Communication papers of the Federated Conference on Computer Science and Information Systems, pp. 17–25 DOI: 10.15439/2017F262 ISSN 2300-5963 ACSIS, Vol. 13

Decision Making in Building Maintenance Using a Graph-Based Knowledge Representation

Wojciech Palacz, Grażyna Ślusarczyk, Andrzej Łachwa, Barbara Strug, Anna Paszyńska, Ewa Grabska
The Faculty of Physics, Astronomy and Applied Computer Science
Jagiellonian University
ul. Łojasiewicza 11, 30-348 Kraków, Poland
Email: {wojciech.palacz, grazyna.slusarczyk}@uj.edu.pl

***Abstract*—This paper is an attempt to support effective decision making in building management by assisting maintenance processes. Knowledge about buildings is stored in a graph with many hierarchies. This representation allows us to express different types of hierarchical dependencies between building parts, like geometrical and functional ones, in one structure. Moreover, such a structure is useful to extract subgraphs containing information necessary for a given computational task, such as locating a desired place and the shortest path leading to it. As maintanance processes often require dynamic path target selection, modified indoor navigation methods are proposed. The paper presents the capability of the described knowledge model to cope with complex queries referring to different types of information. The considered examples show that the proposed approach can be used for various facility maintenance management applications.**

I. INTRODUCTION

A COMPUTER-AIDED facility management system requires extensive information of multiple dimensions as it deals with problems of inspecting, repairing and substituting equipment in a cost-effective way [1]. As facility maintenance management directly affects the satisfaction of building users it should be able to identify problems and maintain the facilities in a time, cost and resource effective manner [2]. Simulation of equipment running conditions in a design phase of a building model can reveal the need of changes in facilities arrangement or even of the redesign of the equipment location spaces.

Good decisions for building maintenance require integration of various types of information and knowledge. Systems providing such an integration are actively researched [3]. Information about the geometrical and physical properties of a building is the most basic one. On top of this data, additional types of information can be added: which doors can be freely opened and which require a keycard, how rooms are numbered, which are used as office rooms and which as storage rooms, which employees are assigned to which rooms, etc. Some types of information may concern not the physical building, but be related to the building users, for example an organisational chart of the company which occupies the building, working hours for company departments, etc.

Having more types of information increases the breadth of knowledge available to a facility management system. This, in turn, makes the system more likely to be able to answer unusual and complex questions, which can enable innovative ways of managing the building. One of the problems of facility maintenance in modern buildings is finding information about equipment which can be turned off for some time to lower energy consumption. For example, in order to reduce the costs of building maintenance the current requirements for air conditioning of different building spaces should be considered.

Finding the useful information in a facility management system can greatly reduce costs of building maintenance. The effective maintenance of building facilities often needs navigation tools to locate a target place, like a storeroom of spare parts or a broken appliance, and the shortest path leading to it. It is also important to navigate the maintenance staff to equipment which should be operated manually by the shortest paths.

Navigation systems are, in general, concerned with determining the current location of their user, finding the optimal path to a target point, and then guiding the user along this path. Outdoor devices usually use satellite-based location services, but these services are unreliable inside buildings. Indoor systems use alternative approaches based on Wi-Fi [4], UWB, RFID [5], etc. Approaches based on matching images from smartphone cameras to the database of unique landmarks were also proposed [6]. In the last resort, the user can simply be asked to mark their current location on a map.

The optimal path is usually defined as the shortest one, and calculated by an appropriate graph algorithm. This implies that the system must have an alternative map in the form of a graph. This graph will usually have nodes representing doors and other important points in the building, and edges representing possible movements of a person walking from one point to another one [7].

The guidance part can be limited to showing a map with a drawn path, or can be provided as a list of commands (“go straight 100 meters, then turn right and walk another 50 meters”). Humans do not have an innate understanding of numeric distances and angles, therefore instructions using landmarks (“walk up the stairs, then turn towards the coffee machine”) are easier to follow [6]. In the extreme case, when the navigation device does not use any location service and cannot update its own position in real-time, the guidance instructions may simply be a sequence of photos showing landmarks the user should walk by on their way from the starting to the target point [8].

Navigation strategy proposed in this paper is based on knowledge about the building layout stored in the hierarchical graph structure. Its multiple hierarchies allow for building models which represent not only the spatial hierarchy of rooms and floors, but also other functional and administrative dependencies. This additional knowledge stored in the graph can be used in the target selection phase.

Currently available navigation tools require users to specify a single point as a target, either by directly marking it on a map or by providing a description which can be translated by the device to point coordinates (“the room with the number 303”, “the coffee machine closest to my current position”). The system based on the graph model proposed in this paper will be able to find targets described in a parametric way or specified by selected attributes (for example, the type of a scientific equipment shop). The hierarchy grouping graph nodes into larger entities makes it possible to accept requests for “the closest laboratory belonging to the Department of Organic Chemistry”. What is even more important, specifying targets which resolve to a set of points will also be possible: “all office rooms belonging to the Department of Games” will make the navigation device guide its user from one room to the other.

The hierarchical nature of the graph model can also be useful for path searching algorithms. It provides the possibility of filtering the graph by hierarchy type and level, eliminating knowledge not related to the task of finding a path between two points in the chosen fragment of the building. The number of data to be analyzed is greatly reduced, thus speeding up the search.

II. RELATED WORKS

Building maintenance can be seen as the combination of administrative and technical actions to ensure the items of the building in an acceptable standard to perform its required functions [9]. The main objectives of building maintenance are: extending the useful life of the buildings, providing save and efficient working or living environment and maximization of economic and aesthetic values of the buildings. Building maintenance can be seen as an activity in facilities management [10] and considered as the part of the construction sector [11].

