RESEARCH Open Access

Reasoning about accessibility for disabled using building graph models based on BIM/IFC

Barbara Strug and Grażyna Ślusarczyk*

Abstract

Background: Building Information Modelling (BIM) is becoming a standard in Architecture, Building and Construction (ABC) industries. It provides a method for a unified representation of information about building throughout its lifecycle – from the design phase to the building maintenance. At the design phase the focus is usually on fulfilling the requirements and applicable norms and standards. One of the issues that is considered to be very important deals with the accessibility of buildings. While the design tools usually support the legal requirements for accessibility for disabled persons, little effort has been observed to address the quality of the access routes in terms of time, length and convenience of the route to be taken.

Methods: This paper introduces a method of searching routes accessible for disabled people in public buildings, which is based on the building-related knowledge. This knowledge is stored in graph-based representations of buildings. A graph model of the considered building is effectively created by extracting information about the building from the IFC file, which is an interoperable BIM standard. The information about the topology of spatial layouts of buildings is stored in the graph representation together with characteristics of elements ensuring access between spaces. This model allows for homogenous encoding of both spatial and non-spatial information of different types, which is necessary for route calculation. Attributes assigned to graph nodes encode types of spaces and their sizes. Attributes of graph edges represent costs of moving between spaces, which depend on width of openings, equipping stairs with ramps, door types and their opening directions. As for people moving around on wheelchairs easiness of passing a route between two places can depend on the route direction, the directed graphs are used to represent building layouts. The optimal routes accessible for disabled are specified using the modified algorithm of graph search for single-source minimum-cost paths, where costs of passing through different spaces and between them are considered. This approach is applied to find best routes to a chosen room starting from all possible building entrances.

Results: The proposed application offers intelligent knowledge-based navigation, which is based on the graph representation of buildings. The system is aimed to aid people to decide about the routes they should take using mobile devices. It enables them to select from all accessible routes the one which requires the lowest effort in traversing it. Apart from the user interface allowing for visualization the results of design analysis in respect to the accessible routes, a visual tool dedicated for designers, which shows on the floor layout the actual route a disabled person has to take, is proposed. Offering a designer such a visualization can shift the focus of the design from mere existence of accessible routes to existence of quality routes for disabled. The presented examples illustrate the proposed approach by depicting the found accessible routes on the building layout.

(Continued on next page)

* Correspondence: gslusarc@uj.edu.pl
Department of Physics, Astronomy and Applied Computer Science, Jagiellonian University, Kraków, Poland

(Continued from previous page)

Conclusions: Building-specific knowledge, which is extracted from the IFC file and stored in the graph building representation, is used to find the best routes accessible for disabled people. The search for these routes is performed by a knowledge-based graph-search procedure, which takes into consideration costs of passing through different spaces and between them. There is a possibility of adding in future more elements which can be treated as different obstacles in moving around the building.

Keywords: BIM, IFC, Knowledge representation, Graph structure, Route planning

Background

Architectural building designs are contemporary created with the use of CAD tools. BIM technology used for CAD applications enables to represent syntactic and semantic building information with respect to the entire life cycle of designed objects. The 3D object model is created using such elements as parameterized walls, ceilings, roofs, windows or doors (Eastman et al., 2011). The file format IFC (buildingSMART, 2013a) is an interoperable BIM standard for CAD applications, which supports a full range of data exchange among different disciplines and heterogeneous applications. Information retrieved from IFC files is used in applications estimating construction cost for tendering in China (Ma et al., 2011) managing construction sites (Hu & Zhang, 2011) or evaluating design solutions (Jeong & Ban, 2011).

IFC specifies virtual representations of building objects as well as their attributes and relationships. It includes most types of geometry, supports many classes of attributes (Eastman, 2012). Although IFC is an open standard, its complex nature makes information retrieval from an IFC model difficult (Zhang & Issa, 2011). The richness and redundancy of the full IFC model constitute an obstacle for particular use cases, as it cannot be fully supported by many specialized applications, and useful IFC subsets should be identified. Model View Definition (MVD), the Information Delivery Manual (IDM), and Semantic Exchange Modules (SEM) concepts have been introduced to efficiently reduce the scope of the full IFC model (Zhang et al., 2013). There is the on-going research on semantically precise and reusable model views and their implementations in real projects.

At the building design phase the focus is usually on fulfilling the requirements and applicable norms and standards (Vreeker et al., 2002; Trinius & Sjöström, 2005). One of the issues that is considered to be very important deals with the accessibility of buildings (Bright & Giulio, 2002). While the design tools usually support the legal requirements for accessibility for disabled persons (Iwarsson & Stahl, 2003; Sakkas & Perez, 2006), little effort has been observed to address the quality of the access routes in terms of time, length and comfort of the route to be taken (Andrade et al., 2012; Ozer & Sener 2013). There is also a lot of research related to searching for routes in respect to evacuation from buildings in case of a disaster (Cepolina, 2005; AlShboul et al., 2007; Papinigis et al., 2010; Yatim, 2012; Ronchi & Nilsson, 2013).

In this paper the problem of searching for the best route for people moving around in wheelchairs in public buildings is considered. The presented approach differs from methods described in the above mentioned works as our procedure does not search for the quickest/shortest way to a secure or appointed area (Zarrinmehr et al., 2013) but for the path first of all accessible, and as convenient as possible. Therefore in finding optimum routes for disabled such elements as existence of ramps, door types and door opening directions are considered.

In this paper reasoning about accessibility of selected rooms is based on building graph models extracted from Building Information Modelling (BIM) process. In (Pu & Zlatanova, 2005) a concept for evacuation route calculation based on 3D geometrical/topological models of buildings is described. In (Rüppel et al., 2010) an approach to model emergency situations in buildings based on BIM is described. However the process of generating graph networks out of BIM, on which route calculation can be performed, is difficult as it consists of merging several separately generated graphs. Therefore the proposed approach offers an effective way of creating a graph model of the considered building on the basis of the information retrieved from IFC files.

The proposed application is aimed to aid people to decide about the routes they should take using mobile devices. The application extracts the information about the building from the external IFC file and searches for best routes to a chosen room starting from all possible building entrances.

