Queries on Semantic Building Digital Twins for Robot Navigation

Rens de Koning[1][0000-0001-9645-7758], Elena Torta[1][0000-0001-9198-1374], Pieter Pauwels[2][0000-0001-8020-4609], R.W.M. Hendrikx[1][0000-0002-1010-8985], and M.J.G. van de Molengraft[1][0000-0002-5095-4297]

[1] Faculty of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
[2] Faculty of Built Environment, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
`r.w.d.koning@student.tue.nl`,
`{e.torta,p.pauwels,r.w.m.hendrikx,m.j.g.v.d.molengraft@tue.nl>}@tue.nl`

Abstract. Autonomous mobile robots are starting to be deployed in complex built environments where they need to navigate to complete the given tasks. In order to navigate, autonomous mobile robots often rely on environmental maps. In this paper, we explore a novel approach to automatically create topological and metric environmental maps from BIM data exported to a graph database. We define queries on the exported graph data-base which retrieve the data needed to create the maps automatically. We validate our approach by applying standard path planning algorithms such as A* on the generated maps showing that they are suitable for computing optimal paths. We regard this work as a first step to connect linked data methods to robotics algorithms and use-cases. The results show the feasibility and potential of exploiting the semantic richness of the data available from BIM.

Keywords: Linked Data · Semantics · Robot navigation · 2D geometry · mapping

1 Introduction

Autonomous mobile robots are operating more and more in complex built environments where they need to navigate from their current position to a designated position. To navigate, a robot often relies on an environmental map which can take the form of an occupancy grid map [6] or, more recently, of a semantic map in which geometrical information, as well as semantic information, is reported [18]. In order to obtain a map, the robot needs to capture sensor data that cover all spaces in the building. During this process, the robot scans the area around it with its sensors, most often 2D or 3D lidars, simultaneously creating a map and localizing within it. This is commonly called SLAM: Simultaneous Localization And Mapping (see [9] for a comparison study of different SLAM approaches). Alongside the obvious advantage of not relying on any prior knowledge of the building, SLAM has the disadvantage that it requires an operator to move the robot around the unexplored building to construct the map. Additionally, dynamic elements (e.g. movable furniture) can be included in the map making it obsolete over time (e.g., when dynamic elements change position requiring constant map updates [25]). Maps generated by state-of-the-art SLAM algorithms (i.e. GMapping [10], HectorSLAM [17] and Cartographer [15]) also lack semantic details since environmental elements are only represented as geometric objects without describing what such objects are. For example, a robot could scan a wall detecting an opening without knowing that the opening represents a door.

In this paper, we propose an alternative method of automatically constructing maps for robot navigation. We demonstrate how spatial and topological maps of a building can be created by querying data from a building digital twin realized in the form of a RDF graph database (see [2] for a review). The content of the RDF graph database can be generated by exporting relevant data from the BIM [3] of the targeted building. The approach is attractive because it has the potential to create spatial and semantic maps for robot navigation seamlessly from already available building data without the need of human intervention to create such maps.

The maps derived by applying the proposed queries to extract relevant data and subsequent algorithms for map creation are dependent only on the ontology adopted to organize the data in the RDF graph (i.e. LBD ontologies[3]) but are independent from the modelling convention adopted when creating the BIM. In this way, knowledge of the BIM modelling convention of a particular building is decoupled from knowledge about how these data can be used by robots. We demonstrate the approach with a concrete use-case: plan the optimal path to move a robot between two rooms of the Atlas building of the Eindhoven University of Technology campus. The RDF graph, the queries and the resulting maps are available in a public code repository that accompanies this paper[4].

The paper is organized as follows. Section 2 presents related work on the use of BIM data for robotic navigation. Section 3 presents the queries and the algorithms used to derive metric and topological maps for robot navigation. Section 4 demonstrates the outcome of the proposed methods in terms of robot path planning. Finally, Section 5 proposes a reflection on the proposed approaches and outlines future work.

2 Related Work

In the past years there have been a few attempts to leverage the rich data that BIM models provide to improve typical functions of autonomous robots such as localization and navigation. With respect to localization, Acharya et al. [1] have proposed a method to generate a data-set of synthetic images with associated known 6-DOF camera locations and orientations that can be used to train Deep Convolutional Neural Network (DCNN) for robot localization. Similarly, the

[3] `https://www.w3.org/community/lbd/`

[4] `https://gitlab.tue.nl/et_projects/rk-semantic-queries.git`

work of [11] generates a data set of synthetic images from a BIM, trains a DCNN on those images to extract features. Extracted features are compared with features extracted from real images to estimate which location in the BIM is more likely to correspond to the real images.

