Semantic mapping extension for OpenStreetMap applied to indoor robot navigation

Lakshadeep Naik[1], Sebastian Blumenthal[2], Nico Huebel[3], Herman Bruyninckx[3], and Erwin Prassler[1]

***Abstract*— In this work a graph-based, semantic mapping approach for indoor robotics applications is presented, which is extending OpenStreetMap (OSM) with robotic-specific, semantic, topological, and geometrical information. Models are introduced for basic indoor structures such as walls, doors, corridors, elevators, etc. The architectural principles support composition with additional domain and application-specific knowledge. As an example, a model for an *area* is introduced, and it is explained how this can be used in navigation. A key advantage of the proposed graph-based map representation is that it allows exploiting the hierarchical structure of the graphs. Finally, the compatibility of the approach with existing, grid-based motion planning algorithms is shown.**

I. INTRODUCTION

Complex service robotic applications have to deal with changing environments, e.g., doors can be open, closed, or at any state in-between, pallets in a factory or warehouse can be added, moved, or removed. Such changes lead to vastly different sensory input for robots. Traditional maps used in indoor robotics focus solely on the (static) spatial representation of the environment [1] and leave dynamic obstacles to motion planning. This makes it hard to cope with changing environments. Therefore, a representation for maps is required that models, not just static spatial information but also semantic information that allows to reason about the state of the environment like open or closed doors.

In addition, complex service robotic applications need to deal with a large variety of tasks in different contexts like hospitals, industrial environments, or elderly care facilities to name just a few. This requires application and environment specific knowledge.

Graphs fulfill both requirements. Domain-specific knowledge can be encoded in separate graphs and then be composed to an application by adding the application specific relations between the different graphs.

There are also existing graph-based map representations, which use vector data such as points, lines, polygons to represent geometric details of an environment and add semantic information to model different entities present in an environment. While the models presented in this work fit into any graph-based map representation, the indoor representation [2] of OpenStreetMap (OSM) was selected for the following reasons: a) it is open-source, b) the OSM data model conforms to a graph model and OSM contains lots of data and semantic tags, c) it has been successfully used for robotic outdoor applications [3]–[8], and d) it has tools supporting the development and usage of the models. Furthermore, it has an established community as well as company-driven use cases for indoor mapping such as OpenStationMaps. However, these maps are made for humans, who can deal with incomplete information. Robots require a more precise modelling of their environment, which is presented in this work.

The remainder of the paper is structured as follows. The following Section II is discussing related work. Section III introduces the suggested models and Section IV explains how the models can be applied to an indoor navigation use case. Finally, Section V concludes the paper with a discussion and explains planned additions to the models.

II. RELATED WORK

OpenStreetMap [9] is an open source, collaborative project for mapping outdoor environments. Users create and edit geographic data. OpenStreetMap has already attracted interest in outdoor robotic applications [3]–[8], [10]–[13]. Those approaches make use of existing OpenStreetMap data for robot localisation, path planning and navigation. OpenStreetMap is most commonly used for outdoor environments. However, there exist some community effort for mapping indoor environments. Simple Indoor Tagging [2] provides an elementary model for mapping indoor entities and also extensions for certain domains exit, e.g., OpenStationMap [14] extends the Simple Indoor Tagging model to map train stations across the world. Other efforts [15], [16] by the OpenStreetMap community focus on the 3D visualisation of an indoor environment.

Simple Indoor Tagging and OpenStationMap focus on modelling indoor environments for human navigation. For example, a door is represented as a single point. However, a robot needs to know where exactly it is located, the type of the door and how to open it. By modelling it as a connection, it can also be used in path planning as an entry/exit point to a room. Furthermore, by modelling its perception features (like color) it can also be used to aid localization. Finally, some actions, like waiting in front of the door, should be avoided. Such semantic information is currently not modelled in OSM. The models introduced in this work are the first step to alleviate this. We propose a semantic mapping approach, in which we model the static properties of different entities typically found in indoor environments. The model includes geometrical, topological,

[1] Bonn-Rhein-Sieg University of Applied Science, Sankt Augustin, Germany lakshadeep.naik@smail.inf.h-brs.de erwin.prassler@h-brs.de

[2] Locomotec GmbH, Germany blumenthal@locomotec.com

[3] Robotics Research Group, KU Leuven, Belgium, member of Flanders Make nico.huebel@kuleuven.be herman.bruyninckx@kuleuven.be

perceptual, and behavioural properties, which are useful for navigating a robot in a large indoor facility.

