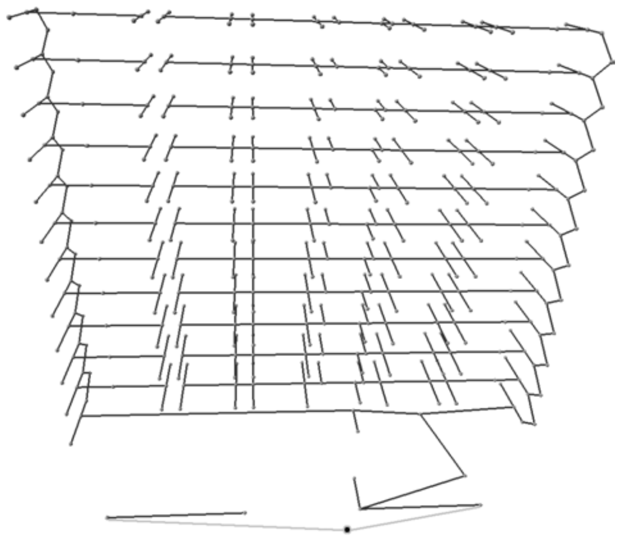

Figure 9: Shortest path from each section to the open space without using the elevator.

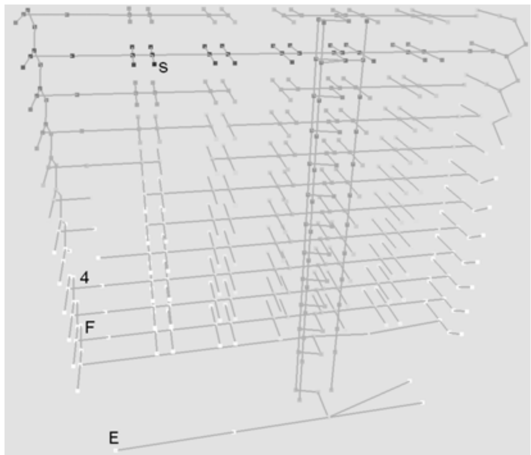

Figure 10: Time required to come from the start section (S) on floor 11 to all other sections is coded in the grey tone. E is the main exit, F the section furthest away, and 4 is a floor number.

In the previous example distances have been measured by Euclidean length. Alternatively, the time necessary for moving from one node to the next can be used to specify the weight of the edge. In such a scenario edges can be directed, giving different weights to upstairs and downstairs movement. For one section, the time required to reach each other section of the building was computed. Walking speed for horizontal movement was assumed to be $1ms^{-1}$, movement over stairs is slower, $0.5ms^{-1}$, for the elevator-to-elevator edges a faster speed was assumed, $2ms^{-1}$, but an average delay time of 6s was assigned to the edges connecting the elevator waiting sections to the corresponding elevator node. This results in an average waiting time of 12s and no delay when exiting the elevator. The resulting graph can be seen in Figure 10. The root (i.e. the starting room) is indicated with an S. The grey tone of each node corresponds to the time required to reach it. The time required to reach the main exit is 1 minute 37 seconds. All sections on the fourth floor or below are fastest reached by elevator. In fact it does not make a difference which elevator is used because the time required entering and exit is independent on the corridor. Dijkstra's algorithm always returns a tree and therefore – more or less randomly – selected one of the three elevators.

6. CONCLUSIONS AND RECOMMENDATIONS

This paper presented fractions of our research on intelligent evacuation routes calculation, i.e. the semantic model and the graph derived out of it. The major benefit of the model is the possibility for an automatic graph creation. Moreover the polygon classification can be adapted to a data collection procedure (e.g. terrestrial laser scanning, not discussed here) as described in Meijers, 2005 and can facilitate the 3D modelling process. However, the model still requires refinements. A further elaboration is needed on the concept for granting polygons. For example it should be differentiated between windows and exits. The windows should be considered only in particular cases (e.g. only on first floor, in case of balcony, etc.). The concept of complex-of-sections is currently also kept relatively simple.

Our initial implementation results (semantic, geometry and graph in geo-DBMS) show promising results: shortest patch can be computed on the graph and the geometry of polygons (grouped in sections) can be used for 3D visualisation. Data organisation of connector sections has to be addressed in more details. The connector section is mapped to a graph (to compute realistic evacuation routes) but it might be more appropriate to have a node (with the geometry of the entire corridor) for visualisation purposes. An option will be usage of a hieratical graph, i.e. the connector section is mapped to a node (which will contain also the geometry of the connector section), which is further linked to a graph as described in section 3 and 5.