Other work has focused on using the information from BIM to derive topological maps from which paths can be planned. In [24], the authors propose to extract information from BIM to set-up a simulation environment (VEROSIM) for robotics development. The environment can connect the OMPL (Open Motion Planning Library [26]) to the imported BIM model to generate collision free paths. On a similar line, [19] derives a topological graph from BIM models upon which an A* planner can retrieve the optimal path. These works focus on either the direct import of BIM models in simulation environments [24] or on its direct usage for path planning [19]. Recently an automatic way of exporting data from the Industry Foundation Classes (IFC-JSON) to metric map for robot localization has been demonstrated by [14], the work, however does not rely on a graph database to extract relevant information and focuses on localization only.

The novel contribution of this paper lies in the definition of queries on a building digital twin (i.e. an RDF graph database) rather than on direct usage of BIM or IFC exports for either localization or navigation. Contrary to a BIM or an IFC export, a building digital twin is considered a living entity and therefore has the potential to be updated during robot operation providing constantly updated information which is essential for reliable long term deployment of autonomous robots. This work furthermore aims to align as good as possible with ongoing developments in terms of LBD ontologies, mainly because they have the potential to provide a real-time representation of building topology and product data linked to 2D and 3D geometric data [8, 27].

3 Method

3.1 Creation of the building digital twin

A building digital twin is a digital representation of a building with real-time data connection. The format of the building digital twin proposed in this paper is an RDF graph database implemented in GraphDB[5]. The data from the initial BIM model is exported to an RDF graph database, following the Linked Data approach [5, 4, 7, 13], with a custom REVIT plugin created by the authors and available in a public code repository [6]. The building chosen as use-case to create the RDF database, apply queries and derive maps for robotic navigation is the Atlas building of the Eindhoven University of Technology campus. A view of Atlas and of its BIM model is shown in Figure 1.

The data exported into TTL format is used to create topological and metric maps for robot navigation. From the exported data, only a subset is used to create such maps which is reported in Table 1. The selected data mostly represent

[5] `https://graphdb.ontotext.com/`

[6] `https://github.com/pipauwel/IFCtoLBD` - development ongoing

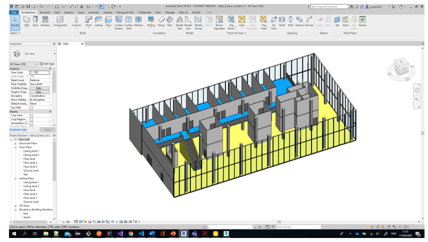


Fig. 1. Left: View of the Eindhoven University of Technology main building (Atlas). Right: BIM model of Atlas (only a part of 8th floor displayed).

geometrical elements of spaces including their semantic (e.g., columns, walls, doors and curtainwalls) and more abstract topological information such as room identification and connectivity. Data follows the BOT [21, 22], BEO and MEP ontologies [20] with 2D geometry represented as Well Known Text (WKT) literals according to recommendations in [8, 27]. A snapshot of the selected RDF data of Atlas is reported in Listing 1.1.

```
inst:space_2936
  a bot:Space ;
  bot:adjacentElement inst:wall_258992 ;
  bot:adjacentElement inst:wall_256212 ;
  bot:adjacentElement inst:door_283489 ;
  bot:adjacentElement inst:door_259071 ;
  props:number "10"^^xsd:string .

inst:wall_258992
  a bot:Element ;
  a beo:Wall .

inst:Interface_79
  a bot:Interface ;
  bot:interfaceOf inst:space_2936, inst:wall_258992 ;
  geom:asWKT "LINESTRING (199140.100211374 -40973.2993467975,
  202425.600211374 -40973.2993467975)" .
```

Listing 1.1. Snapshot of the Atlas data exported to the RDF database

The full RDF graph database on which this paper is based is available in a public code repository[4].