There is also existing work on semantic maps for indoor applications. Kostavelis et al. [17] provides an overview of available semantic mapping approaches for mobile robots. They classify semantic maps based on how they are created, if they contain a topological graph or not and how they make deductions about environment elements. Most of the existing semantic approaches such as [18]–[20] add semantic information on top of a geometric map and builds a topological graph based on detected semantic features and environment geometry. Lang et al. [21] present formal definition for semantic maps created by robots. They combine the classification of map entities with common-sense knowledge to improve human understanding. All these approaches follow a bottom-up approach to build the semantic map, while we follow a top-down approach.

The work of [22]–[24] shows a multi-hierarchical approach for semantic mapping, which has similarity to our approach. However, they extract semantic information from the sensors whereas in our approach we exploit existing semantic information in a map to improve navigation and task execution. Therefore, our approach is complementary to the work in these papers.

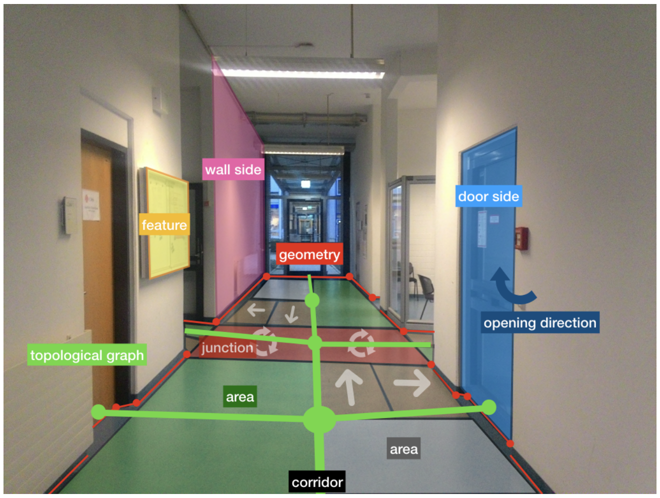


Fig. 1. Different type of information modelled in the proposed semantic map.

III. Proposed model

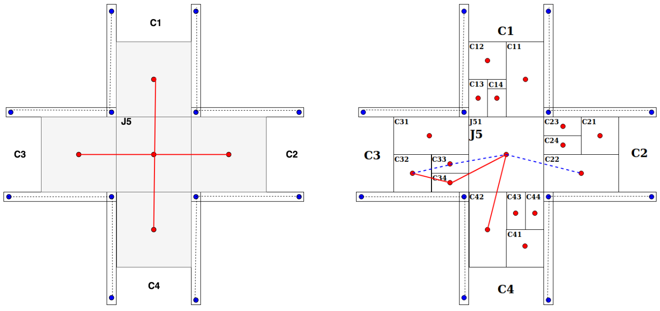
In this section, we present a model to represent different entities, which are typically part of an indoor environment. The models consist of entities and relations. The modelling process is divided into three steps: A) Building a *Domain Specific Modelling* identifies basic concepts in a given domain. B) *Logical modelling* involves identifying data types for entity attributes, different constraints on them, removing redundant attributes etc. It gives an abstract structure of the data model, but it is not specific to a particular technology. C) *Choosing a technology* involves representing the actual design of the data model using a particular technology.

A. Domain Specific Modelling

Geometrical, topological, semantic, perceptual and behavioural properties of an indoor environment are modelled as shown in Fig. 1. In digital maps, walls and doors are typically represented as the lines and points respectively. This information is sufficient for humans, but most of the existing navigation algorithms in robotics require very detailed geometry. Hence we capture the detailed geometry of the environment by representing walls and doors as polygons instead of a polyline. We introduce topological nodes for different indoor elements such as rooms, doors, corridors as shown in Fig. 2 (a). Each entity accessible from another entity is connected using a relation. These graphs can be used to plan human understandable paths.

