From Structure to Actions: Semantic Navigation Planning in Office Environments

Klaus Uhl and Arne Roennau and Rüdiger Dillmann

Abstract—The use of meaning in mapping and navigation is inevitable if a robot has to interact with its environment in a goal-directed way. Moreover, a semantic environment model makes navigation planning more efficient and simplifies the review and communication of the robot’s knowledge. Existing work in this area decomposes the environment into places, which can be distinguished using the robot’s sensors. However, if important features of the environment cannot be detected by the robot’s sensors a different approach is needed.

This paper introduces the Semantic Region Map, an environment model with complex metric, topological and semantic features. It shows how navigation points, so-called semantic positions, can be deduced from the map using a semantic description of the environment. Furthermore, the semantic positions are connected to a reachability graph, whose edges are labelled with robot actions, using a semantic description of the robot’s capabilities. An ontology consisting of a taxonomy and a set of rules are used to implement the semantic models.

The concept of the Semantic Region Map is applied to a robot operating in an office environment.

I. INTRODUCTION

If a service robot has to interact with its environment in a goal-directed way, the use of meaning in mapping and navigation is inevitable. This is especially true if a service robot is designed to assist humans in everyday tasks. The surroundings in which humans live and work are usually divided into discrete spatial regions, such as corridors, offices, bedrooms etc. A robot which can reason about the meaning and relations between those regions is able to more easily and naturally communicate with the people it has to assist as it can understand commands like “bring this batch of letters to the secretary of the public relations department” (cf. [1]).

Aside from that, a semantic environment model makes navigation planning more efficient, navigation execution more robust and simplifies the review and communication of a robot’s knowledge.

Existing work in this area decomposes the environment into places which can be distinguished using the robot’s sensors and uses those places as navigation points for the robot. However, taking the price, dimensions and weight of sensors into account, there will always be important features in an environment which cannot be detected by the robot because it is not equipped with enough sensors to detect them.

If a robot is able to detect and classify regions, perceive relations between the detected regions and determine relations between its own position and the detected regions, more flexibility is possible. Navigation points can suddenly be independent of distinguishable places and can be located in a density only limited by the granularity of distinguishable relations. Additionally, regions which cannot be detected by the robot’s sensors can be handled indirectly via inference. The dense navigation points, in turn, give the planner fine control over the motion behaviour of the robot on a semantic level.

This paper introduces the Semantic Region Map as the basis for abstract, semantic navigation planning for robots operating in indoor environments. It shows how an environment can be modelled using complex region features consisting of metric, topological and semantic information. It shows how the Semantic Region Map can be combined with a generic region algebra and a semantic model of a concrete environment to deduce abstract navigation points, so-called semantic positions, from the map. By adding a semantic model of a concrete robot, the semantic positions can be connected to a reachability graph whose edges are labelled with the actions the robot has to perform in order to move from one semantic position to the next. This gives the planner fine control over the exact robot behaviour along its path. Using this semantically enriched environment model, planning a navigation path is reduced to determining the current and goal semantic positions of the robot using queries to the ontology, extracting the reachability graph and finding the shortest path between the two semantic positions.

This paper is organised as follows. First, we briefly describe related work and describe the semantic mission control system to which this work belongs. Then we introduce the semantic navigation planning concepts, followed by a general procedure for modelling an environment and a robot for a specific application. We apply the modelling procedure to a robot operating in an office environment and show experimental results. Finally, we conclude and give an outlook to future work.

II. RELATED WORK

Belouaer et al. [2] describe an ontology-based, semantic representation of spatial entities, spatial relations and imprecise spatial information. Spatial entities are modelled as axis-aligned rectangles and an algebra of topological relations allows to deduce relations between distant entities. Although the system is designed to support path planning, path planning is limited to the purely geometric level and the system cannot handle different driving strategies like wall following, door traversal and straight driving as navigation actions.
Galindo et al. [3] have developed a semantic map framework in which spatial information is anchored to semantic labels which are, in turn, connected to a conceptual ontology of the environment. The system is tailored to deriving the existence of spatial entities which have not yet been seen and to refine the classification of spatial entities by deduction. A semantic-level planning algorithm uses the semantic map and the conceptual ontology to start planning on the conceptual level. Motion planning, however, is restricted to moving the robot from one spatial area to the next without fine control over the actual motion behaviour.

Guitton and Farges [4] combine a general task planner with a specialised path planner to a hybrid mission planning system. Navigation tasks are modelled as preconditions to other actions. As there can be behavioural as well as geometric constraints for the path planner, it is possible to enforce a specific driving behaviour. However, this behaviour cannot be tied to a single spatial area and cannot be changed in different areas.

