On a hypergraph structuring semantic information for robots navigating and conducting their task in real-world, indoor environments

Sijs, Joris; Fletcher, James

DOI
10.1109/MMAR55195.2022.9874265

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the 26th International Conference on Methods and Models in Automation and Robotics, MMAR 2022

2022 26th International Conference on Methods and Models in Automation and Robotics (MMAR) | 978-1-6654-6858-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/MMAR55195.2022.9874265

On a hypergraph structuring semantic information for robots navigating and conducting their task in real-world, indoor environments

Joris Sijs[1] and James Fletcher[2]

***Abstract*— Robotic systems operating in the real world would benefit from a clear semantic model to understand their interactions with the real world. Such semantics are typically captured in an ontology. Unfortunately, existing world models in robotics focus on its navigation task. They adopt a hierarchical structure decomposing the environment from large spaces into small objects having a position, thereby limiting the robot's interactions as a "go-to-object" task. To allow a richer understanding of the real world this hierarchical structure should be replaced with an ontology, yet one that does not limit the real-time requirements of the robot when it is queried or updated with new observations extracted from sensors. Such an ontology is presented in this article. For now the ontology also focusses on the navigation aspect of robots, yet it is open to model other aspects of the real world as well. Experiments show that multiple environments are successfully modelled supporting the robot to go from one room to another to search for humans.**

I. INTRODUCTION

Autonomous, robotic systems are expected to operate in uncertain environments, and execute various tasks at various locations. To do so, they plan their route and the execution of their task based on information retrieved from the environment. In the rescue operation addressed in this article the robot needs to move between rooms to search for victims. In such operations there is typically no time to first explore the building and then search for humans, though one can expect information regarding its floor map.

The traditional approach to acquire a geometric floor map is known as simultaneous localization and mapping (SLAM). Among the different SLAM techniques there are approaches computing object maps, see [1], [2], and approaches producing semantic annotations on a point cloud, on a mesh or on voxels such as [3], [4]. Other methods, e.g., the ones presented in [5], [6], turn such geometric maps into semantically meaningfull places (halls and rooms). Recently, SLAM has been extended to produce maps that include objects detected online, such as humans, chairs and coffee cups. Scene graphs are such an extension, which are computer graphics models to describe, manipulate, and render complex scenes [7], [8]. State of the art scene graphs, such as [9], [10], adopt class hierarchies to structure the semantic information of spaces and objects. These class hierarchies merely specify a spatial taxonomy to objects in the scene, for example, which classes are spatially a subset of others. Scene graphs support the robot to understand typical navigation tasks as a Go-To. However, why a robot should go to a space or object cannot be modeled with scene graphs, since their hierarchical structure is limited to a spatial understanding of the world. For that, this hierarchical model should be replaced with an ontology, yet ontologies are known to be complex and cumbersome making real-time aspects of the robot a challenge. Nonetheles, the benefits of an ontology is the advanced logical reasoning to update its semantic information, e.g., with routes between rooms via doors so that path planning can be done locally per room.

To study these drawbacks and benefits of an ontology and its inference rules this article presents an online knowledge graph that captures geometric information in combination with semantic -topological- information. Our knowledge graph, or knowledge base, structures aspects of a robot relevant for navigation in a building. The knowledge structure itself is based on a hypergraph instead of RDF triples, as in our experience real-world phenomena are more naturally expressible in a hypergraph. The knowledge base is initialized with prior information about the floor map, i.e., rooms, walls and doors, and victims to search for (see Figure 1). Once initialized it is used by the robot to plan routes and paths to search for victims in the different rooms of the building. Important aspects are whether the robot is able to plan and execute the operation in a limited amount of time, and whether the structure of the knowledge base is tractable.

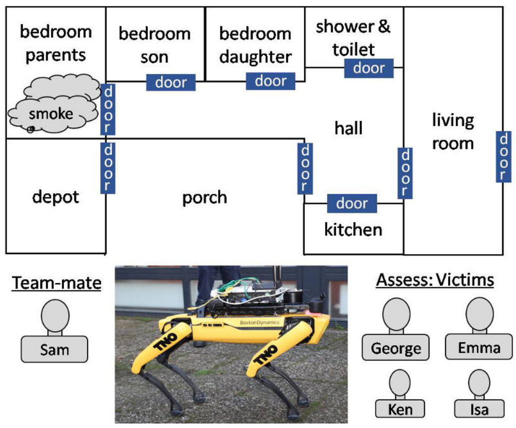


Fig. 1: Knowledge that is relevant to the operation, such as a floor map, victims and environmental conditions. This knowledge will be implemented on an actual robotic system: the Spot of Boston Dynamics extended with a camera, microphone, speaker and an embedded PC.