3.2 Construction of topological maps

A topological map abstracts metric information and represents, in a bidirectional graph (unlike the RDF graphs), how spaces are connected to each other. In topological graphs for robot navigation, nodes represent spaces, edges that connect nodes represent a direct connection between two spaces that can be navigated by a robot. For example, when a room is connected to a corridor via a door, the room and the corridor would be represented as nodes with a bidirectional edge. The edges are thus identical to the bot:Interfaces in the LBD graph. Edges are

Code in RDF	Type	Description
	Class definitions	
bot:Space	node	class of type space (BOT ontology)
bot:Interface	node	class of type Interface (BOT ontology)
bot:adjacentElement	edge	sub elements of a subject (BOT ontology)
beo:Wall	node	type definition of wall (BEO ontology)
beo:Door__DOOR	node	type definition of door (BEO ontology)
beo:Column__COLUMN	node	type definition of column (BEO ontology)
beo:CurtainWall	node	type definition of curtainwalls (BEO ontology)
props:number	edge	Identification number of a subject (e.g. space)
props:level	edge	Floor containing the subject
geom:asWKT	edge	2D coordinates of an object (WKT representation)
	Instances example	
inst:space_xx	node	Space instance
inst:interface_xx	node	Interface instance
inst:wall_xx	node	Wall instance
inst:door_xx	node	Door instance
inst:column_xx	node	Column instance
inst:curtainWall_xx	node	Curtain wall instance

Table 1. Overview of the RDF classes and instances that are relevant for this paper. Classes are either defined in a BOT ontology [21] or a BEO ontology[20]; geometry representation is done in a simple WKT string literal with a local coordinate reference system.

labelled with a cost which represents the effort needed to go from one space to the other. A typical example of effort is the metric distance between two adjacent nodes. When a topological map is available, it can be used for path planning, i.e., a robot can compute the optimal path to go from an initial space to a target space by minimizing the cost [23].

In order to construct a topological map from the building digital twin, we start from the observation that two spaces are connected (i.e. they share an edge in the topological graph) if they share a door. From this observation, the first query reports all spaces X and Y of a certain level of the building that have as adjacent element the same door. This is realized by the SPARQL code available in the accompanying public code repository[4] whose pseudocode is reported in Algorithm 1. A graphical visualization of the derived topological map is shown in Figure 2. It is important to notice that the topological map obtained is not yet ready to be used for path planning because there is no cost associated to the edges. By deriving a metric map of the environment we are also able to derive such a cost based on metric distance and complete the topological map. The method proposed to derive a metric map is described in Section 3.3.

3.3 Construction of metric maps

A metric map describes the geometrical layout of a space which is mostly defined by structural elements such as walls, curtain walls, columns and doors. A metric map is commonly used by robots for different purposes such as localization and path planning [23].

It is important to notice that the metric map derived by the proposed approach incorporates structural information only and does not represent movable

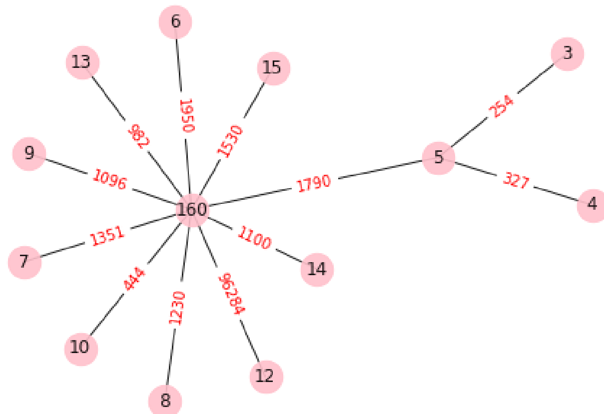


Fig. 2. Partial topological map of the Atlas building. Nodes represent rooms. Edges are labelled with costs which represent the distance between two adjacent nodes expressed in *dm* as outlined in Section 3.3.

Algorithm 1 Query for the extraction of the topological map. All spaces that share a door are extracted. The SPARQL implementation is available in the accompanying public code repository[4].

```
Select spaceX , spaceY and door
Where
    spaceX is on building level N
    spaceX has at least one adjacent element of type 'Door'
    geometry of 'Door' is assigned to variable door
    spaceY is on building level N
    spaceY has door as adjacent element
    If spaceX is equal to spaceY Then discard the tuple
```

furniture such as bookshelves, tables, chairs and beds, as this information was not modelled nor exported from the BIM model. The latter could be included in metric maps when a robot recognizes the presence of such objects by its perception algorithms and updates the building digital twin with this new information. It is also important to notice that the geometry reported in a metric map can be 2D or 3D. Data exported from a BIM model can support both types, yet, in the research presented in this paper, we only consider 2D geometry. This 2D geometry can easily be obtained using the Revit API by retrieving all elements bounding a space, and then retrieving its 2D line representations.