Mozos et al. [5] propose a multi-hierarchical map which links a metric map, a topological navigation map, a topological area map and a conceptual map. The ontology which backs the conceptual map has similar deduction capabilities and limitations as the work of Galindo et al. [3].

Shi et al. [6] propose an algorithm to create a semantic grid map from laser range data. Each cell of the grid map is semantically labelled to be either a room, corridor or doorway. By using a grid map Shi et al. are able to classify subregions of a single laser scan to different semantic classes. However, they do not currently use their maps for navigation planning.

III. SYSTEM CONTEXT

The semantic navigation planning system described in this paper is part of a larger semantic mission control system [7]. The system architecture is shown in Fig. 1. It consists of nine modules in four layers which are distinguished by the kind of data processed.

The semantic level consists of the User Interface which communicates with the system’s user. It also contains the semantic navigation planning system (Semantic Navigation).

The symbolic-semantic level contains the Semantic Mapping which computes and updates the Semantic Region Map of the environment. It uses a semantic SLAM algorithm with complex features that capture metric, topological and semantic properties [8]. Also located on this level are the Semantic Localisation which determines and tracks the robot’s current semantic position and the Execution Unit which decomposes plans from the Semantic Navigation into individual symbolic actions and monitors the plan’s execution.

The subsymbolic-symbolic level contains the Navigation Data Analysis and the Basic Control. The Navigation Data Analysis continuously locates and classifies regions in the robot’s sensor data and determines their parameters and relations. The Basic Control receives a single symbolic action from the Execution Unit at a time, passes it as subsymbolic commands to the sensor and actor interfaces and monitors its execution. The subsymbolic level contains the sensor and actor interfaces to the robot.

The implementation of the mobile robot is mostly independent of the semantic mission control. In the case of our mobile research robot Odete (see Fig. 4) and our autonomous shopping trolley InBot (see Fig. 5), behaviour-based robot control systems have been implemented. They are capable of executing a set of complex behaviours which are mapped to subsymbolic commands in the actor interface. The detection and tracking of dynamic and semi-dynamic obstacles is also implemented in the robot control software as it has to be tightly integrated with the robot’s safety functions. Several different obstacle avoidance behaviours, which can be activated independently, use this tracking information to safely navigate in crowded environments.

IV. SEMANTIC NAVIGATION PLANNING

The semantic navigation planning system consist of two parts: an ObjectLogic [9] ontology, which contains knowledge about the application domain, the robot and the environment, as well as a planner, which extracts knowledge from the ontology and creates navigation plans.

A. Semantic Region Map

The first major concept of the semantic navigation planning system is the Semantic Region Map. It segments an environment into a set of regions with metric, topological and semantic features. Each region is an instance of a subclass of the Region concept in the ontology. The region class represents the semantic meaning of a region. Regions are topologically connected to their neighbours via the neighbourOf relation. They can also be fully contained in other regions, in which case they are connected via the containedIn relation. Additionally, the relative orientation of neighbouring regions is specified via one of the four relations northOf, eastOf, southOf and westOf.

The metric feature of a region depicts its approximate geometric extent in a global coordinate system. It consists of a centre rectangle and two connected sub-rectangles, which can be moved along the left and right edges of the centre rectangle. Therefore the region geometry can be described...
via the tuple \((x, y, w, h, \gamma, w_r, h_r, h_l, y_r)\). The left and right sub-rectangle can be omitted if they are not needed to describe a region’s geometry. In this case the values \(w, h\) and \(y\) or the values \(w_r, h_r, y_r\) are set to zero. The region geometry is attached to a region instance in the ontology via the hasShape relation.

The ObjectLogic specification for the Region concept is defined as follows:

\[
\text{Region} \{\text{eastOf} \{0:*\}, \text{inverseOf} \{\text{westOf}\} \} \Rightarrow \text{Region},
\]
\[
\text{northOf} \{0:*\}, \text{inverseOf} \{\text{southOf}\} \Rightarrow \text{Region},
\]
\[
\text{southOf} \{0:*\} \Rightarrow \text{Region}, \text{westOf} \{0:*\} \Rightarrow \text{Region},
\]
\[
\text{hasShape} \{1:1\} \Rightarrow \text{Shape}, \text{containedIn} \{0:*\} \Rightarrow \text{Region},
\]
\[
\text{neighbourOf} \{0:*\}, \text{symmetric} \Rightarrow \text{Region}. \]

The semantic navigation planning system expects a Semantic Region Map as its input. This map has to specify the regions in the environment, their shape as well as the neighbourOf and containedIn relations. For regions which are connected with a neighbourOf relation, it must also specify one of the northOf, eastOf, southOf and westOf relations.