The outline of this article is that Section II will cover rel-

[1]Joris Sijs is with Delft Univeristy of Technology, Mekelweg 2, Delft, The Netherlands j.sijs@tudelft.nl[2] James Fletcher is with Vaticle, 47-50 Margaret Street, London, UK james@vaticle.com

evant literature on ontologies for navigation and summarize the aim of the knowledge base. Section III then continues with the developed ontology, which has a general applicability for robots that need to conduct tasks at various locations and in different operational conditions. The section will also cover some illustrative design patterns while presenting the ontology. Some insights on the actual implementation of the knowledge base is presented in Section IV. Then, Section V discusses experimental results of the implementation, followed by Section VI on conclusions and future work.

II. BACKGROUND AND CONTRIBUTION

The introduction of ontologies in robotics became apperant with the standarization of a core ontology on robotics and automation (CORA) in [11]. This ontology is directly connected to SUMO, one of the largest upper-level ontologies that is publicly available and largely covered in [12]. Interesting for our use-case is that developments in the field of "building information modelling" resulted in the open standard called IFC (Industry Foundation Classes), which is data structure to specify a building with all its objects and components (plant, chair, HVAC, pipes, and so on). This structure, or ontological model, has extenstions to robotic navigation, e.g., [13], [14], yet the complexity of the model render online updates by the robot infeasible. Still, the (tractable) navigation ontology developed in this article will benefit from the structure in IFC models so that parsers between the two may be developped.

Long before the introduction of CORA, IFC, and ontologies in the robotics community, the importance of world modeling and map representations has been central in the robotics and AI community. Hierarchical structures that classify spatial and semantic information was already proposed in [15], and inspired others to create their hierarchies to represent concepts in the world [16]. Most interesting are methods on scene graphs that also adopt this hierarchical representation, although they have not yet taken the step towards an ontological representation, i.e., a representation of not only classes but also of relations and constraints. The combination of scene graphs with ontologies is, however, of extreme relevance for *autonomous* robots: scene graphs are limited to spatial aspects of the objects from sensor observations, while ontologies create a richer understanding of the situation. For example, that doors are not only passages between rooms for the robot -when open- but also for smoke and noise, or that at night victims are more likley to be found in a bedroom rather than the living room or kitchen.

Therefore, our first contribution is an actual, robotic knowledge base in which new (factual) information from sensor-data can be added online according to an ontology, while extended (semantic) information may be derived from that. It is thus possible to integrate our knowledge base with state-of-the-art scene graph methods such as Kimera [10].

Our second contribution, as argued next, is that the ontology is structured as a hypergraph instead of a plain graph, thereby resulting in a more tractable and accessible knowledge structure in which inference is not so cumbersome as in plain graphs. To argue our second contribution, let us review some of the existing ontologies in robotics. In [17] an ontology was developed for planning and navigation in indoor environments, while [18] presented a similar planning ontology for underwater inspection. Studies on ontologies to structure knowledge about the environment are also available. For example in [19] for household robots, or in [20] for intelligent vehicles. Typically, existing methods are not able to include new information, i.e., information derived from sensor measurements. The main reason is that their implementation as a plain graph enforced developpers to create large and complex structures within the knowledge base that are too sensitive to be updated automatically online. Note that this statement does not apply for knowledge bases in robotic systems that adopt a hierarchical structure rather than an ontology. One approach does create an ontological knowledge base for robots, which is KnowRob, first introduced in [21] and later extended to an updated version in [22]. KnowRob offers a solution to semantics in the kowledge structure of robots operating in the household domain. Information of the environment from sensor observations is further stored in the knowledge base of the robot, followed by local knowledge updates based on the robot's contextual knowledge. However, since KnowRob adopts a plain graph, its knowledge structure is quite complex and thus sensitive to programming errors from expert users. Nonetheless, KnowRob has been used on actual robots, albeit for conducting a single task in a very structured environment.