Methods

Representation of information extracted from IFC

In this section the problem of extracting knowledge about the building, in which routes convenient for disabled people are searched for, is considered. The information about the topology of spatial layouts of the building is retrieved from the IFC file. IFC supports XML format storage, so information can be retrieved from the IFC model by parsing the XML file. On the basis of IFC entities and relations between them the accessibility relations between building spaces are computed and stored in the graph structure. This structure allows an easy access to the retrieved information.

The graph-based building model allows for homogenous encoding both spatial and non-spatial information of different types, which is necessary for route calculation. The attributes assigned to graph nodes encode types of spaces (rooms, corridors, stairs, etc.) and their sizes. Attributes of graph edges represent costs of moving between spaces, which depend on width of openings, equipping stairs with ramps, door types and their opening directions. The topology of the building spaces extracted from IFC together with additional information about accessibility between them is used to search for the optimal routes.

IFC specifies different types of building entities and their basic properties. Information about the building, which is needed from the point of view of the problem being considered, includes topology of floor layouts, accessibility between spaces, height of the floors, stairs and doors specification. IFC entities, which store the data required by the proposed system, are of the types IfcSpace, IfcDoor, IfcWall and IfcStair. According to the IFC 2x Edition3 Model Implementation Guide and the IFC specification (buildingSMART, 2013b) the above mentioned classes can be described as follows:

- IfcSpace is the instance used to represent a space as the area or volume of a functional region. It is often associated with the class IfcBuildingStorey representing one floor (the building itself is an "aggregation" of several storeys) or with IfcSite, which represents the construction site. A space in the building is usually associated with certain functions (e.g., hall, bathroom). These functions are specified by attributes of the class IfcSpace (Name, LongName, Description).
- IfcWall is the instance used to represent a vertical element, which is to merge or split the space. In IFC files two representations of a wall can be distinguished. The subclass IfcWallStandardCase of IfcWall is used for all walls that do not change their thickness (the thickness of a wall is the sum of the materials used). IfcWall is used for all other walls, in particular for the walls with non-rectangular cross-sections.
- IfcStair represents a vertical passage allowing for moving from one floor to the other. It can contain an intermediate landing. Instances of IfcStairs are treated as containers, by which we refer to component elements as IfcStairFlight using IfcRelAggregates.
- IfcDoor represents a building element used to provide access to a specific area or room. Parameters of IfcDoors specify dimensions, an opening direction and a style of the door (IfcDoorStyle). IfcDoor is a subclass of IfcBuildingElement and a superclass of IfcDoorStandardCase. Door instances are usually located in a space IfcOpeningElement to which we refer by IfcRelFillsElement.

The above mentioned instances inherit from IfcProduct class, which allows for determining their positions using IfcLocalPlacement and PlacementRelTo attribute. The coordinates obtained in this way specify the relative position of the object to the other instances of the class IfcProduct. Obtaining the actual position of the considered IfcProduct instance is possible by tracking all references IfcLocalPlacement and PlacementRelTo, until the final one.

In our approach the spatial configuration of all building floor plan layouts is derived from the IFC file and stored in the graph. The relations between appropriate IFC entities required to compute the topological relationships of spaces are searched for. Two rooms are adjacent if two IfcSpace entities refer to the same IfcWall or to the same IfcWindowStandardCase using IfcRelSpaceBoundary relation (Langenhan et al., 2013). Two rooms are accessible if the wall between them has an opening or door. Therefore IfcWall or IfcWindowStandardCase entity should refer to IfcOpeningElement by

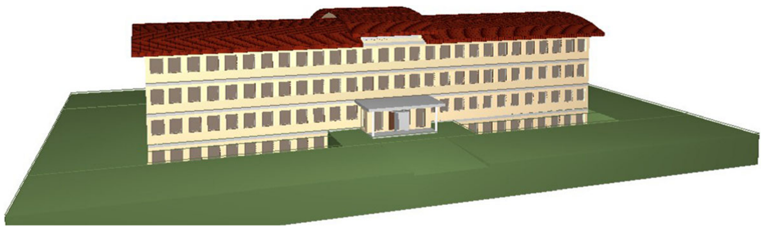


Fig. 1 A visualization of the IFC file information

IfcRelVoidsElement relation, or additionally IfcDoor entity should refer to IfcOpeningElement by IfcRelFillsElement relation (the opening is filled with a door). The extracted information is then saved in the graph structure.

We use attributed, labelled and edge-directed graphs. Graph nodes represent building spaces, while edges correspond to accessibility relations between these spaces and therefore represent doors, openings and accessibility between storeys through stairs/lifts. Labels assigned to graph nodes store names of spaces, while node attributes store other properties of spaces, for example their sizes or types. Attributes of graph edges represent costs of moving between spaces. The proposed system creates a three dimensional visualization of the building graph. The information about the building layout together with attributes representing costs of moving between spaces and through them is used to compute optimal routes accessible for disabled people.

The algorithm in pseudocode presented below generates a graph on the basis of the given IFC-Step file. It assumes the correctness of the file. Not all the details of the algorithm are included in the pseudocode but in general it works in the following way. Firstly, the list of all ifcSpace and ifcStairs is iterated and for each element a node is created in the graph using *AddNode* method. Additionally, if an element belongs to the class of ifcStairs elements and the stairs it represents are accessible for disabled users a node attribute *acc* is set to TRUE. Then for each ifcSpace element all its bounding ifcWall elements are found, and for each such an element all openings are collected. For each opening a position is saved to be used later for the calculation of the lookup table. Moreover, for each ifcWall element a space adjacent to the previously found space is found and two edges connecting these spaces are added to the graph using *AddEdge* method and appropriate values for the edge cost are also added. These values depend on the style of the doors between spaces and the direction in which they actually open. If two spaces are accessible through an open wall then the edge cost is set to 0. For each entrance door a node is created in the graph and the edges connecting it with the node representing the space to which they lead are added. At the end for each ifcSpace and ifcStairs element the lookup table, where the cost of passing between different openings in walls bounding the space, is computed using *ComputeLUT*$_r$ method.