Building management requires taking many important decisions regarding the way of inspecting, maintaining, renewing or replacing existing facilities [1]. To support decision making at different stages of a building lifecycle the constant acquisition and analysis of data is needed [12]. Limited access to accureate data, knowledge and integrated of multiple domains makes it difficult to create user-friendly interfaces for decision support in operation and maintanance of building facilities [13].

Good decisions for building maintenance require integration of various types of information and knowledge. Current BIM applications are not able to fully exploit the capabilities of the knowledge systems. In [14] and [15] Radio Frequency Identification (RFID) is proposed to collect and share maintenance data with the most minimal manual data entry. The method is used in an integrated system proposed in [3] which captures both, information and knowledge. The system developed in [3] concerns rather capturing information and knowledge, while the efficiency of the system is improved by the automation of data capturing by means of technologies such as RFID. The system helps maintenance teams learn from previous experience and trace the history of a building elements on the base of Case-Based Reasoning system (CBR).

Majority of the current building maintenance systems mainly focus on capturing either information or knowledge. On the other hand many currently available building maintenance system capturing information and knowledge are concentrated on tracing the history of the building and learning from previous experience.

In our approach, we focus on another aspect of the effective maintenance of building facilities—navigation to locate a target place, like a storeroom of spare parts or a broken appliance, and the shortest path leading to it as well as finding information about equipment which can be turned off for some time to lower energy consumption. To allow for effective navigation and finding information about equipment, the hierarchical graphs with multiple hierarchies of various types are used as a representation of building models.

Graph structures are commonly used to model relations in many systems [16], [17]. They can be used in design systems to model topological relations between components, in databases to model logical relations between objects as well as to model data and control flow or specification and analysis of software systems. In many real applications, modelling systems by plain graphs (graphs where edges and nodes are sets without additional structure) is insufficient. The complex systems are very often hierarchical, thus graphs with hierarchy should be used as the models [18]–[20]. Graphs of this kind allow not only for modelling hierarchical relations between components of the system but also for hiding some details of “encapsulated” subgraphs. Hierarchical graphs used in conjunction with graph transformation rules can be used to manage changes in the state of big and complex systems like computing grids [21].

During the last years several models based on graphs with hierarchy have been defined [22]–[33]. They were used for modelling global computing systems, syntactical structure of agents, mobile systems, design systems, granular computing, etc. It turns out that existing hierarchical graphs are not sufficient to efficiently model objects considered in architectural design and engineering problems. The good representation should allow for expressing multi-argument relations between parts of different components, hierarchical dependencies between different parts of design, as well as many hierarchies of different types, where not all components must belong to the hierarchy. None of the mentioned models allows for modelling of all required features.

Therefore in this paper hierarchical graphs with multiple hierarchies of various types are proposed as representation of building models. The considered graphs are composed of nodes, which represent object components, and edges representing relations between them. In given graph nodes other nodes and edges can be nested. Graph nodes with nested atoms are called hierarchical. Hierarchical nodes may be used to hide certain details of a designed object that are not needed in a given moment or to group objects having common features (geometrical or functional). Graph hierarchies enable us to group elements according to different objectives and consider the project at a chosen level of detail. Hierarchical graphs encode the knowledge about created designs and therefore enable to efficiently reason about the conformity of designs with specified design criteria [34].

In this paper, a multi-hierarchical graph model is used to search for possible routes in a given building. The most common graph algorithm for route calculation is the Dijkstra's algorithm. Other graph algorithms like A*, Floyd-Warshall or Bellman-Ford [35] are also used to determine the shortest path from a source to a destination. In [36] a reduced visibility graph is used to reflect possible route choices and the Fastest-Path Algorithm finds the fastest path for travelers [37]. In [38] metrical-topological models constructed from labelled floor plan geometry are used to describe the shape and connectivity of space, and constitute useful underlying tools for wayfinding. However, for almost all current navigation applications complex building interiors with large open spaces and vertical communication means present challenges.

III. THE GRAPH-BASED REPRESENTATION OF BUILDING-RELATED KNOWLEDGE

In this section the extended graph-based data structures, which better reflect the intrinsic nature of design objects, are described. They enable to express geometrical properties of objects together with their other attributes and different types of relations between object components.

Flat graphs are often used to model internal structures of objects. Nodes represent components of an object, and relations between them are represented by edges. These simple models are no longer convenient when components themselves need to be decomposed into sub-components without losing their identity. In such a case, the model has to contain nodes representing both the component and its parts.

Hierarchical graphs can represent the connection between the whole and its parts in a straightforward way, because nodes and edges can be nested inside other nodes. A graph node with nested atoms is called hierarchical and represents a component as a whole, while the children atoms represent this component's internal structure. Graphs considered in this paper also allow for expressing relationships between components located on different hierarchy levels, i.e., edges can connect nodes nested in different parents.

The considered graphs are multi-hierarchical, i.e., there is a possibility to define hierarchies of different types in one graph. Many hierarchies defined in one graph allow us to specify different types of hierarchical dependencies between graph elements. Such a representation makes it possible to filter the knowledge stored in the graph by selecting specific hierarchies. Moreover, there is a choice of different views of the project consisting of selected hierarchies including only relevant subsets of graph atoms.

The proposed multi-hierarchical graphs can represent a complex object at different levels of detail and different stages of the building lifecycle. They allow for hiding unnecessary at the moment low-level data and presenting only an outline of the object or showing a detailed view of the whole object (or of its part). The considered multi-hierarchical graphs are an extension of labelled and attributed graphs. Nodes and edges are labelled by names of components and relations, respectively. Semantic aspects of components and relations are encoded in sets of attributes assigned to nodes and edges.