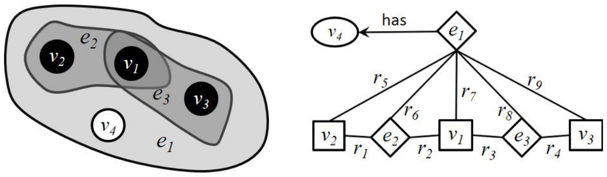
In future, robotic systems will operate more and more in open and complex environments. Designing a knowledge structure, such as an ontology, capturing such complexity as a plain graph with edges linking two vertices often requires a complex administration to maintain the proper structure [23]. This limitation can be removed when the structure of an ontology assumes a hypergraph design. This hypergraph model is the main reason that a new ontology, or knowledge structure, should be developed. Another reason, since robots may conduct different tasks at different locations, is that the ontology should allow *non-spatial* -semantic- information relevant for the robot's task and bring that in relation to its *spatial* -semantic- information relevant for navigation.

III. A HYPERGRAPH ONTOLOGY

A. The concept of a hypergraph

In a hypergraph, such as the one depicted in Figure 2a, a hyper-edge as e_1 can join multiple vertices as v_1, v_2, v_3 and v_4, while other edges between these vertices, such as e_2 and e_3, can also be part of the "overarching" edge e_1. For clarity, a hypergraph may be drawn as a plain graph (Figure 2b) by treating the hyper-edge as another type of vertices in the plain graph and add an argument r_i to each of the plain edges.

This article adopts the plain graph representation of a hypergraph when developping its ontology. The ontology is derived from three concepts (solid underline) and two links (dotted underline), where a concept is either a hyper-vertice or hyper-edge and a link is a plain edge:



(a) original representation (b) plain graph representation

Fig. 2: An illustrative example of a hypergraph in its original representation (a) and in a plain graph representation (b).

- Entity: A subclass of Concept that can consist independent from any other concept (v_1, v_2, v_3 in Figure 2);
- Relation: A subclass of Concept on how multiple entities and/or relations are joined (e_1, e_2, e_3 in Figure 2);
- Attribute: A subclass of Concept *with a value* modelling a property of an entity or relation (v_4 in Figure 2);
- Role: A subclass of Link expressing the role that the entity or relation plays in a relation (r_i in Figure 2);
- has: A subclass of Link pointing to an attribute indicating it is owned by some entity, relation, or attribute (*has* arrow in Figure 2b).

With the concepts Entity, Relation, Attribute, and the links has, Role, the ontology specifies possible links between two concepts with the notation $\langle Concept, Concept \rangle : Link$. Links between concepts are inherited by subclasses. When the link is a Role the ontology autogenerates both pointing options, while a has link is unidirectional and pointing to an attribute:

$$\begin{array}{lcl}
\langle Entity, Entity \rangle : Role & ; & \langle Entity, Attribute \rangle : has \; ; \\
\langle Relation, Relation \rangle : Role & ; & \langle Relation, Attribute \rangle : has \; ; \\
\langle Attribute, Attribute \rangle : Role & ; & \langle Attribute, Attribute \rangle : has \; ; \\
\langle Entity, Relation \rangle : Role & \Leftrightarrow & \langle Relation, Entity \rangle : Role \; ; \\
\langle Entity, Attribute \rangle : Role & \Leftrightarrow & \langle Attribute, Entity \rangle : Role \; ; \\
\langle Relation, Attribute \rangle : Role & \Leftrightarrow & \langle Attribute, Relation \rangle : Role .
\end{array}$$

B. Hierarchy of classes

A first step in the construction of an ontology are the class hierarchies of its concepts. Given that our hypergraph has three concepts, i.e., entities, attributes and relations (with roles linked to relations), three hierarchies are developed.

The first class hierarchy, depicted in Figure 3, is the hierarchy of the Entity base class that is firstly divided into Abstract and Physical, followed by other subclasses, such as mathematical, or artifacts, spaces and intelligent beings. For brevity, exact definitions per subclass or omitted and we rely, for now, on a common interpetation by the reader. Extensions to this hierarchy are possible, such as cats and dogs as a subclass of animal. Yet, most importantly, the subclass hierarchy of Artifact has direct relations with IFC data model and may be extended as such.