Methods *AddNode* and *AddEdge* are responsible for adding graph atoms. *AddNode* creates a new graph node, adds it to the graph and labels it with the *name* value of the ifcSpace element. *Add Edge* creates a new edge, sets appropriate start and target nodes and adds it to the graph.

```
Algorithm
Input: IFC-STEP file
Output : g – graph
for r in ifcSpace or ifcStairs
 num = 0
 v(r) ← AddNode(g, r, num)
 num++
  if r in ifcStairs and r.accessible
    v(r).acc = TRUE
 for r in ifcSpace or ifcStairs
   i = 1
   for y in ifcWall where ifcRelSpaceBounding(r,y)
    for w in ifcOpeningElement where ifcRelVoidsElemet(y,w)
      pos[i] = w.position   // wall opening w
      i++
     for s in ifcSpace
      if ifcRelSpaceBounding(s,y)
        e1← AddEdge(g, v(r), v(s))
        e2← AddEdge(g, v(s), v(r))
      if d in ifcDoors where ifcRelFillsElement(d,w)
        if d.style = FireDoor
          if d.dir = (r,s)
            e1.cost = 2; e2.cost = 20
          else
            e1.cost = 20; e2.cost = 2
        if d.style = YardDoor
          if d.dir = (r,s)
            e1.cost = 1; e2.cost = 10
          else
            e1.cost = 10; e2.cost = 1
     {end for s}
   if d in ifcDoors where ifcRelFillsElement(d,w) and d.style = EntranceDoor
      x ← AddNode(g, d, num)
      num++
      e1 ← AddEdge(g, x, v(r))
      e2 ← AddEdge(g, v(r), x)
      LUT_x[1,1] = cost(d.type) // type depends on: stairs or ramp, door opening direction
  {end for w}
 {end for y}
 ComputeLUT_r
{end for r}
```

An example 3D visualization (obtained by DDS-CAD Viewer) of the information, which describes the

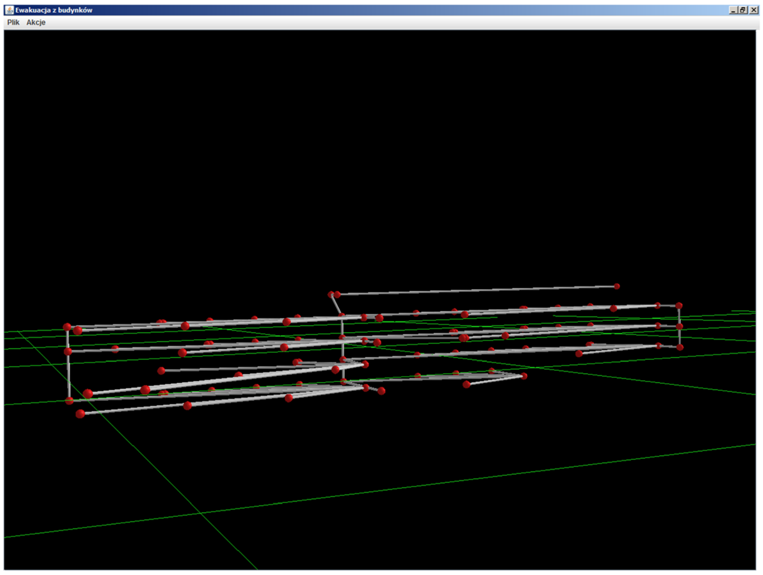


Fig. 2 A visualization of the graph representing the layout of the building from Fig. 1

building, included in the IFC file (Open IFC Repository, 2013) is shown in Fig. 1. A three dimensional visualization of the graph representing layout of this building is presented in Fig. 2. In this visualization labels and attributes are omitted, only the topological structure of the building is shown.

Graph representation of building-specific knowledge

The topology (structure) of the building is stored in the form of a labelled and attributed directed graph. Each space (room, corridor, etc.) is represented by a node in the graph, while edges represent connections between the spaces. Moreover to each edge connecting nodes an attribute representing a cost of passing between corresponding

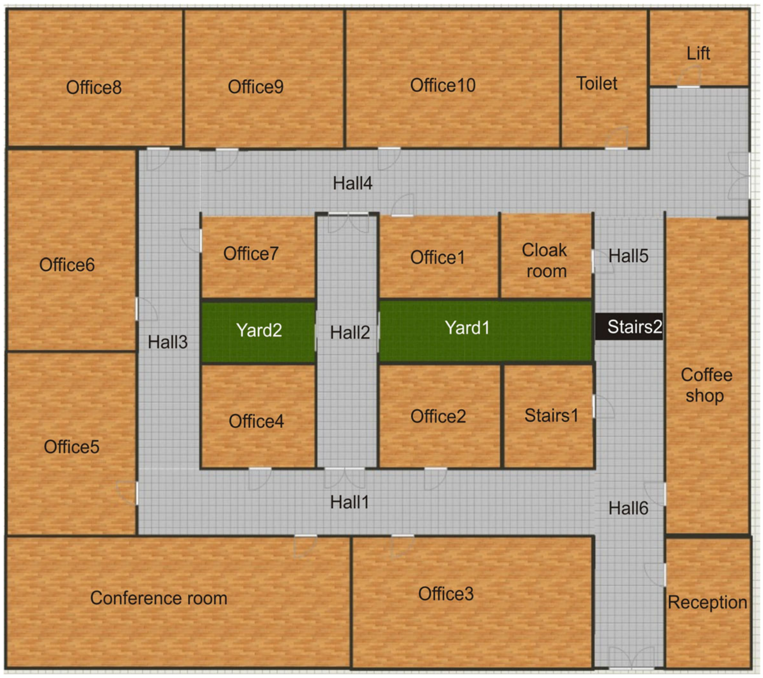


Fig. 3 A fragment of the ground floor layout of an office building

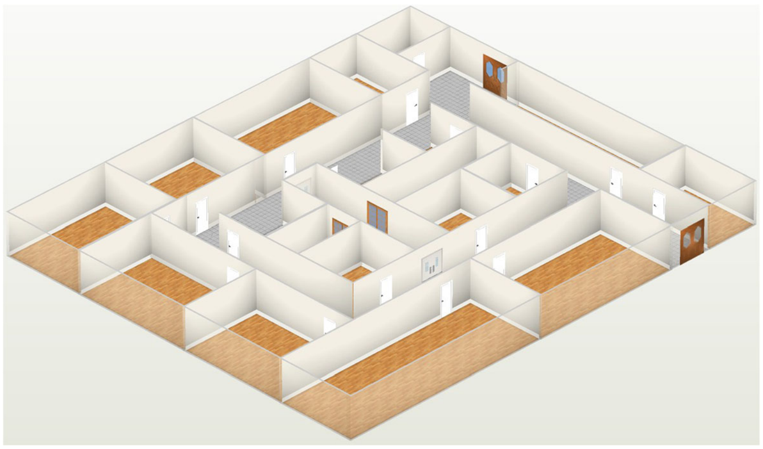


Fig. 4 A 3D view of the floor layout from Fig. 3

spaces is assigned. The value of this attribute is calculated as the cost of opening a door. It has to be noted that in some cases the cost of passing the same path but in different direction may be different, as the cost of pulling the inner door is different from the cost of pushing them. Thus the graph has to be directed with two edges representing passing spaces in both directions.

