Walter Mauricio Mayor Toro

WEB-BASED RELATIONAL ROBOT NAVIGATION UNDER UNCERTAIN PERCEPTION

São Paulo
2014

Walter Mauricio Mayor Toro

WEB-BASED RELATIONAL ROBOT NAVIGATION UNDER UNCERTAIN PERCEPTION

Dissertation submitted to Escola Politécnica da Universidade de São Paulo in partial fulfillment of the requirements for the degree of Master of Science.

São Paulo
2014

Walter Mauricio Mayor Toro

WEB-BASED RELATIONAL ROBOT NAVIGATION UNDER UNCERTAIN PERCEPTION

Dissertation submitted to Escola Politécnica da Universidade de São Paulo in partial fulfillment of the requirements for the degree of Master of Science.

Area of research:
Systems Engineering

Advisor:
Prof. Dr. Anna Helena Reali Costa

São Paulo
2014

A quien yo dedico,

A Dios, mi esposa, mi hija isabella, a toda mi familia y a mis amigos.

ACKNOWLEDGMENTS

First and foremost, I thank God.

I would like to thank my advisor, Anna Helena Reali Costa. She has given me essential guidance for shaping the research and encouraged me to achieve to the best of my ability.

I also would like to thank my wife Andrea and my daughter Isabella, for their understanding and support.

I would like to offer my special thanks to my family, especially my mother, Martha; my father, Arnulfo; my sister, Jessica; and my aunt, Cecilia for their support and encouragement.

I wish to acknowledge the help provided by prof. Fábio Cozman, with advice and assistance in writing papers.

I also thank all the members of LTI (Laboratório de Técnicas Inteligentes - USP) for the valuable discussions and comments regarding my research.

I would like to offer my special thanks to my friend Juan Carlo Perafan, for the valuable discussions and support during the Master's course.

My special thanks are extended to all my friends, for their support and understanding.

I also gratefully acknowledge the financial support from CNPq (National Council of Scientific and Technological Development).

"Every sentence I utter must be understood

not as an affirmation, but as a question."

Niels Bohr

RESUMO

Quando um robô autônomo tenta resolver as tarefas de navegação dentro de um ambiente real interno usando relações qualitativas, vários problemas aparecem tais como observação parcial do ambiente e percepção incerta. Isso ocorre porque os sensores do robô não proporcionam informação suficiente para perceber completamente as situações do ambiente, além de incorporarem ruído no processo. A web semântica dota o robô autônomo com a habilidade de obter conhecimento de senso comum extraído da web, conhecimento este que os sensores do robô não podem proporcionar. Porém, nem sempre é fácil levar efetivamente estes recursos semânticos da web ao uso prático. Neste trabalho, foi examinado o uso de recursos semânticos da web na navegação robótica; mais especificamente, em uma navegação qualitativa onde o raciocínio incerto desempenha um papel significativo. Nós avaliamos o uso de uma representação relacional; particularmente, na combinação da informação semântica web e dos dados de baixo nível proporcionados pelos sensores, permitindo uma descrição de objetos e das relações entre os mesmos. Esta representação também permite o uso de abstração e generalização das situações do ambiente. Este trabalho propõe a arquitetura *Web-based Relational Robotic Architecture* (WRRA) para navegação robótica que combina os dados de baixo nível dos sensores do robô e os recursos web semânticos existentes baseados em lógica descritiva probabilística, com o aprendizagem e planejamento relacional probabilístico. Neste trabalho, mostramos os benefícios desta arquitetura em um robô simulado, apresentando um estudo de caso sobre como os recursos semânticos podem ser usados para lidar com a incerteza da localização e o mapeamento em um problema prático.

Palavras chave: Robótica Semântica, Navegação Robótica, Sistema KnowRob, Raciocínio Lógico Probabilístico, Arquitetura Robótica.

ABSTRACT

When an autonomous robot attempts to solve navigation tasks in a qualitative relational way within a real indoor environments, several problems appear such as partial observation of the environment, and uncertain perception, since the robot's sensors do not provide enough information to perceive completely the environment situations, besides the sensors incorporate noise in the process. The semantic web information endows the autonomous robot with the ability to obtain common sense knowledge from the web that the robot's sensors cannot provide. However, it is not always easy to effectively bring these semantic web resources into practical use. In this work, we examine the use of semantic web resources in robot navigation; more specifically, in qualitative navigation where uncertain reasoning plays a significant role. We evaluate the use of a relational representation; particularly, in the combination of the semantic web and the low-level data sensor information, which allows a description of relationships among objects. This representation also allows the use of abstraction and generalization of the environment situations. This work proposes the framework *Web-based Relational Robotic Architecture* WRRA for robot navigation that connects the low-level data from robot's sensors and existing semantic web resources based on probabilistic description logics, with probabilistic relational learning and planning. We show the benefits of this framework in a simulated robot, presenting a case study on how semantic web resources can be used to face location and mapping uncertain in a practical problem.

Keywords: Semantic robotics, Robotics Navigation, KnowRob system, Probabilistic Logic Reasoning. Robotics Architecture.

LIST OF FIGURES

LIST OF ABBREVIATIONS

WRRA	Web-based Relational Robotic Architecture
OMICS	Open Mind Indoor Common Sense
POMDP	Partially Observable Markov Decision Process
SVM	Support Vector Machines
OMCS	Open Mind Common Sense
RDF	Resource Description Framework
OWL	Web Ontology Language
RMDP	Relational Markov Decision Process
MDP	Markov Decision Process
RL	Reinforcement Learning
AMCL	Adaptive Monte Carlo Localization
HLC	High Level Control
LLC	Low Level Control
tf	transform tree
MORSE	Modular Open Robots Simulation Engine
QR	Quick Response

LIST OF SYMBOLS

Σ	Relational vocabulary
$\mathcal{B}$	Background knowledge
$\mathcal{S}$	Set of ground situations
$\mathcal{A}$	Set of all ground behaviors
T	Transition function of a Markov Decision Process
R	Reward function of a Markov Decision Process
$\mathcal{G}$	Set of goal states
b^0	Probability distribution for the initial situation in a Markov Decision Process
s	Ground situation
σ	Abstract situation
α	Abstract behavior
$\mathcal{A}_\alpha(s)$	Set of all feasible ground behaviors covered by abstract behavior α in situation s
$\mathcal{S}_{ab}$	Set of abstract situations
$\mathcal{A}_{ab}$	Set of all abstract behaviors
a	Ground behavior
ϕ_s	Abstraction function for situations; maps ground situations into abstract situations
$\mathcal{S}_{\sigma_1}$	Set of ground states covered by an abstract state σ
$\pi(s,a)$	Probability of taking behavior a in situation s under stochastic policy π
$\mathcal{A}(s)$	Set of feasible ground actions in situation s

SUMMARY

1 INTRODUCTION

This work is based on the hypothesis that an autonomous robot can interpret the world in a relational and qualitative way as the human beings do, and on the fact that the indoor environments can be described in terms of objects and the relations among them.