Use of multiple graph hierarchies, instead of a single hierarchy, allows for additional knowledge to be encoded in the graph structure. Each hierarchy can correspond to different element grouping criteria. Reasoning about designs [34] can be more efficient thanks to the possibility of selecting both the hierarchy type, level, and the appropriate components of the object.

Let us introduce a formal description of the proposed multi-hierarchical graph model.

Let Σ_V and Σ_E be two disjoint alphabets of node and edge labels, respectively. Let A be a nonempty, finite set of node and edge attributes. Let D_a denote a fixed, nonempty set of admissible values for attribute $a \in A$, called the domain of a.

Let the set of nodes and edges of a graph G, known together as graph atoms, be denoted by U, i.e., $U = V \cup E$. It can be assumed without loss of generality that Σ_V, Σ_E, A, V and E are pairwise disjoint.

Some graph atoms may be completely outside the hierarchy, in the sense that concepts like "set of children" and "parent" are not applicable to them at all. Let $\perp$ be a fixed value, different from all graph atoms, which will be used to denote this case in the following.

Definition 1: A mapping $ch : V \to 2^U \cup \{\perp\}$ is a **child nesting function** if and only if:

- $\forall v, w \in V : w \in ch(v) \Rightarrow ch(w) \neq \perp$, i.e. if a node is a child of some other node, then it is a part of the hierarchy and thus must have its own set of children (which may be empty),
- $\forall x \in U\ \forall v, u \in V : x \in ch(v) \wedge x \in ch(u) \Rightarrow v = u$, i.e., an element cannot have two distinct parents,
- $\forall v \in V : v \notin ch^+(v)$, i.e., no element can be its own descendant.

For node $v \in V$, $ch(v)$ is known as a set of its children, $ch^+(v)$ as a set of its descendants, and $ch^*(v)$ denotes a set containing both node v and its descendants.

Definition 2: A **multi-hierarchical graph** over Σ_V, Σ_E and A with n hierarchies (where n is a fixed natural number) is a system $G = (V, E, s, t, ch_1, \ldots, ch_n, vlb, elb, atr, val)$, where:

- V and E are disjoint finite sets, with V being non-empty, whose elements are called nodes and edges, respectively,
- $s : E \to V$ and $t : E \to V$ are mappings assigning to edges their source and target nodes, respectively,

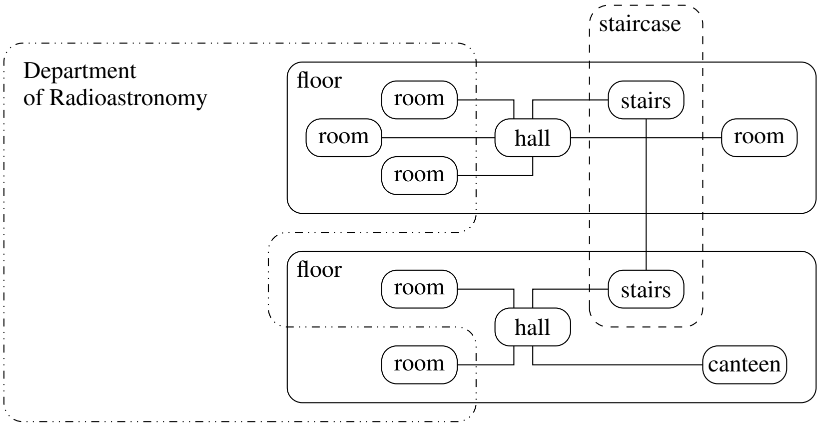

Fig. 1. An example graph model with three hierarchies

- $ch_i : V \to 2^U \cup \{\perp\}$, where $i = 1, \dots, n$ are child nesting functions,
- $vlb : V \to \Sigma_V$ and $elb : E \to \Sigma_E$ are node and edge labelling functions, respectively,
- $atr : U \to 2^A$ is a node and edge attributing function,
- $val : U \times A \to D$, with $D = \bigcup_{a \in A} D_a$, is a partial function such that for all $x \in U$ and $a \in A$ if $a \in atr(x)$, then $val(x, a) \in D_a$.

In Fig. 1 an example of a graph with three different hierarchies is presented. In the first hierarchy, nodes representing rooms and other spaces are nested in hierarchical nodes representing floors. Edges which connect nodes represent accessibility. All graph elements belonging to this hierarchy are drawn as solid ovals and lines.

The second hierarchy is used to group stair segments into staircases. While stair runs and their landings are contained in their specific floors, staircases considered as a whole do not belong to any floor. Therefore, the node in Fig. 1 which represents a staircase is not a part of the first hierarchy, and is drawn with a dashed line to underline this fact.

The third hierarchy stores knowledge about administrative room assignments. In this example there are four rooms assigned to the Department of Radioastronomy. Nodes representing these rooms are nested in the node representing the department.

It should be noted that a graph atom can belong to several hierarchies at the same time. For example, nodes representing stair segments are a part of the spatial and of the vertical spatial hierarchy; they are not a part of the administrative one.

Notions of a subgraph of a multi-hierarchical graph, an induced subgraph and a single view are useful tools for filtering the knowledge stored in a graph model.

Parts of an object represented by a multi-hierarchical graph correspond to its subgraphs.