One of the attributes assigned to each node is a lookup table containing the costs of passing through the space corresponding to this node. Each space can have several entrance/leaving points, especially the ones like halls or corridors. Let us assume that the edges incident to each node are numbered and each number represents one entry/leaving point. The lookup table assigned to a given node v is an array $LUT_v[n,n]$, where each element $LUT_v[i,j]$ contains the cost of passing the space corresponding to the node v from the point i to the point j, and n is the maximum number of possible entry/leaving points. This cost is computed on the basis of distances between positions of points i and j.

A fragment of a floor layout of the ground floor of an office building is depicted in Fig. 3. The building has two entrances: the main one with the stairs and a side one with a ramp allowing access for people with movement difficulties. The layout is composed of several offices, a conference room, coffee shop, cloakroom, reception and yard. Moreover there are fire doors on both sides of the *Hall2* and stairs between *Hall5* and *Hall6*. These stairs make passing through east halls on a wheelchair impossible. There are also stairs and a lift connecting the ground floor with other storeys.

A 3D view of the floor layout from Fig. 3 is depicted in Fig. 4. It can be noticed that there are four types of doors: entrance doors, fire doors, yard doors and standard internal doors leading to majority of spaces. The yard doors and fire doors are harder to open than standard doors.

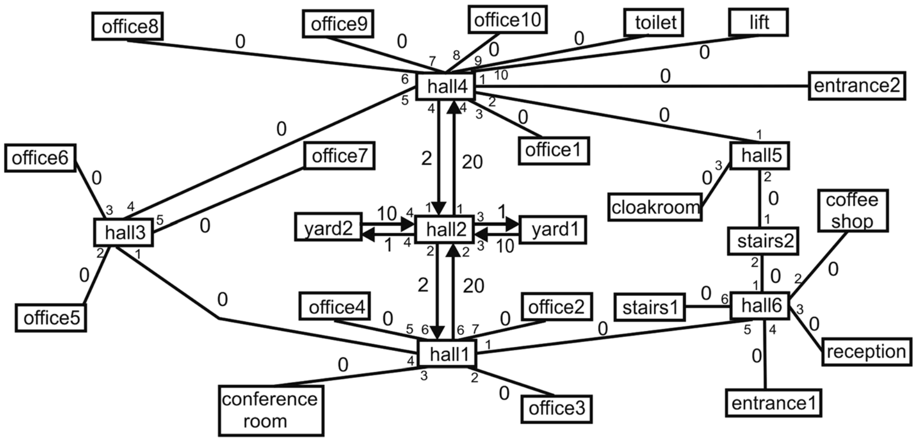


Fig. 5 An attributed labelled graph representing the floor layout from Fig. 3

The layout from Fig. 3 is represented by a graph shown in Fig. 5. The entrance points to each hall represented by edges incident to a node representing it are numbered. In case of two directed edges representing one entry point both edges have the same number. Two edges in this graph have the cost 2, while two others 20, as they represent the effort needed to push and pull, respectively, the fire doors on both ends of the *Hall2*. The cost of opening the yard doors is 1 and 10, respectively, as they are not as heavy as the fire doors. The cost of opening the standard doors is not taken into account as there is no possibility to avoid them.

Route search algorithm

The topology of the building spaces extracted from IFC together with additional information about accessibility between them is stored in the graph structure. This information is used by the system to search for the optimal routes for disabled people.

At the first step of the search process the start and the destination nodes are defined. To lower the computational cost all nodes with a single edge, which are not start and destination ones, are discarded from the graph, as there is no possibility of passing through the spaces they represent. The optimal routes are computed using a modified algorithm of graph search for single-source minimum-cost paths. The modification consists in taking into account not only edge costs but also the values stored in the lookup tables associated with nodes. In the relaxation step the shortest distance to the node l connected to the given node k is computed by summing the value $LUT_k[i,j]$ from the lookup table associated with the node k and the cost of the edge going from k to l. The value $LUT_k[i,j]$ of the lookup table defined for the node k represents the cost of passing through the space represented by the node k from the entry point i to the leaving point j. The number i represents the entry point (edge) through which the shortest path to the node k leads. The number of the leaving point of the node k assigned to the edge going to the node l is represented by the number j.

Table 1 The lookup table for *Hall4*

	1	2	3	4	5	6	7	8	9	10
1	0	5	15	17	22	22	20	15	5	2
2	5	0	10	12	17	17	15	10	2	7
3	15	10	0	2	7	7	5	2	12	17
4	17	12	2	0	5	5	3	6	16	23
5	22	17	7	5	0	2	4	11	20	25
6	22	17	7	5	2	0	2	9	18	23
7	20	15	5	3	4	2	0	7	16	21
8	15	10	2	6	11	9	7	0	9	14
9	5	2	12	16	20	18	16	9	0	5
10	2	7	17	23	25	23	21	14	5	0

A modified algorithm searching for minimum-cost paths with uses lookup tables assigned to graph nodes is presented below.

```
Modified Dijkstra algorithm
Input: g – graph with costs set for all edges, LUT_v - lookup tables for all nodes v
       s – source node
Output: prev – array with nodes on the shortest path
        shortestdist – array with shortest distances from the source to other nodes
for each v in g
  shortestdis[v] ← M AX
  prev[v] ← unknown
  if v ≠ s add v to Q
shortestdist[s] = LUT_s[1,1]  // s represents one of the entrances

for each l in Nb[s]  // Nb[s] - list of neighbours of s
  shortestdist[l] = shortestdist[s] + cost[s,l]
  prev[l] = s
while Q not empty
  k ← vertex from Q such that shortestdist[k] is the smallest for all k in Q
  remove k from Q
   for each l in Q and l in Nb[k] //  Nb[k] - list of neighbours of k still in Q
    xcost ←  shortestdist[k] + cost[k,l]+ LUT_k[i,j]  // where i is the entry point to the space
                                                         represented by node k from prev[k] and
                                                         j the leaving point from the space
                                                         represented by k to the space represented by l
    if xcost < shortestdist[l]
       shortestdist[l] ← xcost
       prev[l] ← k
    {end if}
```