For a better understanding, we present a simple example of our approach, let us suppose that we place a mobile robot within a house with the task of going to the living-room and the robot does not know the house, but it has some knowledge provided by the designer. That knowledge is composed by a set of predicates that represent relations between the robot and his surroundings, which are used to inferred the environment situations; a set of behaviors that the robot can execute to change its environment situation; a data based with common sense knowledge about indoor environments taken from the robot knowledge base KnowRob, which uses the Semantic Web technology that is a web of data that provides a common framework to represent information and to facilitate the integration, sharing and reused of different sources of knowledge. Additionally, the designer also provides to the robot a stochastic abstract policy. Besides, the robot has sensors that allow it to perceive its environment and actuators that allow it to move into the environment. Initially, the robot does not know the place where it is, therefore, it uses its sensors to detect the doors and objects present within the place; with the object detected and by using the common sense knowledge the robot infers the place in which it is and the place is annotated with a semantic label, which is a metadata that permits identify places, objects, doors, etc. Let us now suppose that the robot does not see doors but it sees two objects, which are annotated with the semantic labels "freezer" and "oven", then, probably the inferred place will have the semantic label "kitchen". While the robot navigates, it uses these semantic labels to built a semantic map of the environment, which is a representation of not only metric occupancy information but also of semantic knowledge from multiple sources such as category of the place, presence of objects, topology of the environment, etc (PRONOBIS, 2011). Then, the robot uses the information inferred such as the semantic labels and semantic map to activate some predicates from the set of

predicates that the designer provided to the robot. For this case the predicates activates can be `inRoom(kitchen)`, `nonTargetRoom(kitchen)`, `seeNoDoors(living − Room)` and `notAllDoorsVisited()`, after, these predicates are used to inferred an environment situation, which is used along with the stochastic abstract policy to define the behavior that the robot should execute, in this case the a possible selected behavior can be `findDoor()`. When the robot finishes the execution of the behavior, all the process is repeated until the target room is reached.

There are several approaches (AYDEMIR et al., 2011; BEETZ et al., 2011; BEETZ et al., 2012; KLENK et al., 2011) that are focused on equipping autonomous robot with the ability to understand indoor environments similarly to humans, i.e., the robot can understand a place using semantic labels such as "kitchen", "living-room", etc and not using metric information to represent the place as merely a free space. This ability also allows the robot to construct an abstract representation of the environment which, in turn, allows the transfer of knowledge as well as facilitates human-robot iteration. We are interested in that autonomous robots use this ability to navigate within a real indoor environment perceiving the space qualitatively and relationally, which allows performing housework, searching an object located in the environment and developing other tasks. Robot navigation can be seen as a sequential decision task that the robot can perform within its environment. When the robot navigation is performed in a real-world environment, several problems appear: the robot has a partial observation of its surroundings and sensors and actuators introduce considerable uncertainty into the process which, in turn, does not allow the robot to have a complete perception of the environment situations. Indoor environments have properties such as room size, shape, etc. These properties are known as semantic spatial knowledge that the robot can use to know more about its environment and facing the problems aforementioned. In the case of spaces built by humans such as apartments, houses or offices (indoor environments), there is a large amount of semantic spatial knowledge that a robot can use to provide an intuitive spatial location that can be expressed with semantic labels such as *living-room*, *kitchen* or *bathroom*.

However, it is not always easy to effectively bring the semantic information into practical use. Therefore, three issues need to be solved. First, how semantic information and low-level data can be combined; second, how uncertain sensors can be handled; and third, how incomplete maps can be handled. In conventional systems for robot navigation, when a robot navigates within a real-world environment, a large amount of information from the sensors are only interpreted in low-level data. Therefore, problems, such as partial observability of the environment and sensor noise, hinder the complete perception of the environment situations.

Hopefully an autonomous robot can improve its perception ability by using additional sources of information, i.e, the robot can detect all aspects that are relevant to infer the environment situation by using the data from the robot's sensors and the data inferred from additional sources of information. These additional sources can be the *semantic web resources* that have a large amount of information from diverse sources which can be processed and transformed into data that the autonomous robot can use. Recent experience has shown that applications in robotics can benefit from semantic information carrying commonsense facts (AYDEMIR et al., 2013; GALINDO; SAFFIOTTI, 2013; RIAZUELO et al., 2013). One particular example of semantic knowledge system for robotics is the KnowRob package (Knowledge Processing for Autonomous Personal Robots) (TENORTH et al., 2011; TENORTH; BEETZ, 2013). KnowRob operates on ontology databases such as Open Mind Indoor Common Sense OMICS database (GUPTA; KOCHENDERFER, 2004), mixing description logics (BAADER, 2003) and Bayesian networks (PEARL, 1988). The approach presented by Saito et al. (2011) shows that using the KnowRob package, the robot can infer the common location of a specific object, which is used to define the place or the places that the robot has to visit to find this specific object.

Certainly, several approaches have improved the robot's perception of the environment by taking into account semantic information. When researches use semantic information, it is necessary to consider some important points, such as what type of semantic properties are used, i.e, a robot can represent several semantic properties of space such as place size and shape by using semantic labels such as place "big" and "square" respectively; and how the semantic information is modeled. While approaches such as (GALINDO et al., 2005) model the semantic information using ontology based on pure description logic, and works such as (GÖBELBECKER et al., 2012; POLASTRO et al., 2011) model the semantic information using ontology based on probabilistic description logic (KLINOV, 2011). These ontologies are semantic structures for encoding implicit knowledge, which allow segmentation and labeling of data acquired from the robot's sensors during navigation, i.e, the data from the laser sensor can represented objects or attributes of the space. Thus these data could be divided and semantically labeled as "door", "corner", "wall", etc. Furthermore, the difference between the ontology based on pure description logic or probabilistic description logic is that the latter allows working with uncertain semantic concepts.

Since an autonomous robot often deals with large numbers of data acquired from different sources and there is data that do not provide relevant information to specify the environment situations and other data that are shared by different situations, it is necessary to use a representation that defines all the relevant characteristics to repre-

sent a particular situation within the environment. An alternative is to use a *relational representation* that uses objects and their relationships to describe the world, providing a natural abstraction over objects. Besides, it allows representing factored situations (ZARAGOZA; MORALES, 2009; KOGA et al., 2013) that can be used to share experience among similar situations (VAN OTTERLO, 2009).

In this dissertation we mean to respond to the following general questions: how can an autonomous robot perform a relational qualitative navigation within a real-indoor environment under uncertain situations and perception by using additional information from semantic web resources?

1.1 OBJECTIVE

Our main goal is to propose a framework for robot navigation based on semantic web resources that allows for uncertain reasoning to face the uncertain perception of the environment. This framework should use a higher level description to reason at an abstract level.

We consider the use of a higher level of abstraction that allows saving computation as it handles sparser representations. Additionally, a higher level of abstraction enables knowledge to be generalized and reused in new tasks.

1.2 CONTRIBUTION

The main contribution of this dissertation is a framework that allows the integration of the low-level data from the sensors and semantic web resources to face the robot navigation under uncertain situation and perception. This framework is called Web-based Relational Robotic Architecture (WRRA) and has three major modules: Perception, Reasoning and Learning, besides Actuation.

The following are the contributions for each module of WRRA:

- *The Perception module* captures information about space in a qualitative relational way.
- *The Perception module* allows facing the partial observability problem by using common sense knowledge.
- *The Perception module* automatically constructs predicates that are properties of the environment used to describe situations.

- *The Perception module* allows the robot to infer the situations of the environment by using the robot's sensors and semantic web resources.
- *The Perception module* allows facing the uncertain perception by using a probabilistic reasoning of the environment.
- *The Perception module* allows the robot to locate itself in an indoor environment by using qualitative relations and not metrics.
- *The Reasoning and Learning module* allows selecting the behaviors that the robot needs to perform to achieve a particular task by using a higher level of abstraction.
- *The Actuation module* allows the robot to act in an indoor environment without initially having a metric map of the environment.
- *The Actuation module* allows actuating in an indoor environment by using a relative and not global position.

All these characteristics construct a framework that allows autonomous robots to act and to navigate in an indoor environment similarly to humans.

1.3 ORGANIZATION

This document is organized as follows.

- **Chapter 2 – Robot navigation** problem and related work: details the problem we want to solve and reviews works related to semantic information for robot navigation;
- **Chapter 3 – Semantic robot navigation architecture** describes the proposed Web-based Relational Robotic Architecture (WAAR), which allows the robot to navigate under uncertain perception;
- **Chapter 4 – Experiments** describes the navigation domain in which the experiments were carried out and presents the results;
- **Chapter 5 – Conclusions** summarizes our main contributions and future work.