Definition 3: A graph $G = (V_G, E_G, s_G, t_G, ch_{1\,G}, \dots, ch_{n\,G}, vlb_G, elb_G, atr_G, val_G)$ is a **subgraph** of a graph $H = (V_H, E_H, s_H, t_H, ch_{1\,H}, \dots, ch_{n\,H}, vlb_H, elb_H, atr_H, val_H)$ if:

- $V_G \subseteq V_H$;
- $E_G \subseteq E_H$ and $\forall e \in E_G : s_H(e) \in V_G \land t_H(e) \in V_G$;
- $s_G = s_H|_{E_G}, t_G = t_H|_{E_G}$;
- $\forall x \in V_G : ch_{i\,G}(x) = ch_{i\,H}(x) \cap (U_G \cup \{\perp\})$ where $i = 1, \dots, n$,
- $vlb_G = vlb_H|_{V_G}$, $elb_G = elb_H|_{E_G}$;
- $\forall x \in U_G : atr_G(x) \subseteq atr_H(x)$,
- $\forall x \in U_G, a \in A : val_G(x, a) = val_H(x, a)$.

A subgraph in which all edges connecting selected nodes of a given graph are present is called an induced subgraph.

Definition 4: A graph $\boldsymbol{Ind(V_I)} = (V_I, E_I, s_I, t_I, ch_{1\,I}, \dots, ch_{n\,I}, vlb_I, elb_I, atr_I, val_I)$ is an **induced subgraph** of an graph $H = (V_H, E_H, s_H, t_H, ch_{1\,H}, \dots, ch_{n\,H}, atr_H, val_H)$ if $Ind(V_I)$ is a subgraph of H, where $E_I = \{e \in E_H : s_H(e) \in V_I \land t_H(e) \in V_I\}$.

Searching for specific information about the design, the designer often needs to take into consideration only one graph hierarchy. A subgraph which contains all nodes of the selected hierarchy and edges representing relations between them is called a single view.

Definition 5: Let $H = (V_H, E_H, s_H, t_H, ch_{1\,H}, \dots, ch_{n\,H}, vlb_H, elb_H, atr_H, val_H)$ be a graph and let i be a fixed number $1 \leq i \leq n$. A graph $\boldsymbol{S(ch_i)} = (V_S, E_S, s_S, t_S, ch_{i\,S}, vlb_S, elb_S, atr_S, val_S)$ is a **single view** determined by the hierarchy $ch_{i\,H}$ if $S(ch_i)$ is a fragment of an induced subgraph of H where $V_S = ch^*_{i\,H}(V_H) \cap V_H$ and ch_j for $j \neq i$ are omitted.

IV. CASE STUDY

A university building is used as an example in this section. Fig. 2 displays a part of its floor plan, which includes several rooms belonging to two departments, a student canteen and a staircase. A multi-hierarchical graph which models this building has four hierarchies: spatial, vertical spatial, functional and administrative. A graph with many hierarchies is difficult to arrange on a diagram in a clear way, therefore the following figures show different views of this graph.

The first hierarchy, which groups together nodes representing rooms belonging to the same floor and places them inside a node representing this floor, is the most obvious one. It also

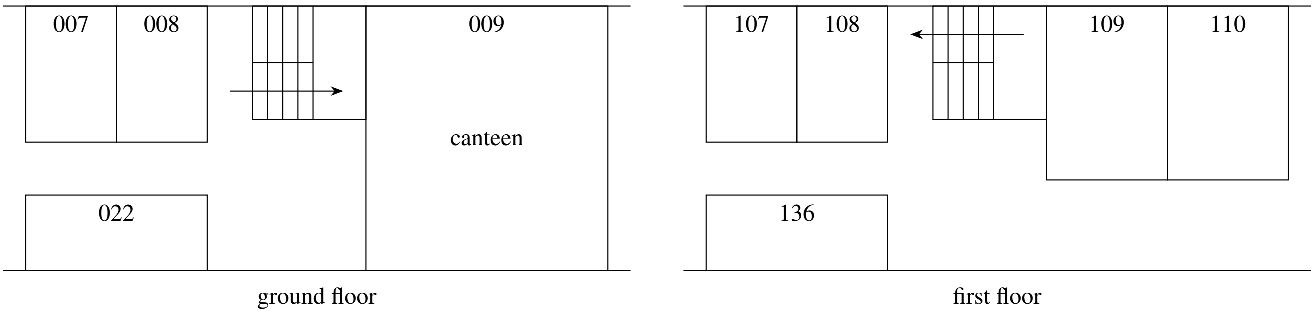

Fig. 2. A fragment of a university building floor plan

provides nodes representing entry/exit points which are used to move between adjacent spaces in the building. Most of them represent doors, some represent stair landings. Edges connecting these points denote accessibility and are attributed with numbers representing costs of traversing the edges (i.e., the time required to walk from one point to the other). A single view showing this spatial hierarchy is displayed in Fig. 3.

The second hierarchy is also based on physical containment, but it groups spaces in the vertical direction. It is named “vertical spatial” and provides hierarchical nodes representing staircases and lifts—that is, objects which cross floor boundaries.

The third hierarchy (see Fig. 4) divides the building into functional areas. All corridors and stairs are assigned as children of the “communication” node, personal rooms of university employees as children of the “staff offices” node, canteens and other places where food can be bought are assigned to the “gastronomy” node, and so on.

The fourth hierarchy (see Fig. 5) provides nodes corresponding to the administrative structure of a university (chairs, departments, etc.) and places rooms inside these nodes.

Usually, floor layouts like the one displayed in Fig. 2 show only the geometry of the building and room labels. A multi-hierarchical graph model is able to store additional information. This extra knowledge can be used to support specific facility management tasks.

A. Case 1: HVAC system

Modern buildings usually have HVAC (heating, ventilation, and air conditioning) systems which are centrally managed by a BMS (building management system). To reduce the energy consumption, the BMS often switches off the HVAC systems at the end of a day and turns them back on the next morning.