For example searching for the shortest paths from *Entrance2* to other spaces we pass through *Hall4*. The cost of passing from *Entrance2* to *Hall4* is obtained by summing the value $LUT_{Entrance2}[1,1] = 5$ (as it is the cost of using the ramp leading to this entrance) and the cost of the edge connecting the nodes representing these spaces (which equals 0). The entry point to *Hall4* from *Entrance2* is denoted by 1. Thus, costs of reaching *Hall3*, *Hall2* and *Hall5* from *Hall4* are obtained using the values stored in the first row of the lookup table for *Hall4* (LUT_{Hall4}) depicted in Table 1, in the following way:

- $LUT_{Hall4}[1,5] + 0$ (where 0 is the cost of the edge connecting nodes representing *Hall4* and *Hall3*) - for the node labelled *Hall3*,

- $LUT_{Hall4}[1,4] + 2$ (where 2 is the cost of the edge connecting nodes representing *Hall4* and *Hall2*) - for the node labelled *Hall2*,
- $LUT_{Hall4}[1,2] + 0$ (where 0 is the cost of the edge connecting nodes representing *Hall4* and *Hall5*) - for the node labelled *Hall5*.

The lookup table representing costs of passing between different entry points of the *Hall4* is shown as Table 1. Lookup tables for *Stairs2* and *Entrance1* contain the value MAX, because it is impossible to pass the stairs between two halls and leading to the main entrance on a wheelchair. Thus the routes passing between *Hall5* and *Hall6* and entering the building through the main entrance have the maximal cost (are the longest ones).

Results and discussion

The proposed interactive application offers intelligent knowledge-based navigation, which is based on the graph representation of the building and its features. The approach described above was implemented and tested on example IFC files for the part of the office building shown in Fig. 3. The given IFC file is converted into an internal representation in the form of the graph. The application searches for best routes to a chosen destination room starting from all possible building entrances.

The simplified graph used by the proposed algorithm searching for accessible routes leading from *Entrance2* to *Office3* is shown in Fig. 6.

The results found by the algorithm for routes between *Entrance2* and *Office3* (in both directions) are depicted in the drawing of the ground floor layout presented in Fig. 7. It can be observed that the cost of opening heavy doors at both ends of *Hall2* causes the shortest path from the *Entrance2* to *Office3* pass through *Hall2*, but the return route passes through *Hall3*. It results from the fact that for the person on a wheelchair pushing the doors is easier then pulling them and the corresponding costs were set at 2 and 20, respectively. The second route is longer in terms of the distance but is less costly in terms of the effort required to operate the door.

The standard algorithms used to search for shortest paths in building layouts return results which are less costly, but the obtained routes are often not accessible by disabled or too difficult to pass. The shortest route found by the standard algorithm, i.e., without taking into account semantic information related to passing through doors and stairs, from the *Entrance2* to *Office3* leads through *Hall5*, *Stairs2* and *Hall6*. However this route is not accessible for people moving on wheelchairs as the stairs between the two halls are not equipped with a ramp.

As mentioned earlier, the designers are well aware of the need to provide accessible routes for disabled persons. At the same time there is no requirement to make these routes comfortable or easy. The applications used by designers do not provide any visual clue to how the most comfortable or easy access for a disabled person would actually look like. To help designers solve this problem we designed a prototype of a user interface that if incorporated into design software could increase the awareness of the problem among designers.

The prototype consists of several screens which allow the designer to see the layout of the building in a visual form and perform basic actions. The first element of the prototype allows the designer to visualize the layout of the analyzed building or part of the building. Fig. 8 depicts the screenshot of this element with the view of the layout used in this paper. Then the designer can enter, define or import from an external source the values for the lookup table associated with a space. In Fig. 9 a screenshot of the interface with the table containing the lookup table values for *Hall4* is depicted. The table can be edited directly by the designer or the data can be obtained from other sources. Figure 10 shows the screen allowing the designer to visually select the space for which the accessibility is to be visualized. The last element of the user interface allows the user to see the

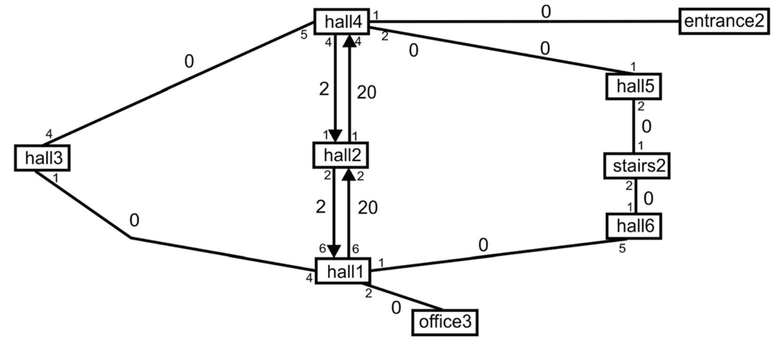


Fig. 6 The simplified graph used to search for accessible routes leading from *Entrance2* to *Office3*