2 ROBOT NAVIGATION PROBLEM AND RELATED WORK

In this chapter we describe in detail the robot navigation problem we want to solve. In Section 2.1 we provide some constraints to clearly define the problem and scope of this work. Then Section 2.2 describes works that are related to this problem.

2.1 PROBLEM DEFINITION

Our major interest lies in the *robot navigation problem* under uncertain situations and perception. That is, taking into account the partial observations of the robot's surroundings, and the sensor noise, the robot must perceive the situations of the environment, which are represented by a set of properties that characterize qualitative relations of the robot within the environment and which may be common to other situations. These properties are obtained by the robot from the environment using its sensors. Although there are approaches that propose to use Partially Observable Markov decision process (POMDP) and other variant structures of that model to solve the problem of partial observability of the environment, we opt for a relational qualitative solution. In this section, we expose the problems and characteristics of our project.

The following are the characteristics of our proposal:

- *Navigation within a real-indoor environment*: we only focus on robot navigation within indoor environments such as apartments, houses and offices.
- *Static environment*: the surroundings in which the robot navigates are unchanging.
- *Uncertain reasoning*: the robot can use common sense knowledge and the sensor information for reasoning about uncertain information.
- *Navigation without initial map information*: the robot knows nothing about the environment, when it is inserted in a new indoor environment.

- *Navigation only uses relative position*: The robot does not know its exact position in the environment; it only knows its relative position with respect to the objects and doors.
- *Navigation in a relational and qualitative way*: the robot defines its location within the environment using qualitative relations and not metric ones.

In this work, we focus on the use of semantic web resources that encode commonsense facts and allow uncertain reasoning by using description logics and relational Bayesian networks. Recent experience has shown that applications in robotics can benefit from semantic information carrying commonsense facts (GÖBELBECKER et al., 2012; SAITO et al., 2011). In the case in which there is qualitative information that the robot's sensors cannot perceive, semantic web resources can be used to infer the information.

The idea that description logics (and probabilistic versions of them) can be used in robotics is not new. Usually, robots have little common sense knowledge and cannot conclude, for instance, that the detection of a refrigerator suggests the current location to be a kitchen. Pure logical reasoning is necessary, but not sufficient, as sensors and actuators introduce considerable uncertainty into the process; hence, probabilities must be employed somehow.

2.2 RELATED WORK

In this section, we are going to present an overview of the related work in the area of semantic web resources in robot navigation; more specifically, in qualitative navigation where uncertain reasoning plays a significant role. Semantic web is a research topic and one of its aims is to enable machines to access and to process the Web information in a more intelligent way. This characteristic of the semantic web can be exploited by robot navigation systems, which can semantically label the sensor data and use these labels to construct several predicates that represent spatial relations of the environment. Additionally, these predicates can be used in a relational representation to define the environment situations. Therefore, first we present several approaches that use, reason and create semantic information and then we present several approaches to obtain semantic information from the semantic web and common sense knowledge.

2.2.1 SEMANTIC INFORMATION FOR INDOOR ROBOT NAVIGATION

There are several place recognition and categorization approaches that recognize the semantic category of a place similarly to that of people, i.e., a semantic label is assigned to each place within an indoor environment, which is a high level concept such as living-room, bathroom, dining-room, etc. Additionally, these approaches can facilitate communication between humans and robots (TOPP et al., 2006). In the literature, these approaches recognize the semantic category of a place using only vision techniques or combined vision with a sonar or laser data. These techniques can be new descriptors, hierarchical multi-modal methods or others. In the case of Descriptors techniques, the idea is to construct a technique to obtain relevant features of images of each place category that allow representing the appearance of each room type, i.e, there are a set of relevant features that describe the kitchen and that are different from the set of relevant features that describe the bathroom. These features can be use to train a machine learning classifier that allows deciding the semantic category of a place (WU; REHG, 2008; WU et al., 2009; MOZOS et al., 2013); and in the case of a hierarchical multi-modal method, the idea is to construct a method that combines features from low-level sources such as images descriptors and laser scans that with features from high level sources such as semantic information by using a probabilistic framework that defines the semantic category of a place (PRONOBIS; JENSFELT, 2011).

Wu and Rehg (2008) present a new image descriptor called Principal component Analysis of Census Transform histograms, which is combined with an Support Vector Machines classifier that allows recognizing place instances in a indoor environment such as "room 113" and the place category of the instance such as "office", i.e, the method recognizes that a image can represent the specific room 113, which is a office room. This approach is extended in (WU et al., 2009) with a new image description called CENTRIS and Bayesian filtering scheme that improves robustness and provides smoother category estimates. In (RANGANATHAN, 2010) the place recognition and categorization is performed using a new technique called Place Labeling through Image Sequence Segmentation, which is implemented using a Bayesian change-point detection framework. Another approach to categorize indoor places is presented by Mozos et al. (2013) that creates a features vector from a range and reflectance data from the laser scan, which is then classified for determining the semantic category by using the supervised learning method called Support Vector Machines SVM (HEARST et al., 1998).

Pronobis et al. (2009) propose a completely supervised multi-modal method that

computes the semantic categorization of the place where the robot is by using a set of properties such as size, shape and appearance of place. These properties are based on local and global image features extracted from camera sensors as well as geometrical features obtained from laser scans. This method is trained using a collection of data that captures properties of each place under various viewpoints, at a fixed time and illumination setting. Furthermore, this approach is extend in (PRONOBIS; JENSFELT, 2011), in which the objects observed in each place are included as a property of the place where the robot is.

Some of the place recognition and categorization approaches presented above are used in other works to construct semantic maps while the robot navigates within an indoor environment. These semantic maps represent the environment with high level concepts by using explicit semantic labels (e.g., living-room, kitchen, etc) to deal with sensor data. The semantic maps can be constructed based on doors, appearance, objects, or using heterogeneous modalities.

Park and Song (2011) propose to construct a semantic map based on door information. This approach uses the doors to separate one place from the other. Then, for each place and each door a node is assigned with a number from 0 to the total number of places and doors. This method recognizes the semantic category of a place by using the frequency of visiting the same door node, when the robot navigates from node 0 that is the first place to the other nodes. Unfortunately, this approach can only recognize two semantic categories: *room* and *living-room*.

There are some semantic mapping approaches based on appearance. Zivkovic et al. (2007) present a semi-supervised method, which begins with a set of unlabeled images taken by the robot from different poses within an indoor environment. Each image is a node that can be joined with other nodes using a similarity measure, i.e., two images are similar if it is possible to perform a 3D reconstruction using the two images. This method proposes to group the nodes in the convex subspace that are parts of the environment that presents similar properties, i.e., an image in a convex subspace is similar with each image within the same subspace and not similar to images in other subspaces. This method also allows a human to specify which nodes are grouped and are not grouped. Finally, each obtained subspace is a semantic category. Ranganathan and Lim (2011) propose to create a semantic map by using the place categorization method present in (RANGANATHAN, 2010), and Kostavelis and Gasteratos (2013) propose another supervised method that extracts the relevant visual features from a RGB-D sensor that are then classified into a semantic category using a Support Vector Machine. Although the methods based on appearance present good results, the change in some condition within an indoor environment such as illumination and robot pose