In a university building there are additional chances for saving energy, offered by irregular events. A special event can be accompanied by a university-wide cancellation of classes, which means that nobody will use lecture halls and student laboratories. On a smaller scale, a conference attended by a department can mean that the staff from that department will be absent for several days, and AC in their office rooms can be turned off for that period.

To take advantage of such events, a manual HVAC schedule adjustment must be made at a BMS computer terminal. Making adjustments room by room is time-consuming and error-prone. The BMS can make this procedure easier by letting the operator specify a group of rooms and then apply the adjustment to the whole group in a single step. The procedure gets even easier if the group can be automatically defined.

Integrating the BMS and the multi-hierarchical graph model makes it possible. The set of graph nodes representing office rooms belonging to a considered department is calculated on the basis of knowledge stored in the administrative (see Fig. 5) and functional (see Fig. 4) hierarchies. Alg. 1 shows the algorithm which implements this calculation. For example, when given the name “Department of Games” as an argument, function *OfficeRooms* returns a set of nodes representing rooms number 007 and 008. Returned graph nodes are mapped to objects representing these rooms in the BMS.

B. Case 2: navigation support

Let us consider a case of guiding a maintenance technician to a correct shop. The technician was called in to check a non-working piece of scientific equipment, and he determined that a repair is possible but requires a spare part and/or use of specific tools. The required items can be obtained from workers in an equipment shop. The problem is, the building contains several such shops, and going to a wrong one is a waste of time. The shops are assigned to departments, and

Algorithm 1 Outline of an algorithm used by the BMS

```
Room selection algorithm.
Input: a name of a department or other administrative unit.
Output: a set of nodes representing office rooms assigned to
this department/unit.

constants
    offices := the node labelled “staff offices”

function OfficeRooms (administrativeUnitName)
    $v$ := the node labelled administrativeUnitName
    $A$ := $ch^{+}_{administrative}(v)$
    $B$ := $ch_{functional}(offices)$
    return $A \cap B$
endfunc
```

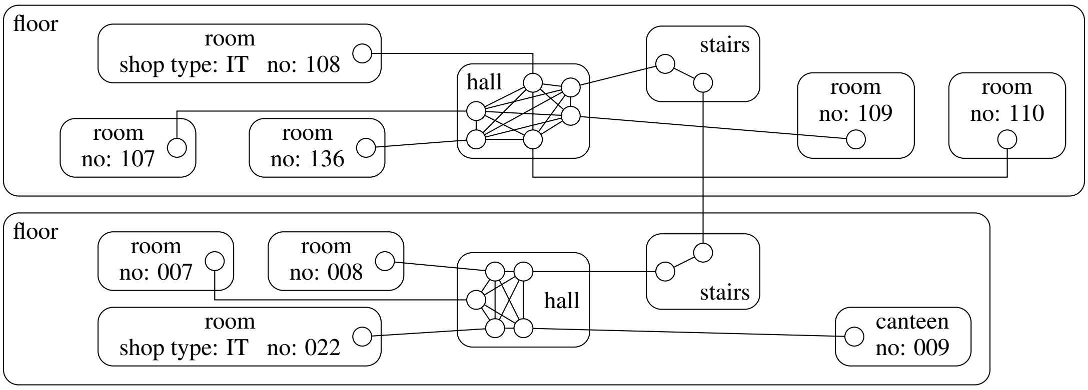

Fig. 3. A view showing the spatial hierarchy of the building model (edge costs omitted)

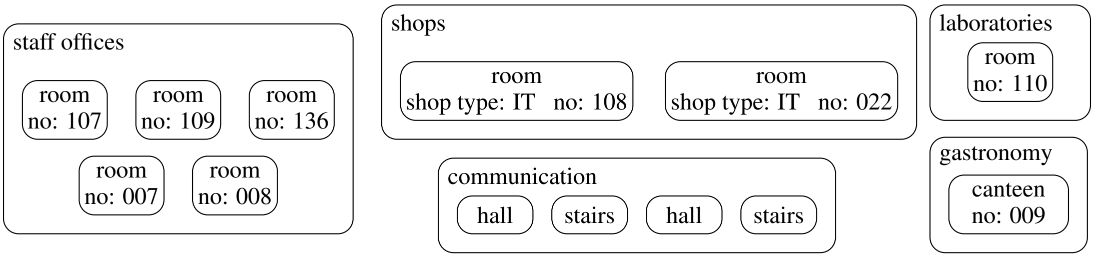

Fig. 4. A view showing the functional hierarchy of the building model

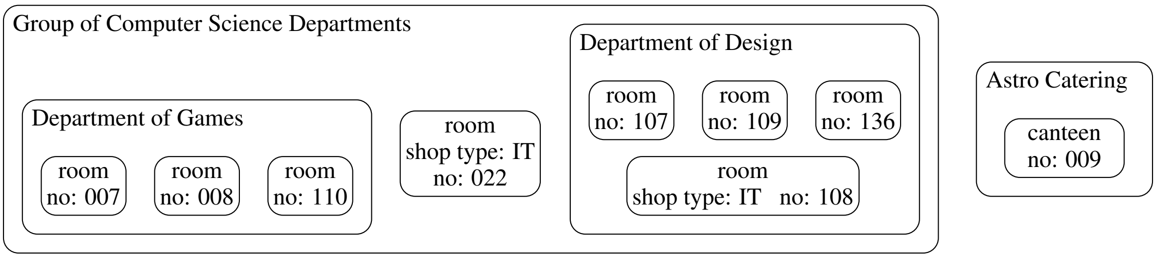

Fig. 5. A view showing the administrative hierarchy of the building model

because of university accounting rules, shop workers may not expend resources to fix equipment belonging to an unrelated department.