The test performed on the graph revealed the feasibility of computing evacuation routes by graph theory algorithms. However, for real emergency evacuation much more parameters have to be considered. While the shortest path for a single person can be computed with the method shown, a scenario where more people need to be evacuated within a short time also has to consider the capacity of each edge, i.e. the number of people that can move along a certain path simultaneously. For more realistic scenarios algorithms from operations research would have to be applied. The objective function to be minimized could be the average time required for people to leave the building or the time required for the last person to leave the building. Similarly, the example where the fastest way to move from one section to another in the building considering different speeds of movements used a simple model for the elevator, which does not hold in practice. For longer elevator travels the chances of a stop in between become larger, adding a further delay.

The next step is the complete organisation of the models in the network model of Oracle Spatial 10g. The goal is development of procedures for automatic graph creation and intelligent evacuation route calculation at a database level.

.

REFERENCES

Bell, J. M.,2003, An API for Location Aware Computing, Master Thesis, Massachusetts Institute of Technology January 2003,
URL: http://graphics.lcs.mit.edu/pubs/thesis_bellj.pdf

Cutter, S., D. B. Richardson and T. J. Wilbanks (eds), 2003, *The geographical dimensions of terrorism*, Routledge, New York and London, pp.75-117

Gillieron, P.-Y. and B. Merminod, 2003, Personal Navigation system for indoor applications in: *Proceedings of 11th IAIN World congress,* 21-24 October, Berlin, Germany, 15 p.

Gillieron, P.-Y., D. Buchel, I. Spassov and B. Merminod, 2004, Indoor navigation performance analysis, in: Proceedings of ENC GNSS 2004, Rotterdam, 9p.

Fritsch, D. , D. Klinec and S. Voz, 2001, NEXUS – positioning and data management concepts for location-aware application, *Computers, Environments and Urban Systems,* 25 (2001), pp. 279-291e

Kwan M.P. and J. Lee, 2005, Emergency response after 9/11: the potential of real-time 3D GIS for quich emergency response in micro-spatial environments, *Computers, Environments and Urban Systems,* 29 (2005), pp. 93-113

Lee, J., 2001, A 3D data model for representing topological relationships between spatial entities in built environments, *PhD Dissertation*, Department of Geography, The Ohio State University, Columbus, Ohio, USA

Meijers, M. 2005, Database management of Segmented Interior Point Clouds of Buildings, Report, available at http://www.gdmc.nl/publications/index.html

Pu S. and S. Zlatanova, 2005, Evacuation route calculation of inner buildings, in: PJM van Oosterom, S Zlatanova & EM Fendel (Eds.), *Geo-information for disaster management*, Springer Verlag, Heidelberg, pp. 1143-1161

Reyes, A. A. Barbr, V. Callaghan and G. Clarke, 2001, The integration of wireless, wired access and embedded agents in intelligent buildings, in: *5th World multi-conference on systems, cybernetics, and informatics*, 22-25 July Orlando, Florida.

Stoter, J. and S. Zlatanova, 2003, Visualisation and editing of 3D objects organised in a DBMS, in: *Proceedings of the EuroSDR Com V. Workshop on Visualisation and Rendering*, 22-24 January 2003, Enschede, The Netherlands, 16p

Van Treeck, C and E. Rank, Ernst, 2004, Analysis of building structure and topology based on Graph Theory, in: *International Conference on Computing in Civil and Building Engineering*, ICCCBE, 10, 2004.06.02-04, Weimar, Bauhaus-Universität,
URL: http://e-pub.uni-weimar.de/volltexte/2004/238/

Vosselman, G., B. Gorte, G. Sithole and T. Rabbani, 2004. Recognizing structure in laser scanner point clouds. In: *Proceedings of Conference on Laser scanners for Forest and Landscape assessment and instruments*, Freiburg, Germany

Zlatanova, S., P. van Oosterom and E. Verbree, 2004. 3D technology for improving Disaster Management: Geo-DBMS and positioning. In: *Proceedings of the XXth ISPRS congress*, Istanbul, Turkey, 12th - 24 July, 6 p.