The second class hierarchy, depicted in Figure 4, is that of the Attribute base class indicating ownership, which at some point in the hiearchy should be given a format (Boolean, double, string). Common subclasses of attributes are Name and Height. The combination of the other two subclasses Euler Rotation and Linear Translation is needed for defining the geometric position of any physical concept with respect to

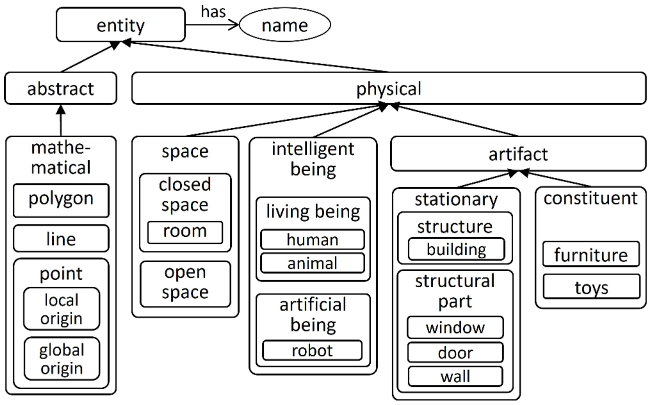


Fig. 3: The Entity class hierarchy (rectangles) giving a type specification. Each entity has a name (of type attribute). Further, an open arrow specifies a has-relation, while a closed arrow is a subclass-relation (or type-specification).

an origin (local or global). A geometric position is modelled as a Normal distribution, which is specified by a mean value (the actual postion defined with a linear translation in x, i.e., lin-x, a rotation around x, i.e., rot-x, etc.) and a sigma value (the actual uncertainty w.r.t. the mean value defined by the uncertainty on the linear translation in x, i.e., dlin-x, etc.).

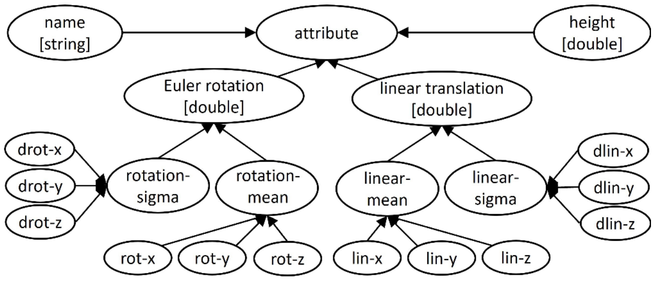


Fig. 4: The Attribute class hierarchy (ellipses) giving a type specification with a [format] per subclass. If not specified, a subclass inherits (with recursion) the format of its superclass.

The third and final class hierarchy, depicted in Figure 5, is that of the Relation base class, which also depicts the roles that are defined in each relation. Relations are used to link multiple entities, relations and/or attributes. Therefore, a class (or concept) is a relation if it cannot exists solely by itself, otherwise it is an entity, and if it cannot be owned, otherwise it is an attribute. The relations in our ontology either specify a spatial relation, such as Route, Location and Composing, or some specific mathematical relation that is either a definition, such as Connector or Collection, or how math may model the real world, such as Enclosing, Taking Form and Positioning. Note that some, e.g., location and position, are often modeled in other ontologies as attributes yet here as relations. Our argument is that an object is located in some space, hence it is a relation, and that an object has a position w.r.t. some origin -globally or locally of some space- and is thus also a relation yet with the typical position attributes. The next section on design patterns gives more insight as to why a subclass is a relation, entity or attribute.

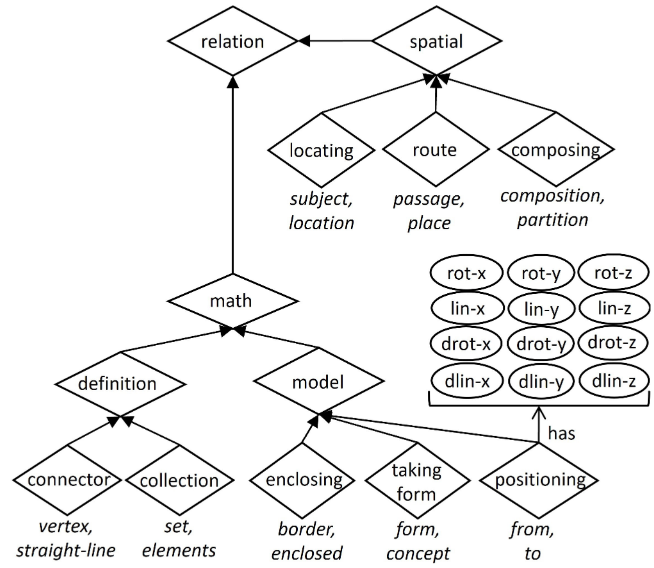


Fig. 5: The Relation class hierarchy (diamonds) giving a type specification and its roles (specified in italic text). If applicable, a subclass inherits the roles of its superclass.