A navigation application with a dynamic target selection algorithm can help the technician to get to the correct shop. Pseudocode in Alg. 2 shows how to implement this guidance algorithm. We assume that the technician starts in the room with a broken piece of equipment. The *SearchForShop* procedure uses this assumption to determine which department this equipment belongs to. It can then take the set of nodes contained in the node representing this department (administrative hierarchy) and intersect it with the set of nodes nested in the “shops” node (functional hierarchy). This determines the room—or, if the department has more than one shop assigned, the set of rooms—to check in the first search phase.

Then, the application should direct the technician to the nearest room from this set. This is implemented in the *TryNodes* procedure. The code finds the shortest path from the node representing the room the technician is currently in to any of the nodes representing rooms to check (or rather, from a node representing an entry/exit point in the current room to any node representing a point in the destination rooms). To speed up the search the Dijkstra’s algorithm is applied not to the complete graph model, but to its subgraph. Specifically, to a spatial hierarchy subgraph induced by start and destination nodes as well as by nodes which in the functional hierarchy are nested in the “communication” node.

If, when arriving at the destination point, the technician finds the shop closed or the replacement part unavailable, then they command the navigation application to direct them to the next shop. The application removes the node representing the visited shop from the set of destination nodes and repeats the procedure described in the previous paragraph.

If required parts can not be obtained from any shop in the department, the application checks the administrative hierarchy to find the parent unit of the department, and considers shops assigned directly to that unit. If necessary, it then proceeds to the grandparent unit, the great-grandparent unit, etc.

Algorithm 2 Outline of an algorithm used by the navigation application

```
Dynamic target search algorithm.
SearchForShop determines which sets of rooms to visit in each search phase.
TryNodes is an auxiliary procedure which navigates the user between rooms.

constants
    shops := the node labelled "shops"
    comm := the node labelled "communication"
    $D$ := $S(ch_{spatial})$, i.e., the single view of the spatial hierarchy (see Fig. 3)

procedure SearchForShop
    $u$ := node representing the room user is currently in
    $v$ := $par_{administrative}(u)$
    while $v \neq$ NULL do
        $A$ := $ch_{administrative}(v)$
        $B$ := $ch_{functional}(shops)$
        TryNodes ($A \cap B$)
        $v$ := $par_{administrative}(v)$
    endwhile
    report failure and end program, all possibilities exhausted
endproc

procedure TryNodes (setOfRoomNodes)
    while setOfRoomNodes $\neq \emptyset$ do
        $p$ := node representing the point user is currently in
        $A$ := subgraph of $D$ induced by $ch^{*}_{spatial}(ch_{functional}(comm) \cup \{p\} \cup setOfRoomNodes)$
        use Dijkstra's algorithm on $A$ with $p$ as the source node and entry points of setOfRoomNodes as the target nodes
        let target be the closest target point
        targetRoom := $par_{spatial}(target)$

        display map with the route from $p$ to target
        direct user to target
        wait for user's decision

        if user wants to check another room then
            setOfRoomNodes := setOfRoomNodes $-$ {targetRoom}
        else
            report success and end program
        endif
    endwhile
endproc
```

Fig. 6 illustrates the result of this process: the shop assigned to the Department of Design is closed, and the navigation application displays directions to the shop belonging to the Group of CS Departments. This path was calculated by the Dijkstra's algorithm invoked on a graph displayed in Fig. 7.

It should be noted that while *SearchForShop* implements a very specialised algorithm specific to this case, *TryNodes* is a general navigational procedure, able to direct people to successive rooms from the set given as its invocation argument. It should be possible to reuse *TryNodes* without modification in many other algorithms solving different navigation problems faced by university visitors, students and staff.

V. DISCUSSION

As demonstrated by these case studies, the use of multiple hierarchies in a graph model allows that model to represent several different hierarchical relations in a natural way. Knowledge stored in these relations can be combined and used to extract subgraphs which contain information necessary for a given computational task. This reduction of data size speeds up processing.

Knowledge concerning properties of a single building component can be stored in attributes of the graph node corresponding to that component. In the case studies, types of shops are represented in this way. This additional information could

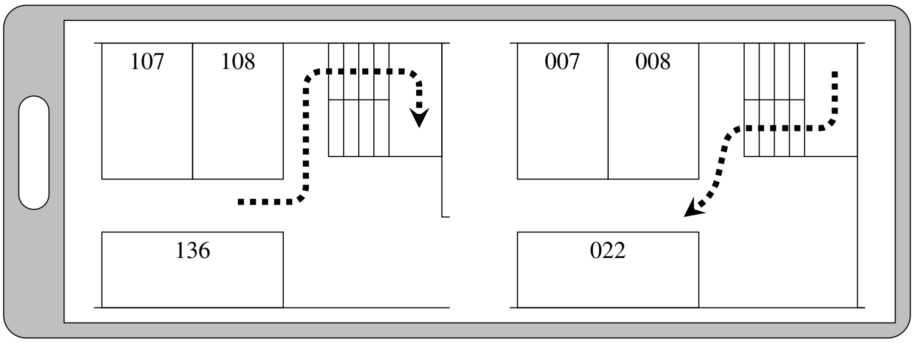

Fig. 6. A smartphone with the navigation application

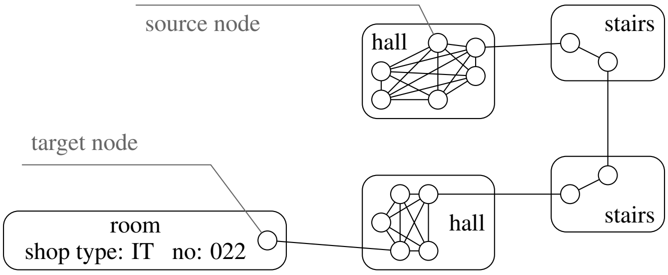

Fig. 7. One of temporary subgraphs used by the navigation algorithm (edge costs omitted)

be used by an enhanced navigation algorithm to narrow the set of rooms to be visited, but it would require the technician to specify the type of a shop he is looking for, which would consist an additional input parameter of the *SearchForShop* procedure.