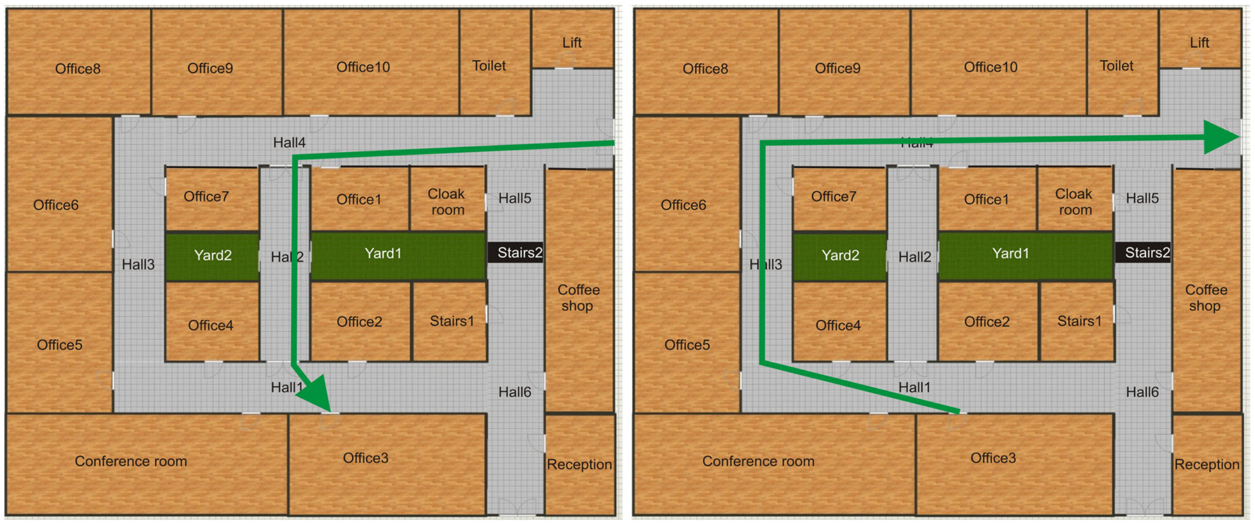


Fig. 7 Routes found by the algorithm as the optimal access between *Entrance2* and *Office3*

actual route a disabled person would have to take within the building. Depending on the selection made by the designer either the route from the entrance to the selected room or the route from this room to the exit is visualized. As depicted in Fig. 11 the route to and from the selected room can significantly differ.

The prototype of the interface has been implemented using the Axure prototyping tool which allows to generate an interactive and responsive interface which can be run as the web application within the application of choice. For this paper the prototype has been run within the Firefox application to test it.

The proposed method allows us to include semantic knowledge about IFC entities in the graph representation of floor layouts. The information about existence of stair ramps, existence and style of doors between spaces, and the direction of their opening is very important in the process of searching for routes both accessible and easy to be traversed. The information about the distances which should be covered between different entry/leaving points in each space is also used. The described application not only helps the designer to include special requirements during the design process but also offers the knowledge-based intelligent navigation which proposes the best

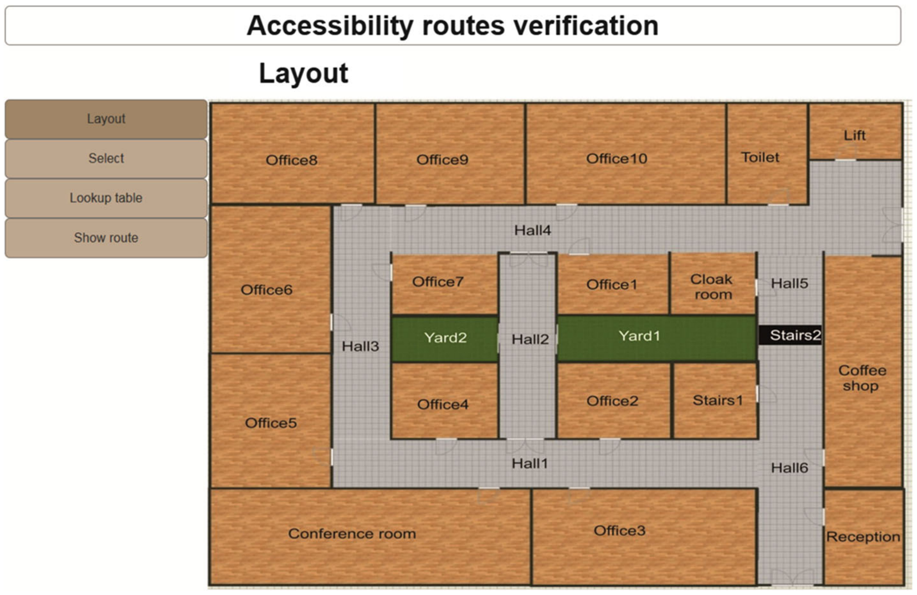
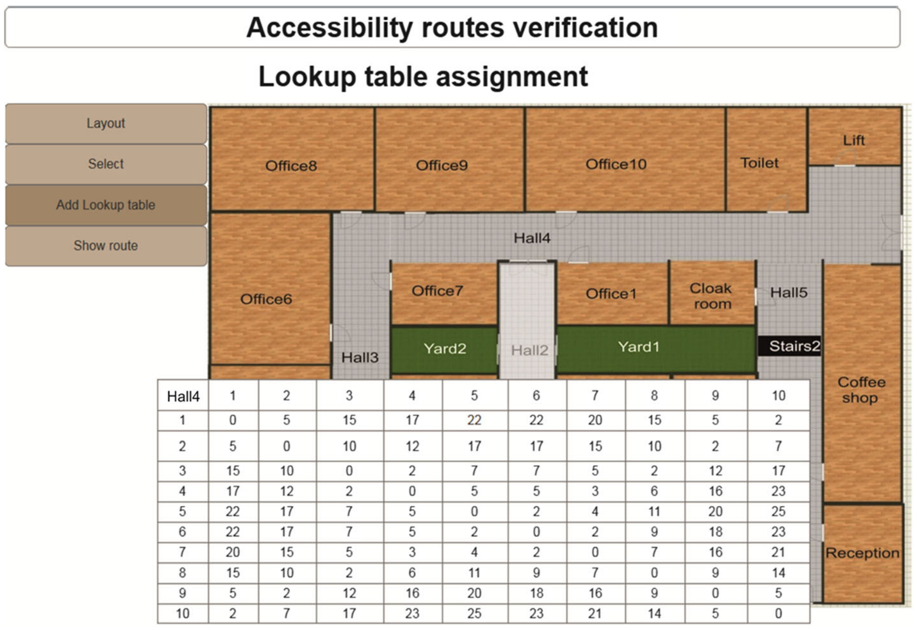


Fig. 8 A screenshot of the *Layout* view of the prototype interface



Hall4	1	2	3	4	5	6	7	8	9	10
1	0	5	15	17	22	22	20	15	5	2
2	5	0	10	12	17	17	15	10	2	7
3	15	10	0	2	7	7	5	2	12	17
4	17	12	2	0	5	5	3	6	16	23
5	22	17	7	5	0	2	4	11	20	25
6	22	17	7	5	2	0	2	9	18	23
7	20	15	5	3	4	2	0	7	16	21
8	15	10	2	6	11	9	7	0	9	14
9	5	2	12	16	20	18	16	9	0	5
10	2	7	17	23	25	23	21	14	5	0

Fig. 9 A screenshot of the *Lookup table* view of the prototype interface, which allows the user to enter the cost values or import the data from an external source

routes connecting selected places and comfortable for disabled persons.