C. Patterns between classes

The developed ontology continues with so called design patterns of knowledge, which are parts of the ontology specifying how a logic set of concepts (entities, attributes and relations) are joined through roles they may play in a relation. Examples of how these knowledge patterns come together in instantiations are presented in Section IV.

The first design pattern is with regards to the topological knowledge in a floor map of buildings, indicating a composition of spaces, such as rooms, and where doors result in routes between rooms, as depicted in Figure 6. The pattern states that Building may play the role of Composition to specify that it is Composing of partitions as Room. In its turn, a Room, or Closed Space, may also play the role of Composition to specify that it is Composing of partitions that may be played by Structural Part, such as Wall, Window and Door. A last specification in this pattern is the relation Route modelling knowledge that Door, not Wall nor Window, may play the role of Passage between places like Room.

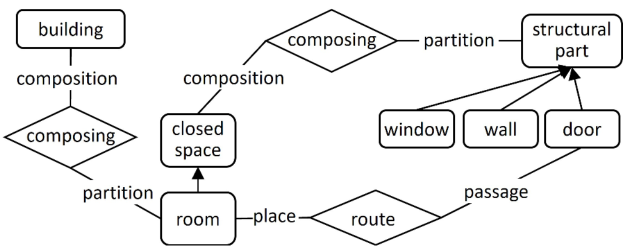


Fig. 6: The knowledge pattern specifying how buildings are composed of rooms and how rooms are composed of structural parts. A door may define a route between rooms.

The second design pattern concerns the location of beings and artifacts in rooms, as well as their geometrical position with respect to some origin (see Figure 7). It also concerns how closed spaces, such as a room, are mathematically modeled as a polygon, while a line (e.g., of a polygon) might take the form of a structural part in the real physical world. Now, the pattern defines that Intelligent Being and Artifact, in their role of Subject, may be Locating at a Location that is a Space. Here, it is important to recall from Figure 6 that a Room is a subclass of Closed Space, implying that intelligent beings and artifacts may be located in rooms. The pattern further specifies that Polygon may act as the Border that is enclosing a Closed Space, where the Closed Space is the one that is Enclosed. Polygon may also act as a Set that is defined by a Collection of Element(s) each played by Line. When a Line acts as a Straight Line, then it is defined by two Vertices each played by Point in the relation Connector. Further, Line may act as a Concept that is Taking Form in the real physical world, in this case the Form of a Structural Part. The last specification of this pattern is the relation of Positioning, for which it is important to recall (Figure 5) that it has attributes to define its geometric information. This Positioning models a positioning-vector From a Global Origin or a Local Origin To either a Point, an Artifact, or an Intelligent Being. In addition, this Positioning also models a positioning-vector From a Global Origin To a Local Origin.

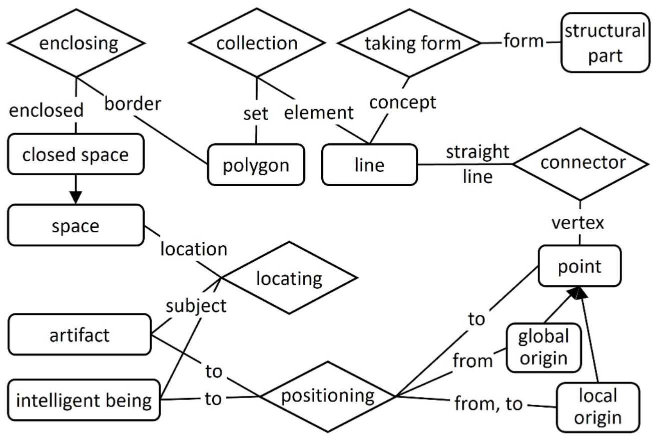


Fig. 7: The knowledge pattern specifying how beings and artifacts are located in a space. The pattern also specifies how a closed space, like a room, is defined by a polygon and how the lines in that polygon relate to the structural parts that compose a closed space, while a line may be modeled by (two) points. The position of points, artifacts and beings is defined with a positioning relation that starts from either a global or a local origin, while a local origin is also defined with a positioning vector starting from a global origin.