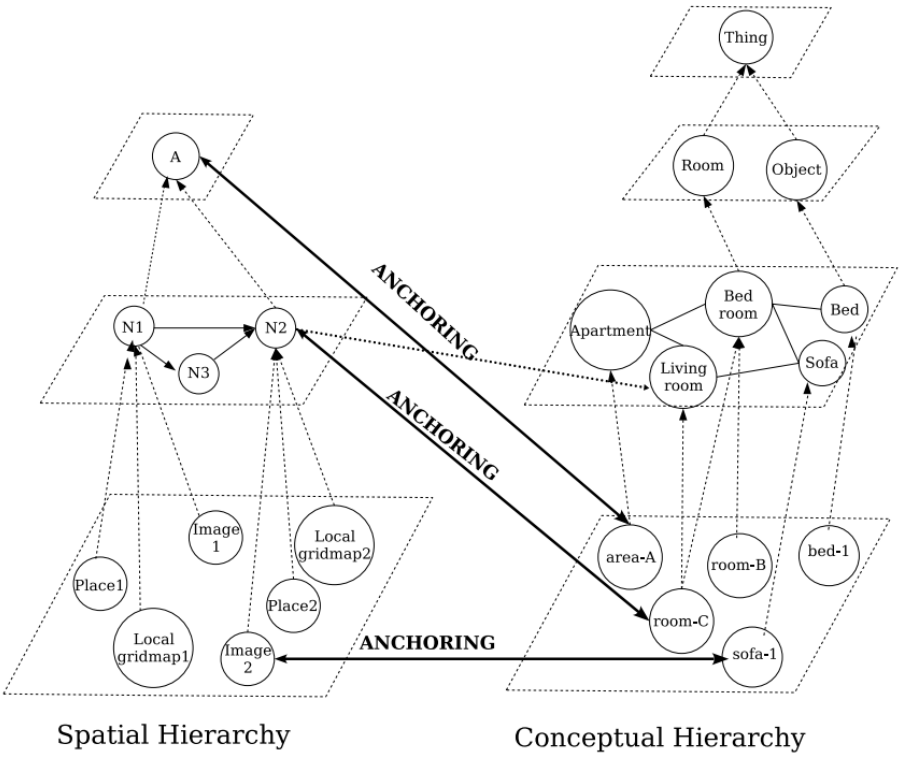
can hinder the categorization of a place.

Recently, semantic maps of indoor environments based on objects has received significant attention (WANG; CHEN, 2011; WU et al., 2014; VASUDEVAN; SIEGWART, 2008; VISWANATHAN et al., 2009; VASUDEVAN et al., 2007; VISWANATHAN et al., 2010), since many indoor environments such as libraries, restaurants, apartments, houses, etc can be identified more easily by taking into account the objects present in them. The approaches present in (VASUDEVAN; SIEGWART, 2008; VISWANATHAN et al., 2010) use probabilistic frameworks based on objects to determine the semantic category of a place. In (VISWANATHAN et al., 2010), the number of occurrences of an object in each place type is obtained from the *LabelMe* database (RUSSELL et al., 2008) by querying each image scene and obtaining the number of occurrences of each object in the image, as in (VASUDEVAN; SIEGWART, 2008). In (WANG; CHEN, 2011), the indoor environment is represented by using a Web Ontology language (OWL) based on objects, this OWL ontology describes the hierarchical organization of ideas in a domain, which is used to describe web information and its relationships. In (WU et al., 2014), a semantic map is constructed by using the objects obtained from the Quick Response (QR) codes that are artificial labels stuck in the objects.

The following works (ROTTMANN et al., 2005; ZENDER et al., 2008; PRONOBIS; JENSFELT, 2012; GALINDO et al., 2005) are approaches that use heterogeneous modalities to construct a semantic map, i.e., they combine different sources of data to determine the semantic category of a place in a semantic map. The robotic system presented by (GALINDO et al., 2005) combines spatial and semantic information of the environment that is stored in two hierarchical structures (see Figure 1).The spatial information is collected from the sensor data. These data are camera images and local grid maps that are stored in the spatial hierarchy. The conceptual information is the common sense knowledge that represents concepts and their relations. This information is stored in the conceptual hierarchy by employing a standard artificial intelligence languages, such as the NeoClassic language (PATEL-SCHNEIDER et al., 1996). Although this research allows inferring the type of room based on detected objects, uncertainties about object locations are not taken into account.

Zender et al. (2008) propose a framework for constructing a semantic map of an indoor environment. This approach has three blocks: the perception block that allows getting information from the robot's sensors; the multi-layer spatial representation block that divides the map representation into layers such as metric, navigation, topological and conceptual that are different levels of abstraction. The conceptual layer allows recognizing the room and object category and how they are related by us-

Figure 1: The spatial and semantic information hierarchies. On the left, spatial information gathered by the robot sensors. On the right, semantic information that models concepts and relations between them. Anchoring is used to establish the links between the two hierarchies (solid lines). Additional links can then be inferred by symbolic reasoning (dotted line).



Source: Reproduced from (GALINDO et al., 2005).

ing common sense knowledge encoded in an OWL-DL ontology that is the knowledge representation OWL languages for semantic web based on description logic (DL). The situated dialog block recognizes the words that the human says. All the information obtained from the sensors are instances of the conceptual layer.

Currently, several mobile robot systems use commonsense reasoning to perform navigation. Approaches such as (KUNZE et al., 2010) that search objects within an indoor environment, believe that a mobile robot can infer control decisions under changing environmental conditions, by integrating the mobile robot's knowledge bases such as the sensor information, with a collection of commonsense knowledge.

The approaches in (GALINDO et al., 2008; GALINDO et al., 2011; GALINDO; SAFFIOTTI, 2013) use semantic knowledge to reason about indoor environments. The work by Galindo et al. (2008) uses common sense knowledge encoded in an OWL-DL ontology to infer navigation goals within an indoor environment taking into account the usual position of the objects. This work is an extension of the approaches (GALINDO et al., 2011; GALINDO; SAFFIOTTI, 2013), which use semantic knowledge to define

norms that are descriptions about the usual object location within an indoor environment. For example, a norm can be that the *towel* should be in the *bathroom*; then, if the robot sees a *towel* in the *kitchen*, the norm is violated and the robot uses this violation to generate a new goal such as "bring the *towel* to the *bathroom*", which corrects the inconsistency. Similarly to (GALINDO et al., 2005), these approaches do not take into account uncertainties regarding object locations.

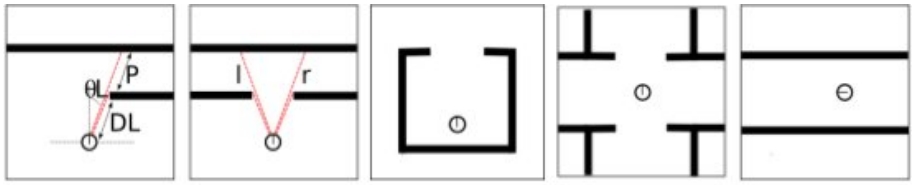
The following works (PRONOBIS et al., 2010; AYDEMIR et al., 2011; HANHEIDE et al., 2011; AYDEMIR et al., 2013) are a sequence of approaches and their principal interest is the robot's ability to find objects in large-scale environments solely using visual sensing. These researches enable the mobile robot to compute the probability of a perceived object to be located in a particular place, using common sense knowledge database and characteristics such as room shape and appearance.

The approaches in (BEETZ et al., 2011; BEETZ et al., 2012; PANGERCIC et al., 2012; SCHUSTER et al., 2012) are a set of works that use common sense knowledge to learn what organizational principles the human uses within a kitchen such as perishable items are stored in a fridge or freezer; or food items are not usually stored together with detergents. These organizational principles are used by the robot to perform intelligent complex tasks such as preparing pancakes.

While an autonomous robot navigates within a real-indoor environment, the robot receives large amount of information from its sensors that must interpret. Semantic labels can be combined with the low-level information from the sensors to help to interpret the large amount of information received. Additionally, these semantic labels can also used to represent environment situations within an indoor environment by using a relational representation. Klenk et al. (2011) propose reasoning about spatial regions in human environments, such as classroom, by using two predicates *leftOf* and *below* that allow relating the position between the object, which are semantically labeled, i.e., two objects with the semantic labels *desk1* and *desk2*, which are located within the classroom as *desk1* is to the left of *desk2*, can be related with the statement *(leftOf desk1 desk2)*. Another approach, such as (AYDEMIR et al., 2011), also proposes relating the object with two different predicates *on* and *in*.