In the presented approach, it is assumed that architectural building designs are created by means of CAD tools with the use of Building Information Modeling (BIM) technology. As the IFC (Industry Foundation Classes) format [39] is an interoperable BIM standard for CAD applications, a graph model of the considered building can be created by extracting the information about the topology of its spatial layout from the IFC 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. However grouping nodes into other hierarchies than the spatial one have to be done later by the person responsible for the maintenance of the building information (BMS operator).

Therefore the BMS operator has to take care of modifications in the building design structure or functionality during its lifecycle in order to maintain the consistency of multi-hierarchical graph representation of the building with its current status. The proposed formalism has an additional capability of supporting the preservation of information consistency and integrity. For example, when a room is divided into two new ones, the system can provide the additional level of securing correct assignment of information. It assigns to new rooms places in both functional and/or administrative hierarchies. In a similar situation of adding an extension to the building (new rooms) the formalism will also help to assure that all required information is provided.

However, there are some situations which require our formalism to be adjusted to cope with them. For example in case when two staff members work in the same room, and one of them changes the department in which he works, a problem arises as the same room can not be assigned to two different functional units, as an in graph terms it would mean that one node has two parents in the same hierarchy. In some cases this problem can be solved by assigning a room to a functional unit at the higher level of the hierarchy, in others it could be virtually divided into two parts.

VI. CONCLUSION

This paper deals with supporting effective decision making in building management by assisting maintenance processes. As building maintenance require integration of various types of information and knowledge, the multi-hierarchical graph representation of buildings was proposed. This representation allows us to express different types of hierarchical dependencies between building parts, like geometrical and functional ones, in one structure. The additional knowledge concerning properties of building components is stored in attributes of the graph nodes corresponding to these components.

The described multi-hierarchical graph-based building model is useful in facility maintenance aspects. It provides the possibility of filtering the graph by the hierarchy type and level, eliminating knowledge not related to the given task. On the one hand, it helps in finding information constituting answers to complex queries referring to different types of information. On the other hand, it helps in finding a path between a starting point and a dynamically selected target point in the chosen fragment of the building, by enabling to consider only relevant subgraphs specified by selected hierarchies. In both cases the number of data to be analyzed is greatly reduced.

The considered problems were presented on examples set in the environment typical for the campus/educational establishments. The proposed solutions can be used with minor changes for BM systems of public buildings or other complex buildings where people have to be navigated. The proposed model can be extended in several ways. It was used in this paper as a knowledge base for the navigation within the building on the basis of the shortest path algorithm. It can also be used for creating tools which would support persons with disabilities helping the to find the easiest/most accessible route (not always the shortest).

REFERENCES

[1] D. J. Vanier, "Advanced asset management: Tools and techniques," *Innovations in Urban Infrastructure Seminar of the APWA International Public Works Congress*, pp. 39–56, 2000.

[2] J. C. P. Cheng, W. Chen, Y. Tan, and M. Wang, "A BIM-based decision support system framework for predictive maintenance management of building facilities," in *Proceedings of the 16th International Conference on Computing in Civil and Building Engineering (ICCCBE2016)*, 2016.

[3] I. Motawa and A. Almarshad, "A knowledge-based BIM system for building maintenance," *Automation in Construction*, vol. 29, pp. 173–182, 2013. doi: 10.1016/j.autcon.2012.09.008

[4] B. Köbben, A. H. van Bunningen, and K. Muthukrishnan, "Wireless campus LBS: Building campus-wide location based services based on WiFi technology," in *Geographic Hypermedia: Concepts and Systems*, E. Stefanakis, M. P. Peterson, C. Armenakis, and V. Delis, Eds. Springer, 2006, pp. 399–408. ISBN 978-3-540-34238-0

[5] H. M. Khoury and V. R. Kamat, "Evaluation of position tracking technologies for user localization in indoor construction environments," *Automation in Construction*, vol. 18, no. 4, pp. 444–457, 2009. doi: 10.1016/j.autcon.2008.10.011

[6] A. Basiri, P. Amirian, A. Winstanley, S. Marsh, T. Moore, and G. Gales, "Seamless pedestrian positioning and navigation using landmarks," *The Journal of Navigation*, vol. 69, no. 1, pp. 24–40, 2016. doi: 10.1017/S0373463315000442

[7] D. Büchel and P.-Y. Gilliéron, "Pedestrian navigation inside buildings," *Géomatique Suisse*, vol. 11/2004, pp. 664–668, 2004, (in French).

[8] C. Faure, P. Benci, A. Danzart, and E. Lecolinet, "Design of mobile services for students," in *Proceedings of the UbiMob'06 Conference*, 2006, (in French).

[9] Y. A. Wahab and A. S. H. Basari, "Building maintenance management preliminary finding of a case study in ICYM," *Middle-East Journal of Scientific Research*, vol. 17, no. 9, pp. 1260–1268, 2013.

[10] P. Barrett and D. Baldry, *Facilities Management: Towards Best Practice, 2nd Edition*. Blackwell Science, Oxford, 2003.