IV. THE ONLINE KNOWLEDGE BASE

The ontology introduced in Section III has been implemented as a hypergraph knowledge base using a tool called TypeDB. The knowledge base is initialized with instances of prior information assumed to be available before the operation (floor map, victims to search for, etc.). These instances represent the actual information the robot has about its environment. This section will discuss two examples of geometric and semantic information of the floor map and how they are captured by the knowledge base. Our dicussion assumes a path planning ability in which the robot plans a *semantic route* from one room to any other room in the building, while planning a *geometric path* to go from any point in a room to a point just at the other end of some door that is part of that room. Planning of possible routes between rooms may be done directly after the knowledge base is initialized, thereby saving online computation time.

Our first example concerns actual information about a single room, in this case the depot of the floor map of Figure 1. The instances in the knowledge base related to this room are depicted in Figure 8, while the mathematical illustration of this room as a polygon is shown in its top-right corner. There are instances that define a room, named "depot", which is composed of a wall and of a door named "depot-door". Creating these instances is done with variables as a to define concepts and $a.role$ to define edges.

$$
\begin{aligned}
&insert(r \in Room),\\
&\quad r.hasName = \{\text{“depot”}\};\\
&insert(w \in Wall);\\
&insert(d \in Door),\\
&\quad d.hasName = \{\text{“depot-door”}\};\\
&insert(c \in Composing),\\
&\quad c.composition = \{r\}, c.part = \{w, d\}.
\end{aligned}
$$

The instances further specify that the room is enclosed by a polygon, which is a collection of six lines where each line is a connector between two points (for clarity, not all lines and points are depicted). Most lines represent a wall in the real world (taking form). One line represents the depot-door. The starting point of the polygon is the local origin of the room while all other points of this room have a positioning relation to this local origin specifying the position of that point in 2D from this local origin (lin-x, lin-y). For clarity of the figure not all positioning relations are illustrated, just like the position from the local to a global origin is not illustrated. The advantage of this local origin is that it specifies the position of the robot, of the door and of any other object in that room, thereby scoping the navigation of the robot to planning a geometric path within the polygon of that room. When exiting the room a path is planned to go just accross the polygon's line that models the exiting door.

Our second example, depicted in Figure 9, focusses on the topological information of the rooms depot, patio and hall. It shows that each of those rooms is composed of walls and doors. Note that the door of the depot is part of two compositions, i.e., of the depot and of the patio. The same holds for the front-door being part of both the patio and the hall. Now, a logical rule may infer that when the same door is part of two rooms, there is a route between those rooms: the depot and the patio are linked to a route passing the door of the depot, and there is a route between the patio and the hall via the front-door. Furthermore, another logical rule

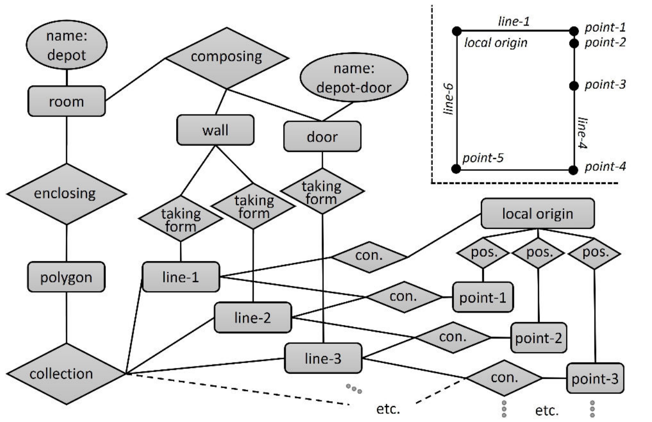


Fig. 8: Instances of the knowledge base to structure all knowledge (semantic and geometric information) about the room called depot, i.e., the structural parts is it composed of, the polygon for a mathemtical represantation of the room, and the geometry of this polygon as points to a local origin.

may infer that when there is a route between rooms A and B and between rooms B and C, there is also a route between rooms A and C: the depot and the hall are linked to a route passing the depot door and the front-door. Therefore, when the robot wants to go to any other room in the building it may query its knowledge base upon an existing (semantic) route between its current room and the destination room. If such a route exists, than it will recieve a list of all doors that it needs to pass. It may than plan a geometric path to exit via the first door in the list, after which the robot will be in a new room in which it may plan a geometric path to exit via the next door on the list, and so on.