Further, there exist dedicated *areas* in an indoor environment where a robot has to respect specific motion constraints such as non blocking areas in front of a door. We model such information by introducing virtual areas inside the rooms and corridors. Movement between these local areas is constrained using local topological graphs as shown in Fig. 2 (b). The lines represent the topological connections between the areas (not the path of the robot). The semantics are similar to a traffic system, where robots taking a right turn (red line) use area C34 while robots going straight (dashed blue line) are using area C33. This makes robot behaviour more predictable to people, who are familiar with traffic semantics.

Perception features such as wall/door material properties, objects like fire extinguishers, windows, etc., can be added as entities and their properties. Then robots can use them for feature-based navigation and task planning. We also model the behavioural properties of the environment such as the opening direction of the door, which is very important for robot navigation.



(a) (b)

Fig. 2. Path planning at different levels of abstraction. a) shows corridors, junctions and wall entities while b) further includes areas for navigation.

B. Logical modelling

In this subsection, we present logical models for different static entities found in an indoor environment. The models are represented using directed graphs. Arrows represent a 'has-a' relationship. Fig. 3 describes different color conventions used.

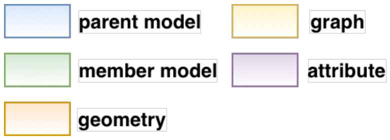


Fig. 3. Colour conventions used in the directed graphs.

1) Perception features: These are features present in an indoor environment such as windows, fire extinguishers, electric switches, etc., which robots can use as perception features. They are represented as points. Their dimensions and perception properties are modelled as attributes.

2) Door/Wall side: The two sides of doors or walls are usually not explicitly modelled in a digital map as it's not essential for human navigation. However, perception features like color can differ on two sides of a wall. It is modelled as the relation between two 'corner' entities to identify which side of the wall/door it represents can have relations to other perception 'feature's. 'Material' properties such as colour or texture are stored as attributes. Fig. 4 shows the logical data model of a door/wall side.

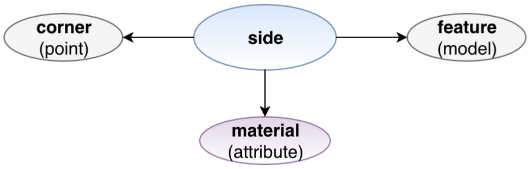


Fig. 4. Logical data model for a door/wall side.

3) Door/Wall: A wall is represented as a relation between a wall and its sides. The geometry of a wall is modelled as a polygon to capture its actual shape. This polygon is spanned by the corner nodes, which are also used to describe the sides. Fig. 5 shows the logical data model of a wall. A door is represented analogously to a wall except that it has additional behavioural properties such as an opening direction, type (automatic/manual door), etc.

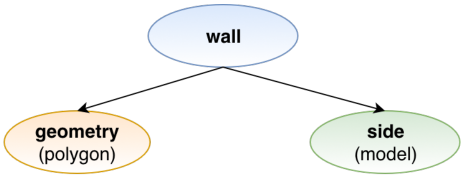


Fig. 5. Logical data model for a wall.

4) Area/Room/Corridor: An area/room/corridor is represented by its geometry and identifier. The identifier is human readable, e.g., a room number. The geometry is modelled as a polygon. Furthermore, all these models have relations to connections and areas. Connections link neighboring areas and represent movement constraints between them. Areas are task/domain dependent and impose constraints, e.g., speed limits. They also can contain other areas, meaning area models can be hierarchically composed (see Fig. 1). Room/corridor models are area models with additional semantics represented by additional attributes and relations, such as the relations to doors and walls forming them. Elevators/stairs are another type of areas with additional semantics represented by additional motion constraints and connections to areas on other floors. Fig. 6 shows the logical data model of an area/room/corridor entity.

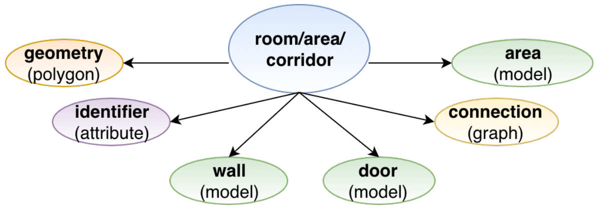


Fig. 6. Logical data model for a room/corridor

5) Level/floor: The model of a level/floor has a human understandable identifier in the form of an attribute and three types of relationships: the rooms/corridors it contains, additional walls not part of those rooms and corridors, as well as the connections, which model the relations between the rooms and corridors of the level and can be used for path planning on a higher level of abstraction (see Fig. 1). Fig. 7 shows the logical data model of a level/floor entity.