Figure 12 presents the use case diagram which explains how the tool can assist disabled people. In the first use case the system shows the layout of the selected building floor. In the second case the user specifies the entrance point and the target space and the system draws the route easiest to traverse on the floor layout. In the third case the system shows instructions on how to use the application.

Conclusions

This paper describes our prototypical application which aids disabled people to find accessible routes to selected rooms using mobile devices. The information about the

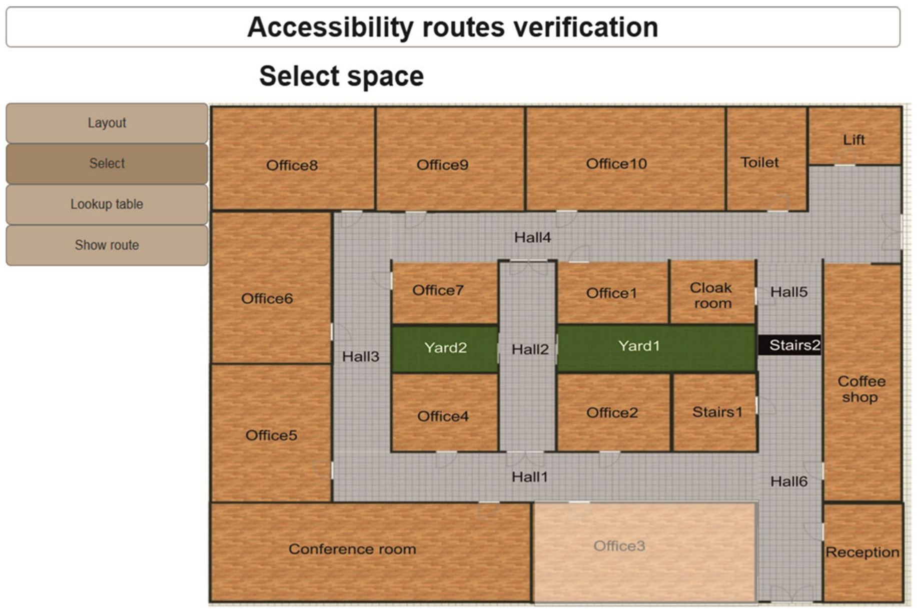


Fig. 10 A screenshot of the *Select* view of the prototype interface

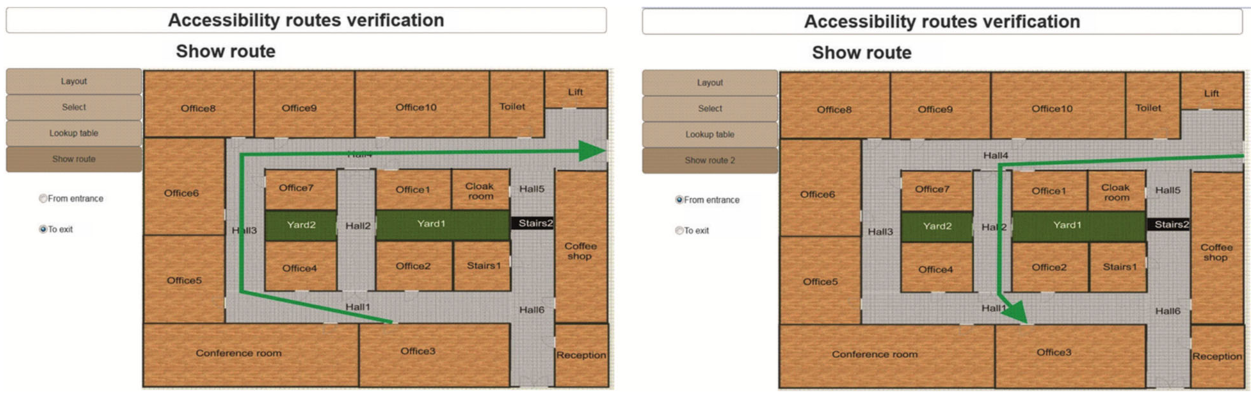


Fig. 11 A screenshots of the *Show route* view of the prototype interface, which depicts the accessible route within the building from or to the exit, depending on the designers choice

building layout is extracted from the IFC file and stored in the graph representation together with information about the accessibility between spaces. As for people moving around on wheelchairs easiness of passing a route between two places can depend on the route direction, the directed graphs are used to represent layouts.

The paper contributes to the field of computational methods applied to searching for routes inside buildings. It proposes a methodology based on extracting topological and semantic information from IFC files describing building layouts. Building-specific knowledge is used to search for accessible routs by means of the modified one-source minimal-cost search algorithm, where costs of passing through different spaces and between them is considered. In our method we assume that the input is in the form of the IFC compliant STEP (Standard for Exchange of Product Model Data) data file (ISO 10303–21, 2002). Although BIM tools produce data in various formats, an open standard IFC data model developed by the International Alliance for Interoperability allows the building information to be exported from these tools to the standard IFC compliant format (Dhillon et al., 2014). Many systems can implement converters between the IFC schema and their native data models to export and import IFC instances to exchange information with each other (Zhang et al. 2015).

Moreover, in our approach the information about style and types of doors (left, right) and the existence of stair ramps is indispensable. It is assumed that during the design phase the designer has properly modelled such information into building models before exporting them to the IFC format.

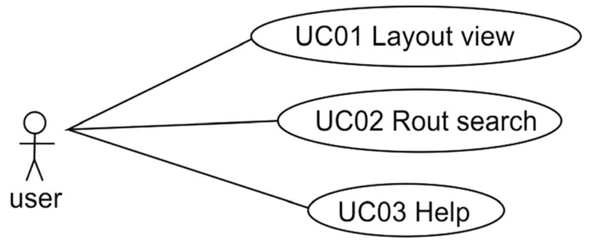


Fig. 12 The use case diagram

One of the practical challenges of the proposed method is determining the coefficients which are assigned as cost of opening doors in the lookup tables. Up to now we assume that this cost corresponds to the distance which would be covered if the doors were opened automatically in the time interval needed to open them manually. Moreover the application should adapt itself to dynamical changes in the building. For example it should take into account the situation when one of the doors are temporarily out of order.

