# On Planning in a Knowledge-Hypergraph

Joris SIJS [a,1], Willeke VAN VUGHT [a] Jeroen VOOGD [a]

[a] *TNO, The Netherlands*

**Abstract.** Autonomous robots that operate in indoor, uncertain environments move around from one location to the next to fulfill a set of ordered task. To do so, they need a floor plan of the building to plan their route. In case the building is unfamiliar to a robot it can employ online mapping techniques, though these techniques are resource intensive and time consuming. In addition, processing such geometric maps in an automated planner to determine a feasible route is resource demanding as well. The solution proposed in this article is an ontology that combines knowledge of floor plans and actual information of a building with concepts in automated planning, so that robots become more efficient in route planning. Moreover, the ontology is implemented as a hypergraph to benefit from its advances in creating elegant inference-rules, e.g., to infer route-alternatives while preparing the operation.

**Keywords.** task, route planning, knowledge representation, hypergraph

## 1. Introduction

Autonomous, robotic systems are expected to operate in uncertain environments for executing various tasks at various locations. In case there is sufficient freedom in task order they need to plan their route and their next task simultaneously. An example is a search and rescue operation in a building, where the robot needs to move between rooms to search and assist victims. Such operations are typically conducted in environments that are new to the robot. This does not necessarily mean that the robot is completely ignorant and has to conduct resource-demanding techniques as Simultaneous Localization and Mapping (SLAM [1,2]) to get a geometric, subsymbolic map of the environment. Instead, information as printed or digital maps is often available characterizing the robot's environment as a floor plan containing geometric and ontological information (subsymbolic and symbolic). A popular, yet resource-demanding route planner, able to exploit the geometric and some of the map's ontological information as *wall*, adopts hierarchical reinforcement learning (HRL [3]). It uses the map to create a digital copy of the real world in which a simulated robot will explore feasible route-alternatives. A more efficient approach, i.e., not requiring an accurate copy of the world, would be a planning procedure that is ontology-driven. By introducing concepts of a floor plan and of automated planning in the ontological knowledge base of the system, and combine those with actual information of the building, it can use inference rules to prepare route-alternatives before an online, automated planner is to determine the actual route.

---

This article discusses such an ontology, represented as a hypergraph, that combines concepts that typically exist in a floor plan with some main concepts used in automated planning. Given this ontology, the robot can deduce via symbolic reasoning how to go from one place on the map to another before the operation starts, without the need for data-driven approaches as SLAM to first create a map that is then used by HRL or Monte-Carlo Tree Search [4] to search for feasible routes. The main difference with existing ontologies on planning and navigation, for example the ones described in [5,6,7], is that these have representations as directed graphs to conform to a formal language as OWL or an informal language as RDF, whereas our ontology has a hypergraph representation to suit the less familiar GRAQL [8]. The reason for this decision is that, in our experience, real-world phenomena are often more naturally expressible in a hypergraph, and as such, they could improve the interoperability between the robot and its human team-mates.

The next Section 2 provides some background on important concepts found in floor plans and automated planning, followed by Section 3 introducing GRAQL as an approach to represent an ontology in a knowledge-hypergraph. Section 4 presents the proposed ontology for route-planning in an indoor environment and how to apply inference to automatically determine route-alternatives in its knowledge-hypergraph, followed by some initial conclusion and future work in Section 5.

## 2. Background of concepts

The ontology developed will not always be as thorough as some of the existing work, as it is intended to explore the benefits of a hypergraph rather than being 100% complete.

### 2.1. Concepts in a floor plan

Inspired by the work of [9], the map of a building can be viewed from three perspectives: physical, functional and geometrical. In its physical view a map presents how the different elements of the building are physically realized. For example, whether they are a *wall*, *door*, *window*, or *stairs*. In its functional view a map depicts the functions of the different *rooms*, such as a *hall*, *kitchen* and *living*. In its geometric view a map is read as *points* and *lines* of a *polygon* characterizing the size and the boundary of each *room*. A robot that moves from one location to the next should have an integrated understanding of all three views. An integration of the three will be discussed in the ontology of Section 4, while their hierarchy is introduced here, in Figure 1.
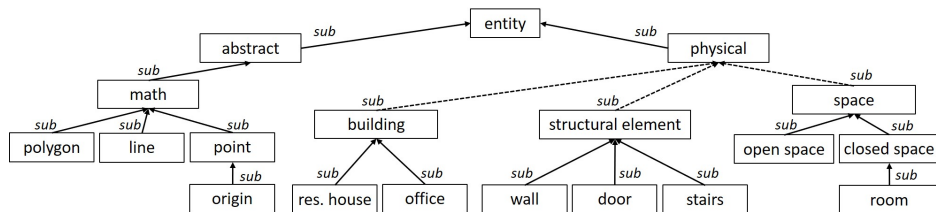


**Figure 1.** The hierarchy of concepts, i.e., sub-relations, that are present in a floor plan of a building or a house, partly inspired by SUMO [10]. The dotted arrow is to denote that there is exists a hierarchical relation between the two concepts via a series of other (not illustrated) concepts for the sake of excess space.