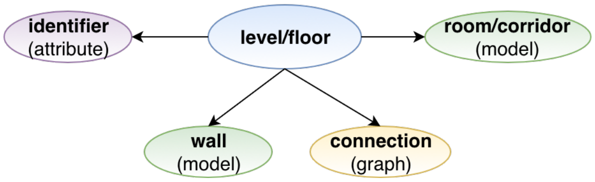


Fig. 7. Logical data model for a level/floor

6) Building: A building is modeled by its geometry (represented as a polygon) and attributes, such as its identifier, use and location, as well as the relations to the levels/floors it contains and the stairs/elevators connecting them. Fig. 8 shows the logical data model of a building.

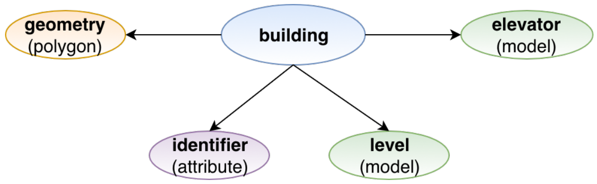
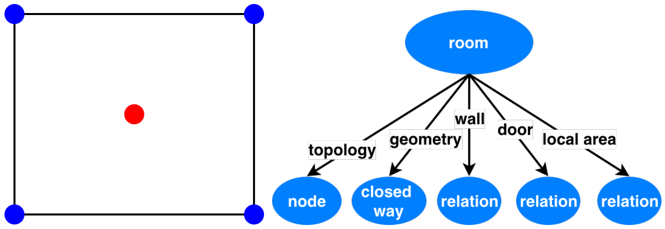


Fig. 8. Logical data model for a building

C. Choosing a technology

OpenStreetMap (OSM) elements [25] are basic components of OSM's conceptual data model of the physical world. They consist of nodes, ways, and relations. A node represents a specific point on the earth's surface defined by its latitude and longitude. A way is an ordered list of nodes that define a polyline. Ways are used to represent linear features such as rivers and roads. Ways can also represent the boundaries of areas (solid polygons) such as buildings, rooms etc. A relation is a multi-purpose data structure that documents a relationship between two or more data elements (nodes, ways, and/or other relations). Each element can optionally have a role within the relation. All types of the data element (nodes, ways and relations) can have tags. Tags describe the semantic meaning of the particular element to which they are attached. Tags are represented as key-value pairs. The proposed tagging schema for robotic indoor mapping is available online[1]. We also provide the *OSM Bridge Library*[2], which offers an API for querying data stored in the map and services for path planning or occupancy grid map generation.

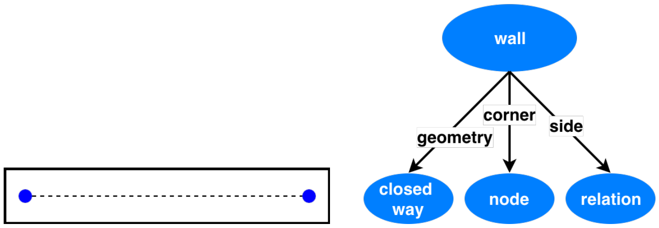
In the following an example for modelling a room, its walls, and their sides using proposed OSM mapping schema is presented. Fig. 9 shows an example for a room, which is modeled as an OSM relation. This relation connects the room's geometry, topological representation in form of a node (red node in Fig. 9 (a)), and its relations with its walls, doors, and areas. Its geometry is represented as an OSM closed way with room corner nodes as its members (blue nodes in Fig. 9 (a)). The connections between the room and other rooms/areas/corridors are represented by connecting its topological node to the topological nodes of other rooms, areas, or corridors using an OSM way (not shown in Fig. 9). The 'area' and 'door' relations will be treated as empty in this example. The 'wall' relation contains the unique IDs of the four walls of the room.