Relational representation can be used to represent the situations of the environment as sets of properties that characterize a particular situation. That set of properties may be common to other situations. The approaches in (KOGA et al., 2014; MATOS et al., 2011; VAN OTTERLO, 2004) use a relational representation to define the states and actions that provide natural abstraction of the environment and allow the generalization of states and actions. The above approaches were only tested in a simple simulated

Figure 2: Attributes and locations.



(a) Discontinuity detection. (b) Discontinuity types. (c) Room. (d) Intersection. (e) Corridor.

Source: Reproduced from (ZARAGOZA; MORALES, 2009).

environment. Koga et al. (2014) use a set of predicates to define a particular state. Some predicates proposed are inRoom(R), seeDoor(D) and nearGoal and one example of a possible state is:

$$s_1 = \texttt{inRoom}(\texttt{r}_{11}) \wedge \texttt{seeDoor}(\texttt{d}_1) \wedge \texttt{closeGoal}(\texttt{d}_1) \wedge \texttt{seeEmptySpace}(\texttt{r}_{12}) \\ \wedge \texttt{closeGoal}(\texttt{r}_{12}) \wedge \texttt{seeEmptySpace}(\texttt{r}_{13}) \wedge \texttt{farGoal}(\texttt{r}_{13}) \wedge \texttt{nearGoal}$$

Where $\texttt{inRoom}(\texttt{r}_{11})$ indicates that the robot is within a room r_{11}; $\texttt{seeDoor}(\texttt{d}_1) \wedge \texttt{closeGoal}(\texttt{d}_1)$ means that the robot can see a door d_1 that is closer to the goal than the robot is. It also sees two other markers in the room, r_{12} and r_{13} , one closer to the goal and the other farther from the goal, respectively.

Zaragoza and Morales (2009) transform the low-level data from the robot's sensor into a relational representation to define the set of states in the environment. Unlike the two approaches above presented, this approach was tested in a more complex simulated indoor environment. They first identify natural features of an indoor environment such as left discontinuity (l) and right discontinuity (r) (see Figure 2-b), wall (w) and obstacle (o). Each feature is represented by the tuple (DL, θL, T, A). DL and θL are shown in Figure 2-a, DL is the relative distance from the feature to the robot and θL is the relative orientation from the feature to the robot; T is the type of feature l, r, w or o, and A depend on the type of feature (if the feature is r or l, A is depth and if the feature is a w, A is its length). These features are used to activate some predicates such as *"place: in-room"* (see Figure 2-c), *"place:in-intersection"* (see Figure 2-d), *"place:in-corridor"* (see Figure 2-e), *"doors-detected: left"*, *"walls-detected: small"* and others. For example, the predicate *"walls-detected: front"* is activated when the robot identify a right discontinuity (r) followed by a left discontinuity (l) and the relative orientation θL of each discontinuity go from 22.5° to −22.5° (see Figure 2-b). All these predicates are used to define environment states.

2.2.2 SEMANTIC WEB RESOURCES

Even raw data in the web can be used in robotics (RIAZUELO et al., 2013; SAITO et al., 2011), but automatic understanding of such data is difficult. The KnowRob system (Knowledge Processing for Autonomous Personal Robots) is an important step towards enabling such understanding (TENORTH et al., 2011; TENORTH; BEETZ, 2013). KnowRob is a common semantic framework that combines knowledge representation and reasoning methods with techniques for acquiring knowledge, it contains a Semantic Web library that loads and accesses ontologies represented in OWL. The system is based on SWI Prolog [1] , resorting to description logics (BAADER, 2003) for knowledge representation, and to Bayesian networks (PEARL, 1988) for uncertain reasoning. The KnowRob knowledge base can either be queried from within the robot control program, from the command line or from a graphical user interface.

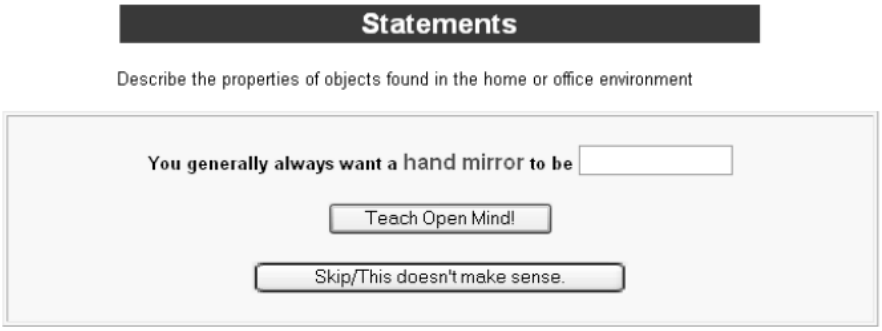
Other research projects are focused on saving commonsense information in relational databases. The Open Mind Common Sense, OMCS, (SINGH et al., 2002) is a system designed to acquire common sense knowledge from the general public over the web for reasoning about ordinary human life. This system is interested in acquiring facts, description and stories that the participants constructed in free-form natural language. For example, given a story such as "*Bob had a cold. Bob went to the doctor.*" the participant might continue the story filling the following sentences:

- *Bob was feeling sick;*
- *A doctor is a highly trained professional;*
- *Bob wanted to feel better.*

Another relational database project is the Commonsense data acquisition for indoor mobile robots ,OMICS, (GUPTA; KOCHENDERFER, 2004) that captures the knowledge through sentence templates generated dynamically. The greatest difference between the OMCS and OMICS project is that the OMICS project is only interested in collecting data from indoor home and office environment. The OMICS database is object-centric, i.e., all the statements in the database are related with objects and their properties. Since this work believes that everything that the robot knows about the environment and can do in it, is based on objects and their properties. In this work a statement is represented by the pair $\phi_{omics} = (o, p)$ where o is an object and p is an adjective that describe a object property. The statements can be assumed as assertions

[1] http://www.swi-prolog.org

Figure 3: Dynamic template used for the OMICS database to acquire information.



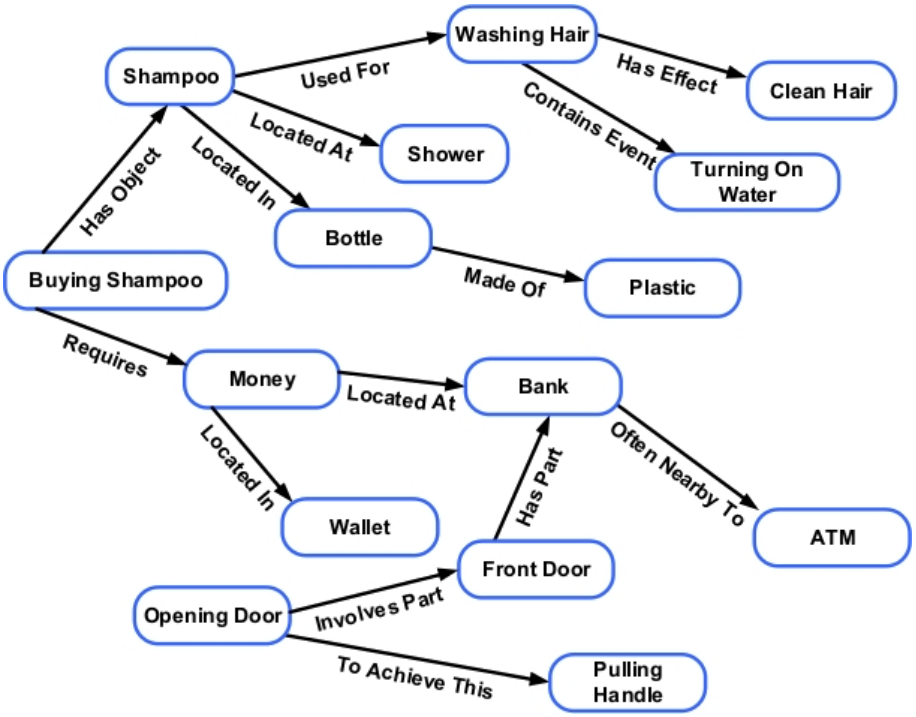
Source: Reproduced from (GUPTA; KOCHENDERFER, 2004).

about the property (or state) of an object in the environment or actions that can be performed on an object to allow a determinate effect, i.e., the statement *(cup-of-coffee,hot)* when the object o is the *cup-of-coffee* and the adjective p is *hot*, can indicate that "a cup of coffee is hot" or represent the action "make a cup of coffee hot". This approaches allow combining statement to produce common sense knowledge as: (o_1,p_1) causes (o_2,p_2), which can be: the statement (*vacuum,on*) cause (*room,clean*). This approach captured the common sense knowledge by using sentence templates (see Figure 3) that non-expert users filled.