## 2.2. Concepts in automated planning

The research domain of automated planning has produced various modelling solutions, such as the Planning Domain Definition Language (PDDL [11]), Hierarchical Task Network (HTN [12]) and Markov Decision Process (MDP [4,5]). Each model has its own principles to describe a planning problem, yet there are many similarities between them. For example, all of them characterize the context and situation of the robot in a so-called state, while the goal is either defined as a particular state or as a utility indicating how preferable each conceivable state is. Further, all models define under which conditions of the state an action, or primitive task (synonym of action in HTN), can be planned. Each primitive task (a.k.a. action) starts a direct operation that is executed by the system taking it from one state to a next one. Often, the modelling solution allows the expert to create a hierarchy on what is planned, in which case it starts at the top level from a so-called compound task (or abstract action) that is further partitioned into either, more detailed compound tasks, or the lowest level of primitive tasks.

Here, the above-mentioned concepts are applied to a robot that plans its route in a building from its current location, called *actual location*, to a *goal* or destination, in this case its *final location*. The *state* that is used by the planner is the *actual location*. It is assumed that the robot has one or more *operators*, which is a designated set of system components via which the system is able to execute a *primitive task* independently. An *operator* that the robot can execute is *exit via door*, which has a matching *primitive task* of *goto adjacent room* that is conditioned on the fact that there exists a door connecting the room the robot is currently in with the adjacent room. Hence, the planner requires particular *contextual conditions*, or in this case *building conditions*, which are the *room connections* in the building. Then, a *composite task* would be to go from one room in the building to another room via a series of intermediate rooms, i.e., *goto room*, which is conditioned on the fact that there is a series of connected rooms from the robot's *actual location* to its *final location*. A hierarchical illustration of these *planning elements* is depicted in Figure 2, where it should be noted that *final location* and *actual location* are not a sub of *goal* and *state*, but they are (for now undefined) concepts merely used by the planner as either *goal* or *state*. In a same way *room connection* is used as a condition.
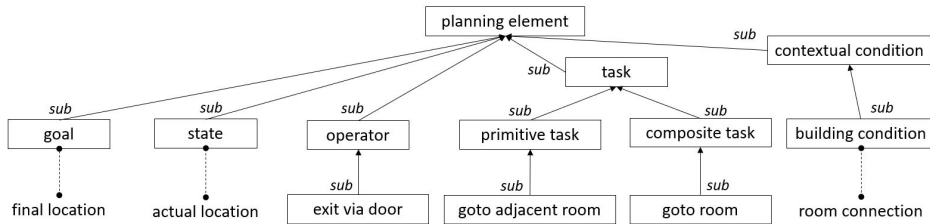


**Figure 2.** A hierarchy of planning elements found in modelling solutions as PDDL, HTN, MDP and HRL, with sub-relations a for a robot that can plan a route from its actual location to a final location.

The background presented in this chapter introduced some important entities that are used in a floor plan and in automated planning, but, apart from the sub-relations, it did not discussed the relations between them. These relations are introduced with the proposed ontology in Section 4, after some background on GRAQL has been given.

## 3. Background on GRAQL

GRAQL is a query language to create, use and maintain an ontological knowledge representation within the GRAKN hypergraph database [8]. GRAQL can be used to apply rules that act on the data as expert knowledge, or to retrieve knowledge as sets of data via queries. GRAQL defines a knowledge-hypergraph in two constructs: the "schema" specifying the ontology and the "data" specifying the instantiations. The schema consists of 4 hierarchical trees each starting from one the four main concepts in GRAQL:

- Entity: A concept that consist independent from any other concept in the domain;
- Attribute: A property of a concept in a domain;
- Relation: A description of how two concepts are related and which role the concepts play in this relation;
- Rule: A description of a pattern applied to the dataset in the when-then format.

The data then defines the instantiations of these concepts. A tool called GRAKN will ensure that only the data that is in accordance with the schema is inserted, and that the rules are automatically called-upon when the knowledge-hypergraph is being queried.