(a) Room geometry (b) Room relation

Fig. 9. Room modelling using proposed OSM tagging schema

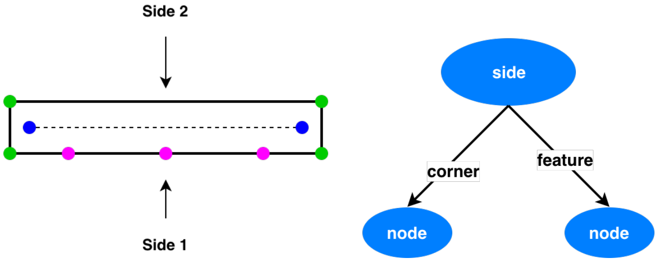
Each wall of the room is represented as a separate OSM relation as shown in Fig. 10. The wall geometry, wall corners, and its sides are a part of the wall relation. The geometry of the wall is represented as a closed way along the corners of the wall (green nodes in Fig. 11 (a)).



(a) Wall geometry (b) Wall relation

Fig. 10. Wall modelling using proposed OSM tagging schema

Each side of the wall is further represented as an OSM relation as shown in Fig. 11. It contains two types of nodes: corners(green nodes) and features (pink nodes). Corners refer to outer corners of a wall geometry and the 'side' relation uses two of them to identify, which side of the wall it represents. Often only two sides of a wall are visible, in which case the two not-visible 'side'-relations are ignored. In general, or model supports (runtime) composition. So unknown parts can be neglected and added only when they become necessary or available. Features refer to any features present on this side such as windows, fire extinguishers, or door tags. Features are represented as OSM nodes with their perception properties stored in its tags. Material properties of this side of the wall are also stored as tags of the 'side'-relation.



(a) Side geometry (b) Side relation

Fig. 11. Side modelling using proposed OSM tagging schema

IV. Results

The proposed approach uses a composable graph structure based on an extension of OSM. Our main hypothesis is: it is possible to use one map type for multiple robotic use cases. In this section, the usefulness of the model is shown in the context of robot *navigation*. The objective is to demonstrate: a) the map can be used for geometric navigation as an in-place replacement for grid maps and b) it can be used for generating waypoints for a robot to follow. Both cases are demonstrated in a qualitative manner.

A. Generation of grid maps

An OSM map for a basement in a hospital was created based on architectural drawings. For this experiment the

[1] https://github.com/ropod-project/indoor_osm_robot_wm
[2] https://github.com/ropod-project/osm_bridge

existing tool chain for OSM was used. The map had to be manually created with the help of an overlain architectural drawing, since an import of architectural CAD files was not available. An excerpt is depicted in Fig. 12.

An according grid map can be generated by rasterization of the edges for the wall geometries and the corridors. The areas circumscribed by the wall polygons are treated as *occupied* (black) cells, while the corridors are *free* (white) cells. The remaining cells are considered to be *unknown*. Fig. 13 shows the output in comparison to an occupancy grid map generated using *GMapping* [26] in the real environment. The results are similar, but the SLAM map has a slightly bent corridor. Here, a loop closure might have yielded more accurate results. Both maps can be used in conjunction with AMCL [27] as a localization method.

Beyond this comparison, the OSM offers more options to generate a grid: a) it can generate maps with different cell sizes, b) it can take the state of the environment (like open or closed doors) into account when generating the grid maps and, therefore, help to reason about sensor outputs, c) it does not require to manually ”polish” some areas[3] and d) it does not contain objects occluding walls.

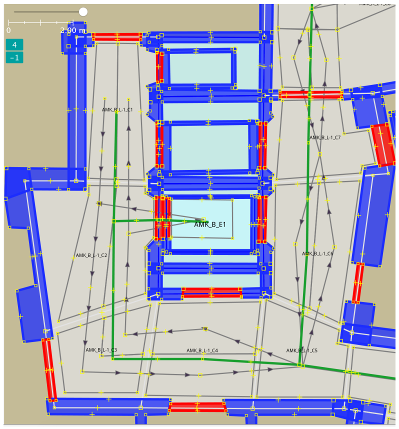
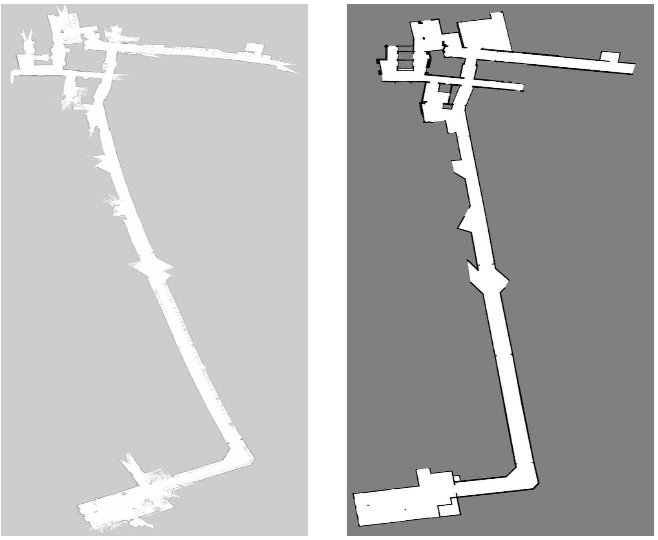