[11] N. Ali, M. Sun, G. Aouad, R. M. Mazlan, and F. D. Mustapa, "Understanding the business process of reactive maintenance projects," in *Proceedings of the International Conference on Construction Industry*, 2006, june 21–25, Padang, Sumatera Barat Indonesia.

[12] P. Teicholz, Ed., *BIM for Facility Managers*. Wiley, 2013. ISBN 978-1-118-38281-3

[13] A. Akanmu, C. Anumba, and J. Messner, "Critical review of approaches to integrating virtual models and the physical construction," *International Journal of Construction Management*, vol. 14, no. 4, pp. 267–282, 2014. doi: 10.1080/15623599.2014.972021

[14] C. Legner and F. Thiesse, "RFID-based maintenance at Frankfurt airport," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 34–39, 2006. doi: 10.1109/MPRV.2006.14

[15] C.-H. Ko, "RFID-based building maintenance system," *Automation in Construction*, vol. 18, no. 3, pp. 275–284, 2009. doi: 10.1016/j.autcon.2008.09.001

[16] B. Strug, A. Paszyńska, M. Paszyński, and E. Grabska, "Using a graph grammar system in the finite element method," *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 4, pp. 839–853, 2013.

[17] M. Minas, "Concepts and realization of a diagram editor generator based on hypergraph transformation," *Science of Computer Programming*, vol. 44, no. 2, pp. 157–180, 2002.

[18] E. Grabska, A. Łachwa, G. Ślusarczyk, K. Grzesiak-Kopeć, and J. Lembas, "Hierarchical layout hypergraph operations and diagrammatic reasoning," *Machine GRAPHICS & VISION*, vol. 16, no. 1/2, pp. 23–38, 2007.

[19] E. Grabska, G. Ślusarczyk, and Sz. Gajek, "Knowledge representation for human-computer interaction in a system supporting conceptual design," *Fundamenta Informaticae*, vol. 124, pp. 91–110, 2013.

[20] B. Strug, E. Grabska, and G. Ślusarczyk, "Supporting the design process with hypergraph genetic operators," *Advanced Engineering Informatics*, vol. 28, pp. 11–27, 2014.

[21] E. Grabska, W. Palacz, B. Strug, and G. Ślusarczyk, "A graph-based generation of virtual grids," *Lecture Notes in Computer Science*, vol. 7203, pp. 451–460, 2012.

[22] R. Milner, "Bigraphs and their algebra," *Electronic Notes in Theoretical Computer Science*, vol. 209, pp. 5–19, 2008.

[23] G. Ferrari and U. Montanari, "Tile formats for located and mobile systems," *Information and Computation*, vol. 156, no. 1-2, pp. 173–235, 2000.

[24] R. Bruni, U. Montanari, G. Plotkin, and D. Terreni, "On hierarchical graphs: reconciling bigraphs, gs-monoidal theories and gs-graphs," in *13th Italian Conference on Theoretical Computer Science*, 2012.

[25] H. Barendregt, M. C. J. D. van Eekelrn, J. R. W. Glauert, J. Kennaway, M. J. Plasmeijer, and M. R. Sleep, "Term graph rewriting," *Lecture Notes in Computer Science*, vol. 259, pp. 141–158, 1987.

[26] F. Drewes, B. Hoffmann, and D. Plump, "Hierarchical graph transformation," *Journal of Computer and System Sciences*, vol. 64, pp. 249–283, 2002.

[27] A. Poulovassilis and M. Levene, "A nested-graph model for the representation and manipulation of complex objects," *ACM Transactions on Information Systems*, vol. 12, pp. 35–68, 1994.

[28] J. F. Sowa, "Conceptual graphs," in *Handbook of Knowledge Representation*, F. van Harmelen, V. Lifschitz, and B. Porter, Eds. Elsevier, 2008, pp. 213–237.

[29] ——, "Conceptual graphs for a database interface," *IBM Journal of Research and Development*, vol. 20, no. 4, pp. 336–357, 1976.

[30] ——, *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.

[31] D. Harel, "On visual formalisms," *Communications of the ACM*, vol. 31, pp. 514–530, 1988.

[32] G. Chen and N. Zhong, "Granular structures in graphs," *Lecture Notes in Computer Science*, vol. 6954, pp. 649–658, 2011.

[33] ——, "Three granular structure models in graphs," *Lecture Notes in Computer Science*, vol. 7414, pp. 351–358, 2012.

[34] E. Grabska and G. Ślusarczyk, "Knowledge and reasoning in design systems," *Automation in Construction*, vol. 20, no. 7, pp. 927–934, 2011. doi: 10.1016/j.autcon.2011.03.009

[35] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms (2nd ed.)*. MIT Press, 2001.

[36] A. Kneidl, A. Borrmann, and D. Hartmann, "Generating sparse navigation graphs for microscopic pedestrian simulation models," in *18th EG-ICE International Workshop*, Twente, Netherlands, 2011.

[37] M. Höcker, V. Berkhahn, A. Kneidl, A. Borrmann, and W. Klein, "Graph-based approaches for simulating pedestrian dynamics in building models," in *8th European Conference on Product & Process Modelling (ECPPM)*. Cork, Ireland: University College Cork, 2010.

[38] E. Whiting, J. Battat, and S. Teller, "Topology of urban environments: Graph construction from multi-building floor plan data," in *Computer-Aided Architectural Design Futures 2007: Proceedings of the 12th International CAAD Futures Conference*. Dordrecht, Netherlands: Springer, 2007, pp. 115–128.

[39] "buildingSMART IFC," http://www.buildingsmart.org/standards/ifc, 2015, accessed on September 24, 2015.