In a plain knowledge-graph, each entity and attribute is a node in the graph and each relation is an edge between two nodes. In the knowledge-hypergraph a relation itself is a node linked to multiple other nodes, as entities, attributes and relations, via edges each specified by a role. The benefit of a hypergraph is that a relation is another type of a concept where multiple ($> 2$) other concepts can play a role in. In addition, hypergraphs allow to define a hierarchy in relations and in their roles. For example, one could specify a *spatial-connection*-relation, having the roles *place* and *connector*, that relates two or more *spaces* (in the role *place*) via a *structural element* (in the role *connector*). In case of a floor plan, one could then specify a sub of this relation as a *room-connection*-relation having the roles *place*, *door-connector* and *stairs-connector*, where the latter two are a sub of the role *connector*, for example to connect two *rooms* (in the role of a *place*) via a *door* (in the role of a *door-connector*) or via a *stairs* (in the role of a *stairs-connector*). The resulting benefit is that it makes knowledge more clear and less cumbersome to query than classical knowledge-graphs. This benefit is used in the ontology proposed next, when creating inference-rules on room connections.

## 4. An ontology for indoor route-planning

The hierarchical relation of the entities for mapping in Figure 1 and for automated planning in Figure 2 are now used to introduce the main part of the proposed ontology depicted in Figure 3. To distinguish between the three different concepts, i.e., entity, relation and attribute, each of them is illustrated with a particular symbol: a rectangle for entity, a diamond for relation and an ellipse for an attribute. Roles are indicated alongside each edge. A brief description of some typical hypergraph-relations is presented next, as entities were already introduced. Please contact the authors for the complete ontology.

- *task requirement* is a relationship with 3 roles to specify that any *task* requires particular *building conditions* and a particular value of the planner's *state*;
- *divisioning* is a relationship with 2 roles to specify that a *compound task* can be partitioned into a number of (more detailed) *compound* and/or *primitive tasks*;

- *room connection* is a relationship with 3 roles to specify that one *door*, or a series of *doors*, connects two *rooms* in the building, which could be a necessary condition to plan one of the *tasks* of the *robot*;
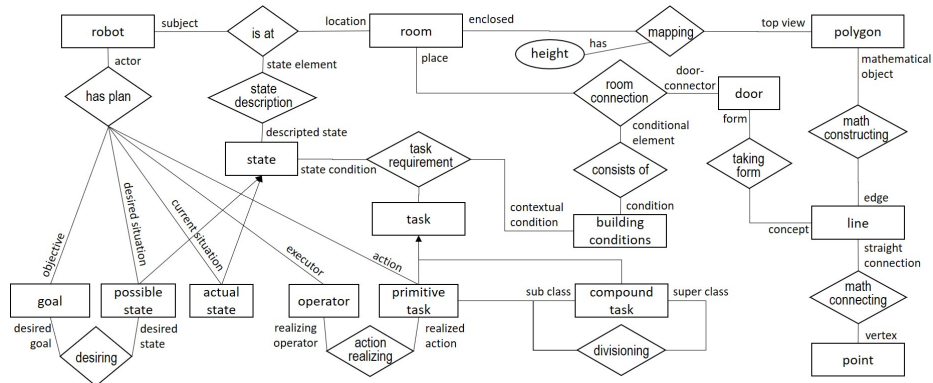


**Figure 3.** The main part of the schema, or ontology, for route planning. Rectangles represent entities, ellipses represent attributes and diamonds represent relations, while arrows without labels indicate a sub-relation.

Let us complete this introduction of the ontology with the example depicted in Figure 4, while a more elaborated example can be received upon request. The instances in Figure 4 are related to the hierarchy in Figure 1 as follows: *LivingWall*, *HallWall* and *KitchenWall* which are instances of a *wall*, *LivingDoor* and *KitchenDoor* which are instances of a *door*, while *Living*, *Hall* and *Kitchen* are instances of a *room*. Further, the figure depicts one prior relation, i.e., *room-connection*, and one new relation of the ontology: a *constructing* relationship indicating how a *room* that has the role of a *construction* is made out of the different *structural elements* having the role of a *part*.
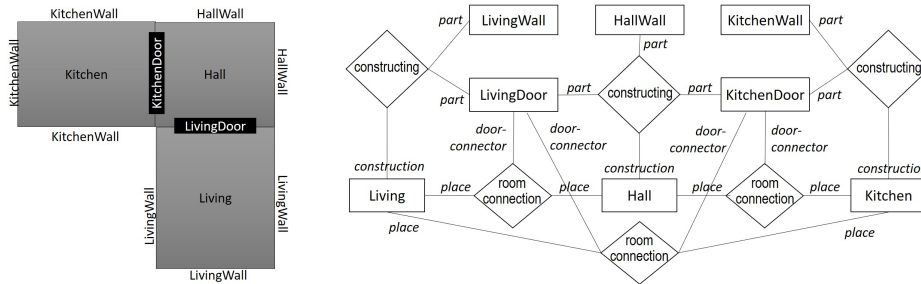