Fig. 12. Map created using proposed OSM model viewed in the JOSM [28] editor. Blue areas are wall geometries. Red ones are for doors. The green lines indicate the connections between corridors and rooms, while the directed arrows indicate connections between areas at a lower level of abstraction.

B. Way points generation

In the second experiment, we used the *area* features in the graph-based map for generating intermediate waypoints to a destination. Fig. 15 shows the difference between the paths planned using a global planner [27] with an occupancy grid map and the waypoints generated using navigation areas in the form of ”traffic lanes” from our graph-based map. The path between these waypoints is planned using local motion planners. The global grid planner returns a smooth and short path. The planner using the proposed map runs an A* graph search on the topology of the *areas* (cf. Section III). Fig. 14 shows an excerpt of such a graph. At the lower level, it forms a directed graph between connected areas. Such areas represent traffic lanes, junctions, waiting zones, or other domain specific concepts. In the presented case this allows to plan paths that will guide robots along the right side of corridors. This a) allows better integration with multi-robot applications and b) a more predictable behaviour for humans: the robots *behave like cars on streets*. At the higher level of the hierarchy, (undirected) connections between, e.g., rooms, corridors, and junctions are represented. This can be exploited for coarse path planning that gets refined in a second step.



(a) Occupancy grid map generated using *GMapping*

(b) Occupancy grid map generated using OSM

Fig. 13. Comparison between occupancy grid map generated using *GMapping* and the OSM based approach. Both maps are similar.

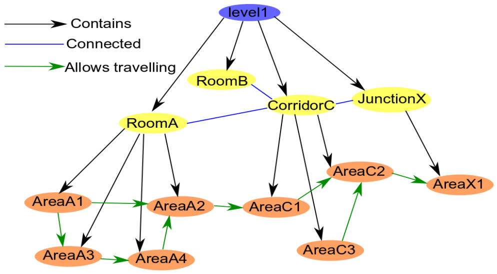
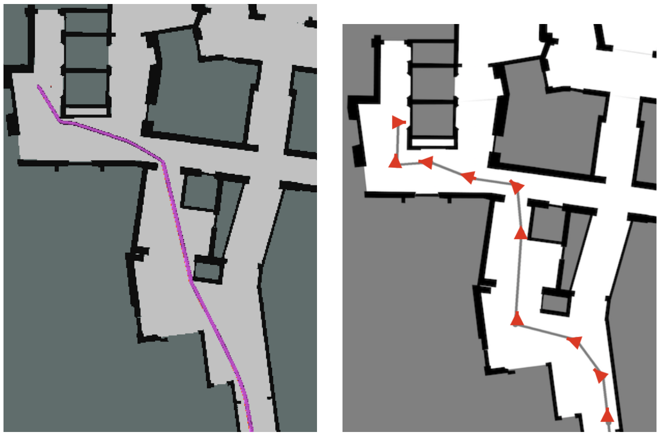


Fig. 14. This is an excerpt of a graph-based map showing the hierarchy in the areas that can be exploited for topological path planning.

C. Comparison to Simple Indoor Tagging model

In this work, we extended maps developed for humans for robotic applications. Thus we conducted another experiment to understand the number of details required for humans and robots. We mapped the same part of the building first using only the Simple Indoor Tagging (SIT) [2] model (used for creating indoor OSM for humans) and then with our proposed model. Table I provides a comparison of the number

[3] Generated maps can contain spurious *noise* in the free cells, when the laser scanner coverage was not good during the mapping process. Usually in those regions planners fail to generate paths, unless they are manually post-processed.



(a) Path planned only using geometric information in grid map.