**B. Semantic Positions**

The second major concept of the semantic navigation planning system are semantic positions. They are fuzzy navigation points that are defined by semantic relations to regions in their surroundings.

Modelling the semantic positions for an application domain is a complex task. It can become even more tedious when different orientations of regions and neighbouring regions have to be considered because the number of possible combinations explodes. To counteract this, each region has a local coordinate system which is rotated against the global coordinate system of the Semantic Region Map according to the orientation of the region. Most relations between a region and its implied semantic positions are specified in this local coordinate system.

The ObjectLogic specification for the SemanticPosition concept is defined as follows:

\[
\text{SemanticPosition} \{\text{impliesBy} \{0:*\}, \text{inverseOf} \{\text{implies}\} \} \Rightarrow \text{Region},
\]
\[
\text{inRegion} \{0:*\} \Rightarrow \text{Region},
\]
\[
\text{spAtStartOf} \{0:*\} \Rightarrow \text{Region},
\]
\[
\text{spAtEndOf} \{0:*\} \Rightarrow \text{Region},
\]
\[
\text{near} \{0:*\} \Rightarrow \text{Region}, \text{visAVis} \{0:*\} \Rightarrow \text{Region},
\]
\[
\text{localEastOf} \{0:*\} \Rightarrow \text{Region}, \text{localNorthOf} \{0:*\} \Rightarrow \text{Region},
\]
\[
\text{localSouthOf} \{0:*\} \Rightarrow \text{Region}, \text{localWestOf} \{0:*\} \Rightarrow \text{Region},
\]
\[
\text{local} \{\text{East}\}/\{\text{North}\}/\{\text{South}\}/\{\text{West}\} \text{SideOf} \{0:*\} \Rightarrow \text{Region},
\]
\[
\text{local} \{\text{East}\}/\{\text{North}\}/\{\text{South}\}/\{\text{West}\} \text{mostAlong} \{0:*\} \Rightarrow \text{Region},
\]
\[
\text{local} \{\text{East}\}/\{\text{North}\}/\{\text{South}\}/\{\text{West}\} \text{mostIn} \{0:*\} \Rightarrow \text{Region},
\]
\[
\text{local} \{\text{East}\}/\{\text{North}\}/\{\text{South}\}/\{\text{West}\} \text{mostAlong} \{0:*\} \Rightarrow \text{Region},
\]
\[
\text{neighbourOf} \{0:*\}, \text{symmetric} \Rightarrow \text{SemanticPosition}. \]

These relations specify the relative orientation (north, east, south or west of a region) and location (at the start, centre or end of a region) of semantic positions. They also define if semantic positions are in a region, outside but near a region or vis-à-vis. And they form local neighbourhood graphs between the semantic positions that are implied by the same region.

**C. Semantic Navigation Algebra**

The semantic navigation system contains an algebra, i.e. a set of rules in the ontology which performs calculations that reduce the complexity of the application domain model and the number of facts that have to be explicitly asserted in the Semantic Region Map.

Although the region geometry contains a rotation angle, most of the time we only deal with a discrete set of semantic orientations. The ontology, therefore, introduces the Orientation concept and the four orientation instances East, North, South and West. The semantic navigation algebra derives the semantic orientation of each region from its rotation angle by assigning a 90° segment to each orientation instance.

As the relations between semantic positions and the regions by which they are implied are specified in the regions’ local coordinate system, the semantic navigation algebra converts them into the global coordinate system. If semantic positions have relations to other regions, the algebra converts from the global coordinate system back into the local coordinate systems of those regions. In order to make the modelling of robot actions easier, the semantic navigation algebra also contains rules which connect the semantic positions to a global neighbourhood graph.

**D. Robot Actions**

The final component is the model of robot actions. It is built around two concepts in the ontology. The actions which the robot in a specific application domain can perform are modelled as instances of the Action concept. The Reachability concept is used to connect pairs of neighbouring semantic positions with actions using the ternary ReachableByAction function symbol, thus generating a directed reachability graph labelled with robot actions:

\[
\text{Action}()\text{. Reachability}(). \text{ReachableByAction}(?\text{StartSP}, ?\text{EndSP}, ?\text{Action}): \text{Reachability}
\]

**E. Navigation Planning**

The Semantic Region Map, the model of semantic positions, the semantic navigation algebra and the model of robot actions yield a reachability graph by deduction through the ontology system. This graph is extracted from the ontology by retrieving all semantic position instances and all instances of the Reachability concept. The semantic positions form the nodes of the graph while the reachability instances form the edges, labelled with the action to be performed. A weight is assigned to each edge by calculating the Euclidean distance between the approximate coordinates of the start and end semantic positions.