**Figure 4.** Actual data, or instances, of the developed ontology for a floor plan of a living, hall and kitchen.

The *room connection* relation shown in Figure 4 is obtained via the following rule, which benefits from the ability of GRAQL to use n-ary relations:

*room-connection sub rule,*
*when { $x isa room; $y isa room; $x != $y; $z isa door;*
*(construction: $x, part: $z) isa constructing;*
*(construction: $y, part: $z) isa constructing;*
*}, then {(place:$x, place:$y, door-connector: $z) isa room-connection};*

If the robot wants to go from one place to the next, it can query the *room connection* between these places as that will produce the *door*-connectors. For example, a query on the *room connections* of the *Living* will show that the *Hall* is reached via the *LivingDoor*. By making the *room connection* rule transitive, it can connect all places with all connectors.

## 5. Conclusions

An ontology for route planning through a building was presented by combining knowledge of a floor plan with knowledge on automated planning. The ontology was further extended with automated inference to prune the possible route-alternative and thereby, prevent that an automated planner explores infeasible routes. As such, our approach makes a separation between resource intensive, data driven methods and inference on symbolic knowledge. Another benefit, as our ontology is implemented as a knowledge-hypergraph, is that inference-rules are much easier and faster to implement compared to more traditional approaches on knowledge databases, mainly due to the fact that a relation is a hierarchical concept of multiple roleplayers. With little effort a human operator can thus enter symbolic knowledge making the use of, e.g., SLAM, unnecessary. Future work is to further integrate this knowledge-hypergraph with a route- planner on an actual robot.

## References

[1] Durrant-Whyte H, Bailey T. Simultaneous localization and mapping (parti & part ii). IEEE Robotics Automation Magazine. 2006 Sep;13:99-18.

[2] Yasin A, Nasser Y, Awad M, Al-Dubai A, Liu R, Yuen C, Raulefs R, Aboutanois E. Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. IEEE Communications Surveys & Tutorials. 2017;19(2):1327-19.

[3] Wang Y, He H, Sun C. Learning to Navigate Through Complex Dynamic Environment With Modular Deep Reinforcement Learning. IEEE Trans. on Games. Dec 2018;10(4):400-12.

[4] Levihn M, Scholz J, Stilman M. Hierarchical Decision Theoretic Planning for Navigation Among Movable Obstacles. In: Frazzoli E, Lozano-Perez T, Roy N, Rus D, editors. Proc. of the 10th Workshop on the Algorithmic Foundations of Robotics, 2013; Cambridge, Massachusetts: Springer; p. 19-16.

[5] Kim GH, Suh IH, Suh H. Ontology-based unified robot knowledge for service robots in indoor environments. IEEE Trans. on Systems Man and Cybernetics - Part A Systems and Humans. June 2011; 41(3):492-17.

[6] Olivares-Alarcos A, Beßler D, Khamis A, Goncalves P, Habib MK, Bermejo-Alonso J, Barreto M, Diab M, Rosell J, Quintas J, Olszewska J, Nakawala H, Pignaton E, Gyrard E, Borgo S, Alenya G, Beetz M, Li H. A review and comparison of ontology-based approaches to robot autonomy. The Knowledge Engineering Review. 2019; 34: 1-29

[7] Joo SH, Manzoor S, Goncalves-Rocha Y, Bae SH, Lee KH, Kuc TY, Kim M. Autonomous Navigation Framework for Intelligent Robots Based on a Semantic Environment Modeling. Applied Sciences, 2020 May 5; 10: 3219-20.

[8] Pribadi H.The GRAKN.AI ontology: simplicity and maintainability, in comparison with traditional ontology languages and tools [Internet]. London, Haikal Probadi [cited 2020 Aug 2]. Available from: https://blog.grakn.ai/the-grakn-ai-ontology-simplicity-and-maintainability-ab78340f5ff6.

[9] Adão T, Magalhães L, Peres M. Ontology-based Procedural Modelling of Traversable Buildings Composed by Arbitrary Shapes. Cham: Springer International Publishing; 2016: 122 p.

[10] Pease A. Ontology: A practical guide. Angwin: Articulated Software Press; 2011. 313 p.

[11] Fox M, Long D. PDDL2.1: An extension to PDDL for expressing temporal planningdDomains. Journal Artificial Intelligence Research. 2003 Dec;20:60-64.

[12] Georgievski I, Aiello M. HTN planning: Overview, comparison and beyond. Artificial Intelligence. May 2015; 222:124-32.