(b) Way-points generated using geometric, topological and semantic information in OSM. Location of the way-points ensure that robot path sticks to the right side of corridors, according to the modelled semantics of ”traffic rules”.

Fig. 15. Comparison between path generated using grid maps and proposed semantic map.

of OSM elements in indoor OSM for robots and humans. There is an exponential rise in the number of OSM elements required to create maps in a robotic context. The additional flexibility and extended capabilities to generate grid maps for interfacing with existing software or to enable topological path planning, which takes semantic constraints like *traffic rules* into account, comes at a high prize in the form of drastically increased modelling efforts. Currently, these efforts are done manually and only supported by the tools offered by the OSM community. These tools were mostly developed for outdoor use cases and do not provide good support for indoor modelling. And while the (graph-based) data model of OSM allows the composition with domain-specific models, the tools lack support for some concepts like (visualizing) hierarchy. A improvement could come from any of the following three directions: a) improving the existing tools or adoption of better suited tool, e.g., from architecture, b) exploiting available digital building information (e.g., BIM [29]) to automatically generate semantic maps, c) (semi-)automatically building semantic maps from sensor information as suggested by several approaches mentioned in Section II.

	SIT	Proposed model
No. of nodes	35	543
No. of ways	9	225
No. of relations	0	289

TABLE I

Comparison of OSM data elements in indoor OSM for robots and humans

V. Conclusions and Future work

In this work, we present a graph-based, semantic mapping approach. OSM was chosen as a starting point for its implementation, since it is a) based on a graph data structure, b) it is successfully used for outdoor environment models and c) has tool support for manual model creation as well as a server infrastructure with tooling for querying information from the map (like osmium and overpass). Further, it is supported by a strong community. The infrastructure of OSM allows sharing maps with others. This has the potential that robots can share their maps with other robots. However, OSM also has shortcomings.

First, the process of manually creating the map is despite the available GIS tools tedious. In future, improving tool support, exploiting available digital building information (e.g. BIM [29]) and integration of bottom up methods for generating semantic maps into a process for (semi-)automatically generating such maps, will be a key factor for the success of the proposed map model. Second, OSM requires the use of georeferenced coordinates. This is rather counterintuitive for indoor robotic applications. Using vector tiles (represented with GeoJSON) might be a solution that is in line with developments in the OSM community. GeoJSON provides partial maps in a local coordinate system, which can be hooked on any external reference system.

The proposed extension (cf. Section III) allows to generate maps useful for localization based on *AMCL*. The generated output is similar to a map generated with *GMapping*. While this shows compatibility with existing systems, it does not exploit the full potential of the proposed models. Querying the graph model for specific features allows selective discovery of task-relevant features in the immediate environment together with plans (”affordances”) to help the robot’s own planner. E.g. one can query for specific perception features based on available sensors and the task like detecting a door frame for the task of moving through a door or just distances to the walls for moving along a hallway.

The introduced models already allow path planning, that takes semantic constraints like *traffic rules* into account. This enables a predictable motion behaviour for humans: *the robots move like cars on the streets*. The composability and hierarchy supported by graph-based models also allow seamless navigation over multiple maps (indoor or outdoor). We showed that the model is at least applicable in the domain of navigation. Our future work will address the impact of such a model on *motion control*: we plan to add further semantics to the local areas, to refine motion behaviour: e.g. slow down and indicate intention with turn signals in front of a junction or *no parking* areas close to fire doors. Investigations for such extensions for perception and localization as well as semantics for motion control (like the mentioned traffic rules) has already started.

Acknowledgment

This work was supported by the European Union’s Horizon 2020 projects ROPOD (grant agreement No 731848) and RobMoSys (grant agreement No 732410).

References

[1] W. Burgard and M. Hebert, *World Modeling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 853–869. [Online]. Available: https://doi.org/10.1007/978-3-540-30301-5_37

[2] "Simple indoor tagging - openstreetmap," http://wiki.openstreetmap.org/wiki/Simple_Indoor_Tagging, accessed: 2017-11-19.

[3] M. Hentschel and B. Wagner, "Autonomous robot navigation based on openstreetmap geodata," in *13th International IEEE Conference on Intelligent Transportation Systems*, Sept 2010, pp. 1645–1650.