The metric map is constructed by extracting the geometry of each space that composes a building (= boundary lines of bot:interfaces). The SPARQL query to retrieve such information with related geometry description per space of the building is available in the accompanying public code repository [4]. The pseudo-code of the query is reported in Algorithm 2.

Algorithm 2 Query for the extraction of the metric map. All structural elements that are interfaces of a space are retrieved with their 2D geometry. The SPARQL implementation is available in the accompanying public code repository[4].

```
Select space, walls, curtainwalls, doors and columns
Where
    element is a wall
    element is an interface of the space
    assign 2D geometry of the interface to variable walls
    Union
        element is a curtainwall
        element is an interface of the space
        assign 2D geometry of the interface to variable curtainwalls
    Union
        element is a door
        element is an interface of the space
        assign 2D geometry of the interface to variable doors
    Union
        element is a column
        element is an interface of the space
        assign 2D geometry of the interface to variable columns
```

A partial visualization of the metric map derived by applying Algorithm 2 to the building digital twin of Atlas is shown in Figure 3. In this paper, the metric map is used to assign a cost to the edges of the topological graph following the procedure reported in Algorithm 3.

Algorithm 3 Algorithm used to assign a cost to the edges between adjacent nodes of the topological graph. The implementation is available in the accompanying public code repository[4].

find all the adjacent elements (walls, columns and curtainwalls) belonging to spaceA
find all the adjacent elements (walls, columns and curtainwalls) belonging to spaceB
Get midpoint coordinates for both spaceA ($A_x A_y$) and spaceB ($B_x B_y$) by acquiring their average X and Y coordinates
find the midpoint coordinates of the door ($d_x d_y$) connecting spaceA and spaceB
Determine a cost by taking the shortest distance from midpoint spaceA to the door and from the door to spaceB $\sqrt{(A_X - d_x)^2 + (A_y - d_y)^2} + \sqrt{(B_x - d_x)^2 + (B_y - d_y)^2}$

4 Results

In this section, we show how the derived topological and metric maps can be used to compute an optimal path to allow the robot to navigate between different spaces in the building.

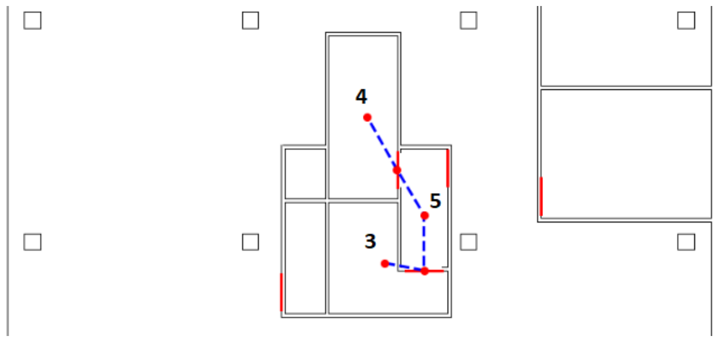


Fig. 3. Partial metric map. Red dots represent the middle point of each space or door. The dashed line indicates the Euclidean distance between points.

4.1 Computing the cost-optimal sequence of spaces

The topological map derived in Section 3.2 with related costs derived in Section 3.3 can be used to determine the most cost effective path to navigate from a space X to a space Y in a building as a sequence of spaces to be visited. As an example, we compute the shortest path to go from space 3 to space 4 by applying the A* algorithm [23] to the topological map that is (partially) shown in Figure 2. The result is that the optimal space sequence is space 3 followed by space 5 followed by space 4.