In order to construct the OMICS database the system proposes to non-expert users different activities about object, tasks, causes, locations and others that allow capturing different types of knowledge. For example, in the location activities, the user is asked about the usual location of a particular object (e.g., "A room where you generally find a dinner table is the _____"). This work has 5804 submissions with the object activities and 3400 submissions with the location activities.

Several approaches have been proposed to convert the relational databases into representations that support reasoning. The ConceptNet (LIU; SINGH, 2004), LifeNet (SINGH; WILLIAMS, 2003) and the OMCSNet (LIU; SINGH, 2003) projects are tool-kits that process the natural language the general public entered in the relational OMCS databases. ConceptNet allows performing textual-reasoning tasks including topic generation, semantic disambiguation and other context-oriented inferences. LifeNet allows reasoning about human activities and the OMCSNet allows reasoning about the general common sense available in the OMCS database (see Figure 4). For example, OMCSNet can infer common sense knowledge as *"shampoo is located in a bottle that is made of plastic"*; *"shampoo is located at shower"*; *"shampoo is used for washing hair that has as effect a clean hair"*, and others.

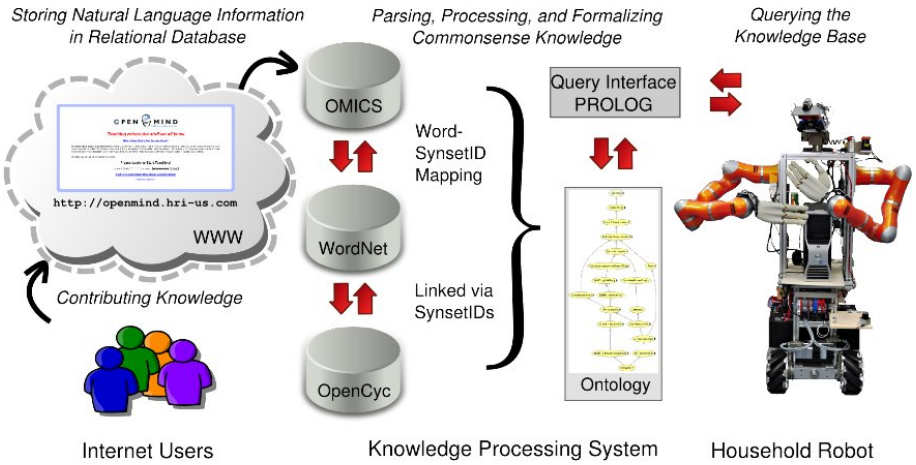
Figure 4: Example of the relational information available in the OMCSNet.



Source: Reproduced from (LIU; SINGH, 2003).

Kunze et al. (2010) integrate the commonsense knowledge from the large OMICS database into the KnowRob system by converting the database into a knowledge representation OWL languages for semantic web based on description logic. This work is summarized in Figure 5. This work takes the OMISC database that was constructed over the web by many non-expert users; the statements of the OMISC database are transformed in first-order representations by using natural language processing techniques such as part-of-speech tagging and syntax parsing that permit extract from the database complex relation as *problems(Task,Problem)*, i.e, two statements such as *"load the dishwasher"* and the related problem *"dishwasher is full"* can be transformed in *problems(dishwasher, load, dishwasher, full)*, the two first terms are the object and the action of the Task and the other terms are the object and the adjective of the Problem. This work uses the lexical database WordNet (MILLER, 1995), which allows grouping words into sets of cognitive synonyms and facing lexical ambiguities, i.e, the expressions such as *"turn on"*, *"switch on"* and *"start"* are mapped to the similar concept *TurningOnPoweredDevice*. These concepts are represented in a OWL-DL ontology based on OpenCyc's concept definitions (LENAT et al., 1990) that can be queried using the Prolog language. From this work is constructed the KnowRob_omisc ros library that allows obtaining probabilistic information of the common sense knowledge via Prolog. Currently, the KnowRob project is the most suitable semantic web applica-

Figure 5: Overview of the process used to integrate the OMICS database with the KnowRob system. Knowledge that is obtained from several Internet users is converted from natural language into a logical representation and can be queried by the robot control program.



Source: Reproduced from (KUNZE et al., 2010).

tion to reasoning about typical object locations within an indoor environment, taking into account the location uncertainty.

2.2.3 DISCUSSIONS

As seen in the previous section, there are several approaches related to the problem that we want to solve and some approaches that have important characteristics that can be used in the solution of our problem. The use of semantic information in a robot navigation can improve the robot's perception ability, by allowing the robot to deal with its sensors data. This semantic information can be processed from the semantic web. Although the semantic web provides a large amount of information, it is necessary to find an application that integrates the semantic information into representations that support reasoning. This representation is commonly modeled using a pure description logic and in a less common way using a probabilistic description logic. Usually, robots have little common sense knowledge and cannot conclude, for instance, that the detection of a refrigerator suggests the current location to be a kitchen. Pure logical reasoning is necessary, but not sufficient as sensors and actuators introduce considerable uncertainty into the process; hence, a probabilistic description logic is more suitable to our work. A framework we can use to process the semantic web information is the knowledge processing system KnowRob that combines knowledge representation with reasoning methods.

In order to facilitate the combination of different sources of information, a relational representation of the environment must be adopted. This representation allows

a higher level of abstraction, which provides knowledge generalization and allows knowledge transfer. All these characteristics permit to combine the information more easily.

Following chapters present our contributions. In Chapter 3 we contribute a new framework for robot navigation called Web-based Relational Robotic Architecture (WRRA) and in Chapter 4 we describe the implementation and tests of the framework.

3 SEMANTIC ROBOT NAVIGATION WITH PROBABILISTIC RELATIONAL REASONING

In this chapter we are going to present a new framework for semantic robot navigation under situation and perception uncertainty. Furthermore, we are going to describe each of the modules that has our new framework.

3.1 A FRAMEWORK FOR ROBOT NAVIGATION

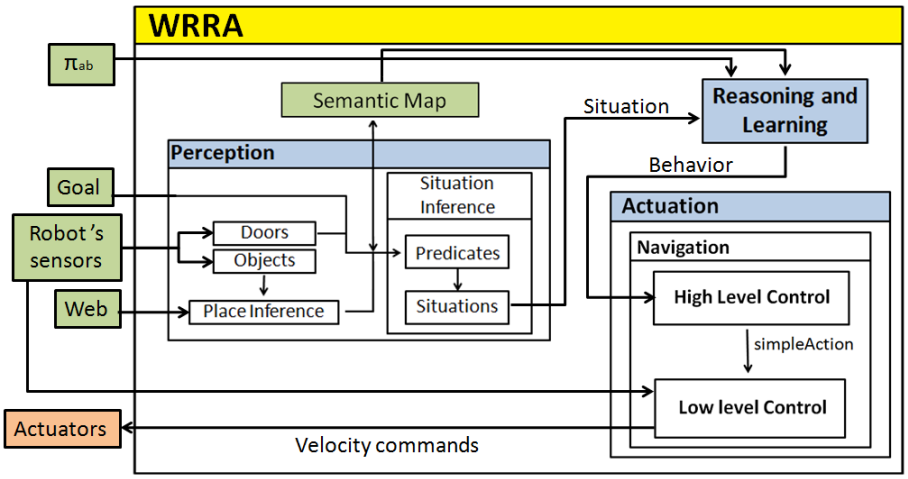
We consider a robot with three important sources of information. First, sensors that detect objects in the environment. Second, a map with semantic information, which is updated during navigation. Third, a web database with commonsense knowledge (for example, likely location of objects in rooms). In our framework, information processing flows through three major modules: Perception, Reasoning and Learning, and Actuation. We call the whole architecture *Web-based Relational Robotic Architecture* (WRRA), as depicted in Figure 6. From the perspective of uncertain reasoning with semantic web resources, the Perception module (Section 3.1.1) is the most significant contribution of this work.