In future we intend to give the possibility of adding more elements which can be treated as different obstacles in moving around the building. The graph representation of information extracted from IFC files will be also used to search for accessible egress routes in case of emergency.

Funding
There is no funding for the research reported that should be declared.

Authors' contributions
Both authors contributed extensively to the work presented in this paper. GŚ worked on creating a graph model of the building on the basis of information about the building layout retrieved from IFC files. Both authors reviewed and analyzed the literature, and developed the algorithm to compute optimal routes accessible for disabled people. Strug designed a user interface prototype for a computer tool allowing for visualization search results on the building floor layouts. Both authors approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 7 November 2016 Accepted: 1 June 2017
Published online: 15 June 2017

References

AlShboul, A. A., & Al-Tahat, M. D. (2007). Modelling of public building evacuation processes. *Architectural Science Review, 50*, 37–43.

Andrade, I., Dorneles, V., & Bins Ely, V. H. M. (2012). Accessibility for All: Going from theory to Practice. *Work, 41*, 3840–3846.

Bright, K., & Di Giulio, R. (2002). *Inclusive Buildings: Designing and Managing an Accessible Environment*. Great Britain: Blackwell Science.

BuildingSMART. (2013a). *IFC*. Retrieved from buildingSMART website: http://buildingsmart.org/standards/ifc. Accessed 24 Sept 2015.

BuildingSMART. (2013b). *IFC2x3*. Retrieved from buildingSMART website: http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/index.htm. Accessed 24 Sept 2015.

Cepolina, E. M. (2005). A methodology for defining building evacuation routes. *Civil Engineering and Environmental Systems, 22*, 29–47.

Dhillon, R., Jethwa, M., & Rai, H. (2014). Extracting building data from BIM with IFC. *Int. J. on Recent Trends in Engineering and Technology, 11*, 202–210.

Eastman, C., Teicholz, P., Sacks, R., and Liston, K. (2011). *BIM Handbook: A Guide to* Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors (2nd ed.). Wiley, Hoboken, New Jersey.

Eastman, C. (2012). The Evolution of AEC Interoperability, Proceedings of the EG-ICE, Herrsching, Germany.

Hu, Z., & Zhang, J. (2011). BIM- and 4D-based integrated solution of analysis and management for conflicts and structural safety problems during construction:2 Development and site trials. *Automation in Construction, 20*, 167–180.

ISO 10303–21. (2002). Industrial Automation System – Exchange of Product Model Data – Part 21:Implementation Methods; Clear Text Encoding of the Exchange Structure, International Organization for Standardization.

Iwarsson, S., & Stahl, A. (2003). Accessibility, usability and universal design-positioning and definition of concepts describing person-environment relationships. *Disability and Rehabilitation, 25*, 57–66.

Jeong, S., & Ban, Y. (2011). Computational algorithms to evaluate design solutions using Space Syntax. *Computer-Aided Design, 43*, 664–676.

Langenhan, C., Weber, M., Liwicki, M., Petzold, F., & Dengel, A. (2013). Graph-based retrieval of building information models for supporting the early design stages. *Advanced Engineering Informatics, 27*, 413–426.

Ma, Z., Wei, Z., Song, W., & Lou, Z. (2011). Application and extension of the IFC standard in construction cost estimating for tendering in China. *Automation in Construction, 20*, 196–204.

Open IFC Repository. (2013). Retrieved from the website: http://openifcmodel.cs.auckland.ac.nz, The University of Auckland, Accessed 2 April 2014.

Ozer, D., & Sener, S. (2013). User accessibility optimization using genetic algorithm: aDA, *A|Z ITU. Journal of the Faculty of Architecture, 10*, 212–230.

Papinigis, V., Geda, E., & Lukošius, K. (2010). Design of people evacuation from rooms and buildings. *Journal of Civil Engineering and Management, 16*, 131–139.

Pu, S., & Zlatanova, S. (2005). *Evacuation route calculation of inner buildings* (Geo-information for disaster management, pp. 1143–1161). Heidelberg: Springer Verlag.

Ronchi, E., and Nilsson, D. (2013). Fire evacuation in high-rise buildings: a review of human behaviour and modelling research, Fire Science Reviews, 2, 1–21.

Rüppel, U., Abolghasemzadeh, P., & Stübbe, K. (2010). *BIM-based Immersive Indoor Graph Networks for Emergency Situations in Buildings* (pp. 65–71). Nottingham: Nottingham University Press.

Sakkas, N., & Perez, J. (2006). Elaborating metrics for the accessibility of buildings. *Computers, Environment and Urban Systems, 30*, 661–685.

Trinius, W., & Sjöström, C. (2005). Service life planning and performance requirements. *Building Research and Information, 33*, 173–181.

Vreeker, R., Deakin, M., Huovilla, P., & Sunikka, S. (2002). The assessment of urban sustainable development. *Building Research and Information, 30*, 95–108.

Yatim, Y.M. (2012). Optimum escape routes designs and specification for high-rise buildings, Proceedings of the 3rd International Conference in Construction Industry, Indonesia.

Zarrinmehr, S., Asl., M., Yan., W., and Clayton., M. (2013). Optimizing Building Layout to Minimize the Level of Danger in Panic Evacuation Using Genetic Algorithm and Agent-based Crowd Simulation, Proceedings of the EG-ICE, Wien, Austria.

Zhang, C., Beetz, J., and Vries, B. (2013). Model View Definition on Semantic Level: A State of the Art Review, Proceedings of the EG-ICE, Wien.

Zhang, C., Beetz, J., & Weise, M. (2015). Interoperable validation for IFC building models using open standards. *Journal of Information Technology in Construction (ITcon), 2014*(20), 24–39. Special Issue: ECPPM.

Zhang, L., and Issa, R. (2011). Development of IFC-based Construction Industry Ontology for Information Retrieval from IFC Models, Proceedings of the EG-ICE, Enschede, Netherlands.