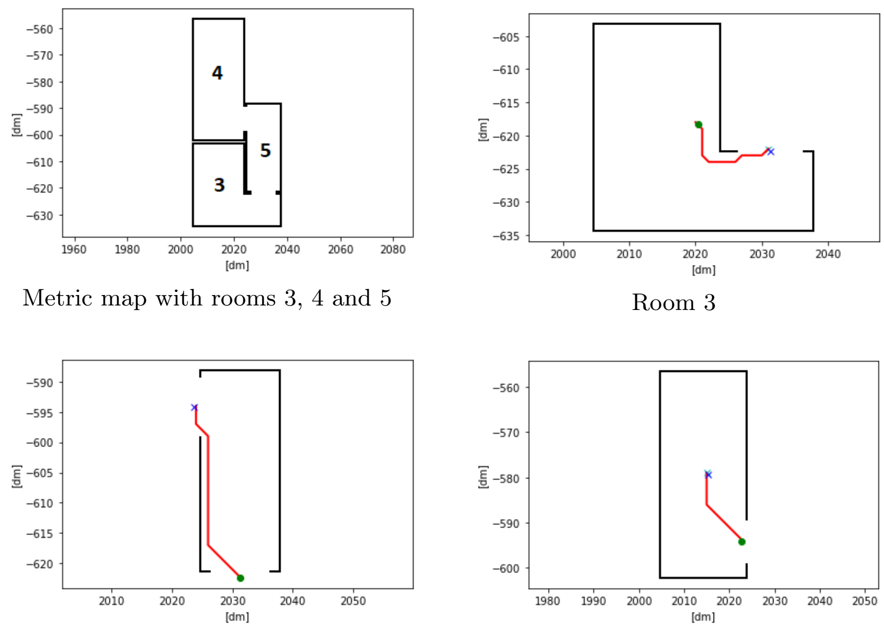
4.2 Computing the optimal path within a space

Once the order of spaces to be visited is known, a robot can compute an optimal path to navigate within each space. To this end, the metric map is discretized and converted to an occupancy grid map, i.e. the map is converted into a grid with walls, curtain walls and columns indicating space a robot cannot traverse and all the rest indicating space the robot can traverse. The chosen dimension of each square cell is 1 dm. The obtained occupancy grid map is input to the A* algorithm [12] which is used to compute the shortest path within a space preventing robot collisions with structural obstacles. Note that doors are considered traversable space.

5 Discussion and conclusion

We presented queries to extract topological and metric maps from a building digital twin represented by an RDF graph database populated by data extracted from a BIM model. We demonstrated that metric and topological maps can be derived from the extracted data and used for robotic path planning.

The connection between building digital twin and robotics is still in its infancy though and several aspects of the work presented call for further investigation. The initial data from a BIM model might not match the actual layout of



Metric map with rooms 3, 4 and 5

Room 3

Room 5

Room 4

Fig. 4. Sequences of generated path per space. The green circle represents the robot starting point, the red line is the computed optimal path, the black lines are structural obstacles, the x is the destination point. Grid discretization is set at 1 *dm*.

the building, small errors in dimensions as well as modelling (e.g., actual doors are not present in the original BIM model) are to be expected. In this sense we regard the building digital twin as a living representation of a building and corrections from the robot are to be expected. When and how to provide such corrections is left to future research.

The semantic information derived from the BIM model was used, in this work, to enrich a purely metric map which from which paths for robot navigation were derived. We can further exploit the semantic information for robot navigation by, for example, make prediction of humans' motion intentions in a similar way as reported by Houtman et al. [16].

The data extraction from BIM to the building digital twin depends on a REVIT plugin that was developed for this project. The plugin is dependent on the modelling convention adopted when creating the BIM and outputs data in a standard format. The quality of data as well as the effort needed to create such a plugin might depend largely on the BIM modelling convention. Providing and following specific guidelines would be beneficial to speed up the use of BIM and building digital twins for autonomous robot navigation. Alternatively, the use of IFC could be considered, as was previously investigated in [14], yet also the quality of this export depends a lot on the same modelling guidelines and does not really resolve that specific challenge.