Navigation goals are specified as a set of relations between the desired target semantic position and regions in its surroundings. Using the reachability graph, navigation planning consists of the following steps:

1) Determine the current semantic position of the robot by querying the ontology with the set of relations to regions the robot has currently detected.
2) Determine the target semantic position by querying the ontology with the set of goal relations.
3) Plan the shortest path between the current and target semantic position in the reachability graph.

Navigation goals may be ambiguous and yield multiple semantic positions when querying the ontology. In this case it is assumed that reaching any of the resulting semantic positions achieves the goal, so the nearest of them is taken.

V. MODELLING METHODOLOGY

When designing the domain ontology for a specific application (i.e. environment and robot) a number of steps have to be performed. First of all, the navigation actions which the robot can perform have to be added to the ontology as instances of the Action concept. Secondly, the relevant region classes that occur in the environment have to be identified. They have to be added to the ontology as subclasses of the Region concept. If applicable, e.g. if regions of a specific class are always longer than wide, a preferred orientation of the local coordinate system has to be defined for some region classes.

Then the interesting navigation points and the conditions in which they are relevant have to be determined. This is done following a three-step procedure:

1) Identify interesting navigation points for each region class.
2) For each pair of region classes and each possible topological relation of the two, identify additional navigation points that are of interest in this special combination.
3) Determine which navigation points generated by the same region should be considered neighbours.

For each identified interesting navigation point a rule which derives a semantic position has to be added to the ontology. The body of this rule has to contain the condition under which the semantic position should be derived. The head of the rule has to assert a semantic position with a unique name. A unique name can be created by choosing a unique function symbol and adding the region from which the semantic position is implied as a function argument. The head of the rule also adds relations to regions and other semantic positions (see Sec. VI for an example).

Finally, the robot actions have to be considered in order to connect the semantic positions to form a reachability graph. This follows a procedure similar to identifying the interesting navigation points:

1) For each region class look at the implied semantic positions and check if the robot can move between two adjacent semantic positions with a specific action.
2) For each pair of region classes look at the implied semantic positions and check if the robot can move between two adjacent semantic positions with a specific action.
3) For each action check if there are generic conditions in which the robot can use this action to reach an adjacent semantic position.

Each identified reachability rule has to be added to the ontology. The rules have to assert a Reachability individual in their head, using the ternary ReachableByAction function symbol.

VI. MODELLING AN OFFICE ROBOT APPLICATION

To validate the semantic navigation system and the modelling methodology, an office robot application has been chosen. The target platform is our mobile research robot Odete, but the model can be easily transferred to any robot that can execute the same abstract actions. Odete’s task is to navigate through an office environment conducting transports. To make the robot’s behaviour more predictable for people in the office, the robot should always stick to the right wall in the direction of travel when driving in corridors.

Following the methodology of Sec. V we first list the actions which the robot can perform:


We assume that the FollowWall action is able to follow walls around corners, although we could easily factor this behaviour out into a separate action if the robot implementation would require it.

Now, we identify the relevant region classes that occur in the office environment:

Corridor::Region. Door::Region. Room::Region.

We define that the longer sides of doors and corridors have to face north in their local coordinate systems.

Next, we look at each pair of region classes. The only combination that is of interest here is a door at the side of a corridor. As the robot should always drive along the right wall in corridors it must be able to turn to a door from the opposite side. Therefore, if a door is at the side of a corridor we place two additional navigation points at the start and end of the doorway vis-a-vis the door.

Now, we can add rules to the ontology, which assert the identified navigation points as semantic positions with appropriate relations in their head. The condition for deriving the navigation points goes into the body of these rules. The following rule derives the three semantic positions east of a door. Similar rules have to be written for the other interesting navigation points.

SP1(?Door):SemanticPosition[impliedBy->?Door, localEastOf->?Door, near->?Door, atStartOf->?Door, inRegion->Other, neighbourOf->SP2(?Door)] AND
SP2(?Door):SemanticPosition[impliedBy->?Door,
Having identified the relevant semantic positions, the next step of the modelling methodology is to look at the robot actions. First, we look at each region class and their implied semantic positions. We find that doors have to be traversed using the TransitDoor action. Therefore the semantic positions at the centre of each side of the door have to be connected in both directions. We further have defined that the FollowWall action should always be used when the robot drives along the right side of a corridor. If a door is at the end of a corridor the semantic positions at its side also have to be connected to the last semantic positions at the corresponding sides of the corridor using FollowWall.