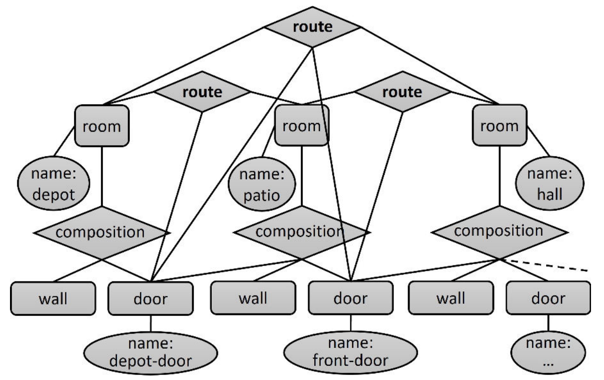


Fig. 9: Instances of the knowledge base to structure topological information about rooms and their composition. Possible routes between rooms can be logically inferred (bold lines).

V. DISUCSSION AND EXPERIMENTAL RESULTS

The previous example on how geometric information is combined with semantic information, and how it may be used for (semantic) route planning that scopes (geometric) path planning, is just one example of how our ontology and knowledge base is of use to the robot. This knowledge base has been implemented on an actual robot in a complete ROS2 setup to evaluate its practical feasibility. ROS2-modules process actual sensor observations, for example to acquire the robot's current location and position every second, and stores their results in the knowledge base at the same rate. Other, ROS2-modules then query the knowledge base when planning the operation of the robot. Details of the implementation may be found here https://youtu.be/suvSvXYec74. In case the knowledge base is not able to keep track of storing the location and position, then path planning will produce incorrect paths. An actual experiment in which a robot passes doors to go from one room to another, viewed in this online video https://youtu.be/kzVILa6buO4, shows that timing of these updates is done in real-time. A more extended experiment in which the robot planned to go from room to room and at its destination room approached a human may be viewed here https://youtu.be/6m1-bsjVHi0, while a similar operation in a completely different building may be viewed here https://youtu.be/uP-B7rt7UWA.

The fact that a floor map of two different buildings could be implemented within the knowledge base in just a few hours shows that our ontology is simple and accessible. Further, the succes of the experiments also shows that the ontology is sound, as route planning is without errors, and that it did not limit the real-time aspect of the robot. Also, parsing software has been created that converts the floor map of the knowledge base (of a third different building) in an IFC model with walls and doors (see Figure 10).

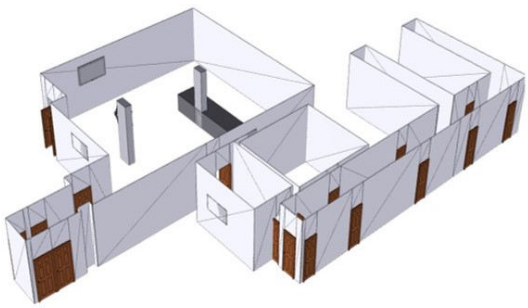


Fig. 10: A floor map in the knowledge base as an IFC model.

VI. CONCLUSIONS

Implementing a knowledge base to understand the real, physical world, while modelling such knowledge with classical ontologies in plain graphs, will require work-arounds rendering the ontology complex and cumbersome. This article studied the use of a hypergraph as the basis of an ontology, thereby making the knowledge base tractable. Also demonstrated was the compartmentalisation of the knowledge base into distinct, but connected,"knowledge patterns". The design of the ontology focussed on the navigation task of a robot in which it needs to combine both semantic information about rooms and doors with geometric information, for which rooms are modeled as polygons. Experimental results revealed that the knowledge base was able to model the real, physical world in a sensible way without limiting the real-time aspects of our task and route planner.

REFERENCES

[1] R. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 512–519.

[2] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam," *IEEE Robotics and Automation Letters*, vol. 24, pp. 1–8, 2018.

[3] K. Eckenhoff, Y. Yang, P. Geneva, and G. Huang, "Tightly-coupled visual-inertial localization and 3d rigid-body target tracking," *IEEE Robotics and Automation Letters*, vol. 2, pp. 1541–1548, 2018.