References

1. Acharya, D., Khoshelham, K., Winter, S.: BIM-PoseNet: Indoor camera localisation using a 3D indoor model and deep learning from synthetic images. ISPRS Journal of Photogrammetry and Remote Sensing **150**, 245–258 (apr 2019). https://doi.org/10.1016/j.isprsjprs.2019.02.020
2. Angles, R., Gutierrez, C.: Survey of graph database models. ACM Computing Surveys (CSUR) **40**(1), 1–39 (2008)
3. Azhar, S.: Building information modeling (bim): Trends, benefits, risks, and challenges for the aec industry. Leadership and management in engineering **11**(3), 241–252 (2011)
4. Berners-Lee, T.: Linked data - design issues (2006), `https://www.w3.org/DesignIssues/LinkedData.html`
5. Berners-Lee, T., Hendler, J., Lassila, O., et al.: The semantic web. Scientific american **284**(5), 28–37 (2001)
6. Birk, A., Carpin, S.: Merging occupancy grid maps from multiple robots. Proceedings of the IEEE **94**(7), 1384–1397 (2006)
7. Bizer, C., Heath, T., Berners-Lee, T.: Linked data-the story so far. International journal on Semantic Web and Information Systems **5**(3), 1–22 (2009). https://doi.org/10.4018/jswis.2009081901
8. Bonduel, M., Wagner, A., Pauwels, P., Vergauwen, M., Klein, R.: Including widespread geometry schemas into linked data-based bim applied to built heritage. Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction **172**(1), 34–51 (2019). https://doi.org/10.1680/jsmic.19.00014
9. Filipenko, M., Afanasyev, I.: Comparison of various slam systems for mobile robot in an indoor environment. In: 2018 International Conference on Intelligent Systems (IS). pp. 400–407. IEEE (2018)
10. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE transactions on Robotics **23**(1), 34–46 (2007)
11. Ha, I., Kim, H., Park, S., Kim, H.: Image retrieval using BIM and features from pretrained VGG network for indoor localization. Building and Environment **140**, 23–31 (aug 2018). https://doi.org/10.1016/j.buildenv.2018.05.026
12. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE transactions on Systems Science and Cybernetics **4**(2), 100–107 (1968)
13. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology **1**(1), 1–136 (2 2011). https://doi.org/10.2200/S00334ED1V01Y201102WBE001
14. Hendrikx, B., Pauwels, P., Torta, E., van de Molengraft, R., Bruyninckx, H.: Connecting Semantic Building Information Models and Robotics: An application to 2D LiDAR-based localization. In: IEEE International Conference on Robotics and Automation (ICRA2021) (may 2021)
15. Hess, W., Kohler, D., Rapp, H., Andor, D.: Real-time loop closure in 2d lidar slam. In: 2016 IEEE International Conference on Robotics and Automation ICRA. pp. 1271–1278. IEEE (2016)

16. Houtman, W., Bijlenga, G., Torta, E., Molengraft, R.v.d.: A probabilistic model for real-time semantic prediction of human motion intentions from rgbd-data. Sensors **21**(12), 4141 (2021)
17. Kohlbrecher, S., Von Stryk, O., Meyer, J., Klingauf, U.: A flexible and scalable slam system with full 3d motion estimation. In: 2011 IEEE international symposium on safety, security, and rescue robotics. pp. 155–160. IEEE (2011)
18. Kostavelis, I., Charalampous, K., Gasteratos, A., Tsotsos, J.K.: Robot navigation via spatial and temporal coherent semantic maps. Engineering Applications of Artificial Intelligence **48**, 173–187 (2016)
19. Palacz, W., Ślusarczyk, G., Strug, B., Grabska, E.: Indoor robot navigation using graph models based on BIM/IFC. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 11509 LNAI, pp. 654–665. Springer Verlag (jun 2019)
20. Pauwels, P.: The Building Element Ontology. `https://pi.pauwel.be/voc/buildingelement/index-en.html`
21. Rasmussen, M.H., Lefrançois, M., Schneider, G.F., Pauwels, P.: Bot: the building topology ontology of the w3c linked building data group. Semantic Web (Preprint), 1–19 (2019)
22. Rasmussen, M., Lefrançois, M., Schneider, G., Pauwels, P.: Bot: The building topology ontology of the w3c linked building data group. Semantic Web **12**(1), 143–161 (2020). https://doi.org/10.3233/SW-200385
23. Sariff, N., Buniyamin, N.: An overview of autonomous mobile robot path planning algorithms. In: 2006 4th student conference on research and development. pp. 183–188. IEEE (2006)
24. Schlette, C., Roßmann, J.: Sampling-based floor plan analysis on bims. In: Proceedings of the 33rd International Symposium on Automation and Robotics in Construction (ISARC). pp. 28–35. International Association for Automation and Robotics in Construction (IAARC) (July 2016)
25. Shaik, N., Liebig, T., Kirsch, C., Müller, H.: Dynamic map update of non-static facility logistics environment with a multi-robot system. In: Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz). pp. 249–261. Springer (2017)
26. Sucan, I.A., Moll, M., Kavraki, L.E.: The open motion planning library. IEEE Robotics & Automation Magazine **19**(4), 72–82 (2012)
27. Wagner, A., Bonduel, M., Pauwels, P., Rüppel, U.: Representing construction-related geometry in a semantic web context: A review of approaches. Automation in Construction **115**, 103130 (2020). https://doi.org/https://doi.org/10.1016/j.autcon.2020.103130, `https://www.sciencedirect.com/science/article/pii/S0926580519307125`