3.1.1 PERCEPTION: SEMANTIC INFORMATION AND PROBABILISTIC REASONING

Our robot is equipped with one laser rangefinder, three semantic cameras, and two odometers on its two front wheels. The laser rangefinder uses a laser beam to determine the distance to the obstacles; a semantic cameras is camera provided by the MORSE simulator that return a semantic label for each object or door detected by it. One semantic camera is configured to detect only doors and the other two semantic cameras are configured to detect objects; and the odometers return the distance traveled by the robot.

This module receives a description of the *goal*, which is to find a *target room* inside

Figure 6: Overview of the Web-based Relational Robotic Architecture.



Source: Author.

a house. The Perception module receives sensory information (detected objects), and must abstract the data into compact symbolic representations. Upon receiving data, the robot accesses its Semantic Web resources to determine the most likely room that generated the data, as described in this section.

Unlike most existing robotic systems, we pursue reasoning at a high level of abstraction, employing concepts, roles and relations between them as expressed within KnowRob. Due to this decision, we can effectively employ semantic web resources. The *situation* of the robot is described in terms of a relational representation that not only allows for abstraction of metric and sensory details, but also enables knowledge to be generalized and reused in new tasks. During navigation, the robot uses sensor data to build a relational representation of the environment (the semantic map).

The output of the Perception module is a description of the robot's situation, which is specified by a conjunction of active predicates (with truth value TRUE). We grouped the predicates into three sets that are shown in Table 1. The three sets are P_{LI} that contains predicates about the location inferred using common sense knowledge; P_{SO} are the predicates obtained only using sensor observations and P_{SM} are the predicates inferred from the semantic map. Here we describe them all.

- P_{LI}
  - `inTargetRoom`() indicates that the target room is where the robot is;
  - `nonTargetRoom`(`P`) means that the robot is in `P` and it is not the target room;

  - `inRoom(P)` indicates that `P` is the most likely room for the robot to be;
  - `inUnknownRoom()` indicates that the robot is in a room that has not yet been identified;

- P_{SO}
  - `seeDoors()` indicates that the robot sees one or more doors from the position where it is in the room;
  - `seeNoDoors()` means that the robot does not see doors from the position where it is in the room;
  - `seeNextPlace(P)` indicates that the robot is in front of a door (1m away) and can even see a room `P` through it. However, the robot will only be able to infer what type of room is `P`, when the robot enters `P`;

- P_{SM}
  - `seeNonVisitedDoor(D)` means that the robot sees door `D` that has not yet been visited;
  - `seeVisitedDoor(D)` indicates that the robot sees door `D` that has already been visited;
  - `allDoorsVisited()` means that all doors in the room have already been visited;
  - `notAllDoorsVisited()` means that all doors in the room have not yet been visited;
  - `notAllDoorsVisitedInNextPlace(P)` means that all doors in the next room `P` have not yet been visited;
  - `allDoorsVisitedInNextPlace(P)` means that all doors in the next room `P` have already been visited;

The truth values of P_{LI} predicates are computed by using *Place Inference* block, as we explain next.

The Perception module is heavily based on reasoning facilities available in the KnowRob package. The knowledge base in KnowRob uses Resource Description Framework (RDF) triples to model semantic information and to represent a large ontology. RDF is a graph-based data model with labeled nodes and directed, labeled edges. The most basic entity of a RDF is the statement, which represents an edge in the graph and it is formed by three components: a subject, a predicate, and an object. The subject

Table 1: Situation predicates proposed in our framework, where P is a room label and D is a door label .

Situation predicates		
P_{LI}	P_{SM}	P_{SO}
inRoom(P)	seeNonVisitedDoor(D)	seeDoors()
inUnknownRoom()	seeVisitedDoor(D)	seeNoDoors()
inTargetRoom()	allDoorsVisited()	seeNextPlace(P)
nonTargetRoom(P)	notAllDoorsVisited()	
	notAllDoorsVisitedInNextPlace(P)	
	allDoorsVisitedInNextPlace(P)	

Source: Author.

is source of the edge; the object is the target of the edge; and the predicate determine the type of relationship between the subject and object. All the three components can be identified by a URI.

Just as an example of RDF triples in the knowledge base, consider the fact, extracted from the OMICS database, that a kitchen contains a refrigerator (`XXX` denotes the string `http://ias.cs.tum.edu/kb/knowrob.owl`):

```
<rdf:Description rdf:about="XXX#OmicsLocations-1">
    <ns1:object rdf:resource="XXX#Kitchen"></ns1:object>
    <ns1:subject rdf:resource="XXX#Refrigerator"></ns1:subject>
    <rdf:type rdf:resource="XXX#OmicsLocations"></rdf:type>
</rdf:Description>
```

In the above example the subelement "rdf:Description" is used to describe three property subelements. Therefore, the same number of statements are encoded. For each statement the subject is given by the "rdf:about" and the three objects of the statements are "#Kitchen", "#Refrigerator" and "#OmicsLocations" and the predicates are "object", "subject" and "type" respectively. Thus, the first statement is that the "#OmicsLocations-1" has a object "#Kitchen", the second statement is that the "#OmicsLocations-1" has a subject "#Refrigerator" and the third statement is that the "#OmicsLocations-1" is the type "#OmicsLocations".

By itself, RDF is just a data model that does not have any significant semantics. Therefore Web Ontology Language OWL that is a set of RDF triples, is used to define a vocabulary that can be use with the RDF model and that allows indicating relationships such as Refrigerator, a subclass of CoolingDevice, or Kitchen, a subclass of RoomInAConstruction (TENORTH; BEETZ, 2013).

The following OWL sentences are examples from the base ontology used by KnowRob and stored in <http://knowrob.org/kb/knowrob.owl>. Accessed: June 2014:

```
<owl:Class rdf:about="&knowrob;PartOfABuilding">
   <rdfs:subClassOf rdf:resource="&knowrob;PartOfAnHSC"/>
</owl:Class>

<owl:Class rdf:about="&knowrob;RoomInAConstruction">
   <rdfs:subClassOf rdf:resource="&knowrob;PartOfABuilding"/>
   <rdfs:subClassOf rdf:resource="&knowrob;SpaceInAFixedHOC"/>
</owl:Class>

<owl:Class rdf:about="&knowrob;Kitchen">
   <rdfs:subClassOf rdf:resource="&knowrob;RoomInAConstruction"/>
</owl:Class>

<owl:Class rdf:about="&knowrob;BedRoom">
   <rdfs:subClassOf rdf:resource="&knowrob;RoomInAConstruction"/>
</owl:Class>

<owl:Class rdf:about="&knowrob;Refrigerator">
   <rdfs:subClassOf rdf:resource="&knowrob;Box-Container"/>
   <rdfs:subClassOf rdf:resource="&knowrob;ElectricalHouseholdAppliance"/>
   <rdfs:subClassOf rdf:resource="&knowrob;RefrigeratedStorageDevice"/>
</owl:Class>
```

The knowledge stored in OWL is loaded into the logic programming language Prolog, and is accessed via Prolog predicates. Prolog allows attaching probabilities to the knowledge and allows inferring new knowledge using Bayesian networks.