[4] L. Marconi, C. Melchiorri, M. Beetz, D. Pangercic, R. Siegwart, S. Leutenegger, R. Carloni, S. Stramigioli, H. Bruyninckx, P. Doherty *et al.*, "The sherpa project: Smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments," in *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 1–4.

[5] S. Govindaraj, K. Chintamani, J. Gancet, P. Letier, B. van Lierde, Y. Nevatia, G. D. Cubber, D. Serrano, M. E. Palomares, J. Bedkowski, C. Armbrust, J. Sanchez, A. Coelho, and I. Orbe, "The icarus project - command, control and intelligence (c2i)," in *2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Oct 2013, pp. 1–4.

[6] J. Wietrzykowski, M. Nowicki, and A. Bondyra, "Exploring openstreetmap publicly available information for autonomous robot navigation," in *Progress in Automation, Robotics and Measuring Techniques*, R. Szewczyk, C. Zieliński, and M. Kaliczyńska, Eds. Cham: Springer International Publishing, 2015, pp. 309–318.

[7] P. Ruchti, B. Steder, M. Ruhnke, and W. Burgard, "Localization on openstreetmap data using a 3d laser scanner," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 5260–5265.

[8] O. Vysotska and C. Stachniss, "Exploiting building information from publicly available maps in graph-based slam," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 4511–4516.

[9] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, Oct 2008.

[10] B. Suger and W. Burgard, "Global outer-urban navigation with openstreetmap," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1417–1422, 2017.

[11] G. Floros, B. van der Zander, and B. Leibe, "Openstreetslam: Global vehicle localization using openstreetmaps," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 1054–1059.

[12] P. Fleischmann, T. Pfister, M. Oswald, and K. Berns, *Using OpenStreetMap for Autonomous Mobile Robot Navigation*. Cham: Springer International Publishing, 2017, pp. 883–895. [Online]. Available: https://doi.org/10.1007/978-3-319-48036-7_64

[21] D. Lang and D. Paulus, "Semantic maps for robotics," in *Proc. of the Workshop Workshop on AI Robotics at ICRA*, 2014.

[13] E. C. M. Pereira, D. A. Lima, and A. C. Victorino, "Autonomous vehicle global navigation approach associating sensor based control and digital maps," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, Dec 2014, pp. 2404–2409.

[14] "Openstationmaps," https://openstationmap.org, accessed: 2017-11-19.

[15] "Simple 3d buildings - openstreetmap," http://wiki.openstreetmap.org/wiki/Simple_3D_buildings, accessed: 2017-11-19.

[16] "F3db - full 3d building - openstreetmap," http://wiki.openstreetmap.org/wiki/F3DB, accessed: 2017-11-19.

[17] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems*, vol. 66, pp. 86 – 103, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889014003030

[18] I. Kostavelis, K. Charalampous, A. Gasteratos, and J. K. Tsotsos, "Robot navigation via spatial and temporal coherent semantic maps," *Engineering Applications of Artificial Intelligence*, vol. 48, pp. 173 – 187, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197615002596

[19] N. Blodow, L. C. Goron, Z. C. Marton, D. Pangercic, T. Rhr, M. Tenorth, and M. Beetz, "Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2011, pp. 4263–4270.

[20] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3515–3522.

[22] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.-A. Fernandez-Madrigal, and J. González, "Multi-hierarchical semantic maps for mobile robotics," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 2278–2283.

[23] H. Zender, O. M. Mozos, P. Jensfelt, G.-J. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 493–502, 2008.

[24] B. Lorenz, H. J. Ohlbach, and E.-P. Stoffel, "A hybrid spatial model for representing indoor environments," in *W2GIS*. Springer, 2006, pp. 102–112.

[25] "Openstreetmap wiki," https://wiki.openstreetmap.org/, accessed: 2017-12-08.

[26] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007. [Online]. Available: http://www2.informatik.uni-freiburg.de/∼stachnis/pdf/grisetti07tro.pdf

[27] "Ros navigation," http://wiki.ros.org/navigation, accessed: 2018-09-08.

[28] "Josm," https://josm.openstreetmap.de/, accessed: 2018-09-08.

[29] P. Vh, T. Heikkil, P. Kilpelinen, M. Jrviluoma, and E. Gambao, "Extending automation of building construction survey on potential sensor technologies and robotic applications," *Automation in Construction*, vol. 36, pp. 168 – 178, 2013.