In the second step we have to look at each pair of region classes and their implied semantic positions:

- If the robot is in a room and needs to pass through a door, we define that it has to drive to the door using the TurnToDoor action, first.
- If the robot has entered a room through a door, it should drive further into the room using the DriveStraight action.
- If the robot turns left after having traversed a door into a corridor, it has to cross the corridor and proceed along the opposite wall. This is accomplished by performing the TurnFromDoor action.
- If the robot drives in a corridor and has to traverse a door on the opposite side of the corridor, it has to cross the corridor using the TurnToDoor action.

All identified reachability rules have to be added to the ontology. The following rule connects semantic positions along the east side (in the corridor’s local coordinate system) of a corridor with the FollowWall action:


VII. EXPERIMENTS

To test the validity of our models we conducted several experiments with two different maps. The first map represents a very simple, artificial office environment (see Fig. 2(a)). This simple map has been used to validate and visualise the individual conceptual steps the semantic navigation system performs.

Fig. 2(b) shows the deduced semantic positions. Remark that the semantic positions are not characterised by their geometric position, although the visualisation might suggest otherwise. Fig. 2(c) shows how the model of the robot’s actions connected the semantic positions to a reachability graph. Notice that the arrows along the right side of the corridor point upwards while the arrows along the left side point downwards.

Fig. 2. Experiments with a Semantic Region Map of a simple office environment. (a) The Semantic Region Map models the environment in an abstract way. (b) Semantic positions have been implied. (c) The semantic positions have been connected with actions to a reachability graph. (d) A path has been planned from the bottom right room to the top left room. Its edges are labelled with the actions the robot has to perform.

Fig. 3. Experiments with a Semantic Region Map of a more complex office environment.
of the corridor point downwards. Therefore, the robot always drives along the right side of the corridor.

Finally, Fig. 2(d) shows a path that has been planned by the semantic navigation system from the bottom right room to the top left room. The robot starts by turning to the door and passing through it. Then the robot follows the eastern wall of the corridor until the south end of the target room’s door. It turns to the door thereby crossing the corridor, passes through the door, and lastly drives straight into the room.

We also conducted experiments with the Semantic Region Map of a larger office environment. Fig. 3(a) and 3(b) show two paths that have been planned by the semantic navigation system, along with the reachability graph. A remarkable result of our model can be seen in Fig. 3(a) in the small corridor on the bottom left side of the map: The robot strictly adheres to the “always drive right” policy, although one might argue that it would be more efficient to drive straight between the two doors in this situation. This could be achieved by introducing a NarrowCorridor region class and modelling semantic positions and robot actions accordingly.

VIII. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

This paper introduced the Semantic Region Map, an environment model with complex metric, topological and semantic features. It presented how navigation points, so-called semantic positions, could be deduced from the map using a semantic description of the environment. Furthermore, it showed how a semantic description of the robot’s actions can be used to connect the semantic positions to a reachability graph, whose edges were labelled with robot actions. An ontology consisting of a taxonomy and a set of rules was used to implement the semantic models. The paper introduced a methodology to model concrete environments as well as robot actions. It applied this methodology to a service robot operating in an office environment. Experiments with the map of a small office showed that the reachability graph was deduced as expected and that paths could be planned by determining the current and goal semantic positions using abstract queries to the ontology, extracting the reachability graph and finding the shortest path between the two semantic positions. Further experiments with a more complex map showed that the approach scales well.

B. Future Works

In future work we will extend the model of the office robot application with additional region classes and actions in order to make it more capable and flexible. It will also be possible to mark regions as “not passable” so that temporarily blocked regions (e.g. closed doors) can be handled. Additionally, we will integrate the semantic navigation planning system with the semantic SLAM algorithm from [8] and a semantic localisation system. This will enable us to test the entire loop from mapping to path planning to path execution on our mobile research robot Odete (see Fig. 4).

We are also planning to port the semantic navigation system to InBot, our autonomous shopping trolley (see Fig. 5), and an automatic guided vehicle (AGV), which transports goods in hospitals.

Moreover, we will add dynamic obstacles to the Semantic Region Map by introducing a DynamicObject concept into the ontology. This information will be used in the planner to adjust the driving behaviour of the robot depending on the types, quantity and motion of dynamic obstacles within a region.

IX. ACKNOWLEDGMENTS

The authors thank ontoprise GmbH for providing research licences for their ObjectLogic ontology products OntoStudio and OntoBroker at no charge.

REFERENCES