[4] J. Wald, K. Tateno, J. Sturm, N. Navab, and F. Tombari, "Real-time fully incremental scene understanding on mobile platforms," *IEEE Robotics and Automation Letters*, vol. 3, pp. 3402–3409, 2018.

[5] J. L. Blanco, J. González, and J.-A. Fernández-Madrigal, "Subjective local maps for hybrid metric-topological slam," *Robotics and Autonomous Systems*, vol. 57, pp. 64–74, 2009.

[6] K. Zheng, A. Pronobis, and R. P. N. Rao, "Learning graph-structured sum-product networks for probabilistic semantic maps," in *In Proc. of the 32nd AAAI Conf. on Artificial Intelligence (AAAI)*, 2018.

[7] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, "Scene graph generation from objects, phrases and region captions," in *In Proc. of the Int. Conf. on Computer Vision (ICCV)*, 2017.

[8] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. (2016) Visual genome: Connecting language and vision using crowdsourced dense image annotations. arXiv: 1602.07332.

[9] U.-H. Kim, J.-M. Park, T.-j. Song, and J.-H. Kim, "3d scene graph: A sparse and semantic representation of physical environments for intelligent agents," *IEEE Transactions on Cybernetics*, vol. 50, no. 12, pp. 4921–4933, 2020.

[10] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone. (2021) "kimera: from slam to spatial perception with 3d dynamic scene graphs. arXiv preprint:2101.06894.

[11] E. Prestes, et.al., "Towards a core ontology for robotics and automation," *Robotics and Autonomous Systems*, vol. 61, pp. 1193–1204, 11 2013.

[12] A. Pease, *Ontology: A practical guide*. Articul. Software Press, 2013.

[13] P. Pauwels, T. Krijnen, W. Terkaj, and J. Beetz, "Enhancing the ifcowl ontology with an alternative representation for geometric data," *Automation in Construction*, vol. 80, pp. 77–94, 2017.

[14] E. González, J. D. Piñeiro, J. Toledo, R. Arnay, and L. Acosta, "An approach based on the ifcowl ontology to support indoor navigation," *Egyptian Informatics Journal*, vol. 22, no. 1, pp. 1–13, 2021.

[15] B. Kuipers, "The spatial semantic hierarchy," *Artificial Intelligence*, vol. 119, pp. 191–233, 2000.

[16] S. Thrun, *Exploring artificial intelligence in the new millennium*. Morgan Kaufman Inc., 2003, Chapter 1: Robotic mapping: a survey.

[17] G. Kim, I. Suh, and H. Suh, "Ontology-based unified robot knowledge for service robots in indoor environments," *IEEE Trans. on Systems Man and Cybernetics - Part A*, vol. 41, pp. 492–509, 2011.

[18] M. Cashmore, M. Fox, D. Long, D. Magazzeni, B. Ridder, and F. Maurelli, "Dynamically extending planning models using an ontology," in *Proc. of the 2nd ICAPS Workshop on Planning and Robotics (PlanRob)*, 2015.

[19] S. Joo, S. Manzoor, Y. Goncalves-Rocha, S. Bae, K. Lee, T. Kuc, and M. Kim, "Autonomous navigation framework for intelligent robots based on a semantic environment modeling," *Applied Sciences*, vol. 10, pp. 3219–3239, 2020.

[20] K. Czarnecki, "Operational world model ontology for automated driving systems – part 1: Road structure & – part 2: Road users, animals, other obstacles, and environmental conditions," Waterloo Intelligent Systems Engineering Lab, Tech. Rep., 2018.

[21] M. Tenorth, D. Jain, and M. Beetz, "Knowledge processing for cognitive robots," *KI-Kunstliche Intel.*, vol. 24, pp. 233–340, 2010.

[22] M. Beetz, D. Beßler, A. Haidu, M. Pomarlan, A. Bozcuoglu, and G. Bartels, "KnowRob 2.0: 2nd generation knowledge processing framework for cognition-enabled robotic agents," in *Proc. of the 2018 IEEE Int. Conf. on Robotics and Automation*, 2018, pp. 512–519.

[23] E. Scioni, N. Hübel, S. Blumenthal, A. Shakhimardanov, M. Klotzbücher, H. Garcia, and H. Bruyninckx, "Hierarchical hypergraphs for knowledge-centric robot systems: a composable structural meta model and its domain specific language,", *Journal of Software Engineering for Robotics*, vol. 7, no. 1, pp. 55–74, 2016.