The following Prolog predicates are part of the Prolog module called *OMICS* that KnowRob uses to attach probabilities to the knowledge obtained from the OMICS database:

```
:- module(omics,
          [
           probability_given/4,
           bayes_probability_given/4,
           probableObjectInRoom/2,
           allProbableObjectInRoomAboveThreshold/3,
           probableLocationOfObject/2,
           allProbableLocationOfObject/2,
           mostProbableLocationOfObject/2,
```

```
        allProbableLocInstancesOfObject/2
        ]).
```

the above PROLOG predicates receive a number of data that depends of the arity of the predicate. These data can be input information of output information. For example the predicate $bayes_probability_given/4$ is equal to $bayes_probability_given(Type, Obj, Subj, P)$, where $Type$ and $Subj$ are inputs and Obj and P are outputs.

The Perception module queries KNOWROB, which returns, for each observed object, the probability that the location is each possible room, given the observed object. Queries are sent to KNOWROB through PROLOG sentences via function calls in the Python language; as an example, consider (a complete query is given in Section 4.1):

```
for obj in DetectedObjects:
    q="bayes_probability_given(knowrob:'OmicsLocations',
          Room,knowrob:'"+obj+"',Pr)"
    query = prolog.query(q)
```

Such a query returns probabilities such as:

```
Room = 'knowrob.owl#Kitchen'
Pr = 0.1031101853182014 ;
```

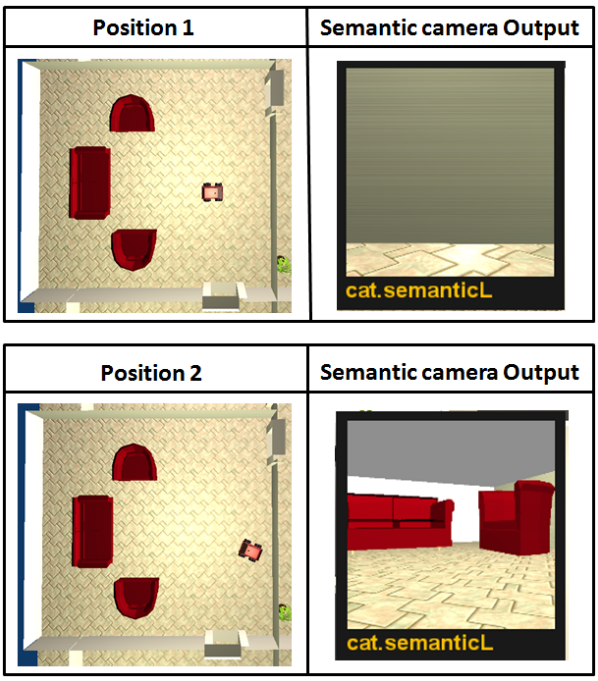
That is, given a perceived object o_i, KNOWROB uses inference with its probabilistic description logic (LUKASIEWICZ; STRACCIA, 2008; LUKASIEWICZ, 2008) to return $P(r_j|o_i)$ to each room r_j, where $P(r_j|o_i)$ is the probability of r_j given o_i. The problem now is to combine these pieces of information into a probability that the robot is in room r_j, given *all* detected objects $o_1, \ldots, o_n$. We have:

$$\begin{aligned} P(r_j|o_1,\ldots,o_n) &= \frac{P(o_1,\ldots,o_n|r_j)P(r_j)}{P(o_1,\ldots,o_n)} \\ &= \frac{P(o_1|r_j,o_2,\ldots,o_n)P(o_2|r_j,o_3,\ldots,o_n)\ldots P(o_1|r_j)P(r_j)}{P(o_1,\ldots,o_n)}. \end{aligned}$$

We now assume that, given r_j, an observation (of an object) is independent of other observations (of other objects in the same room). Hence:

$$\begin{aligned} P(r_j|o_1,\ldots,o_n) &= \frac{P(o_1|r_j)P(o_2|r_j)\ldots P(o_1|r_j)P(r_j)}{P(o_1,\ldots,o_n)} \\ &= \frac{(P(r_j|o_1)P(o_1)/P(r_j))\ldots(P(r_j|o_n)P(o_n)/P(r_j))P(r_j)}{P(o_1,\ldots,o_n)} \end{aligned}$$

Figure 7: Two possible robot's position within the environment and its respective semantic camera output



Source: Author.

$$= \left(\prod_{i=1}^{n} P(r_j|o_i) \right) \frac{\prod_{i=1}^{n} P(o_i)}{P(o_1,\ldots,o_n)(P(r_j))^{n-1}}. \tag{3.1}$$

We now introduce a substantive assumption, namely, that every room has identical a priori probability $P(r_j)$. Thus, $P(r_j|o_1,\ldots,o_n)$ is proportional to $\prod_{i=1}^{n} P(r_j|o_i)$. Once the Perception module gets, for each room, each term of this product from KnowRob, it compares each room with respect to this product, setting the truth value of `inRoom(P)` as:

$$\texttt{inRoom(P)} = \begin{cases} true & if \ \ \texttt{P} = \arg\max_{r_j} \prod_{i=1}^{n} P(r_j|o_i), \\ false & \text{otherwise.} \end{cases} \tag{3.2}$$

Although *Place Inference* block can infer the truth value of the P_{LI} predicate `inRoom(P)` using Equation 3.2, there are robot's positions within the environment in which `inRoom(P)` predicate is impossible of inferring, since the robot cannot perceive any object. Therefore *Place Inference* block activates the P_{LI} predicate `inUnknownRoom()`. Figure 7 shows two robot's positions within an indoor environment. Initially the robot does not perceive any object from position 1 and the `inUnknownRoom()` predicate is set with TRUE and then the robot perceives some objects from position 2 and the `inRoom(P)` is set with TRUE being $P = livingroom$.

The *Place Inference* block output is compared with the goal input. This comparison allows setting with the truth value the P_{LI} predicates $inTargetRoom()$ and $nonTargetRoom(P)$. If the *Place Inference* block output is equal to the goal input, the predicate $inTargetRoom()$ is set with TRUE, otherwise the $nonTargetRoom(P)$ predicate is set with TRUE.

Additionally, the truth values of P_{SO} predicates are computed by using *Doors* block, as follows.

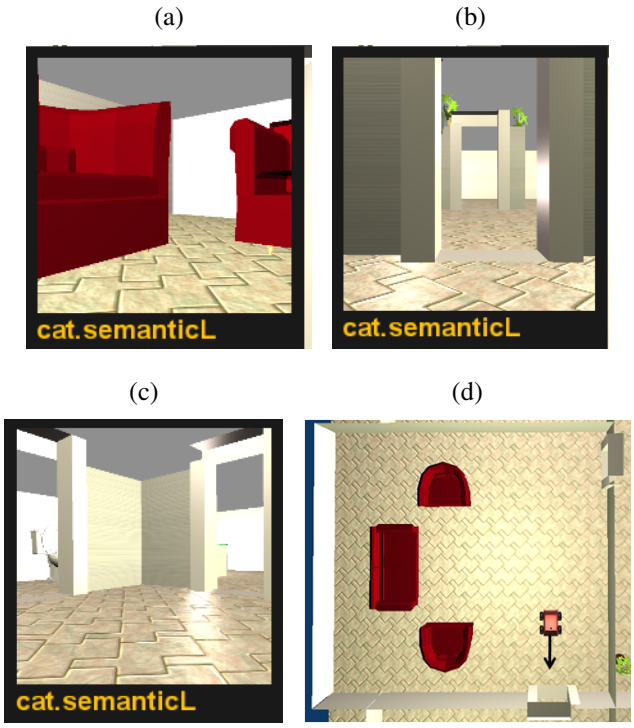
Doors block takes all the doors perceived by the semantic camera sensor, but for computing the P_{SO} predicates $seeDoors()$, $seeNoDoors$, and $seeNextPlace(P)$ it uses the doors perceived within the same room, e.g. when the robot sees several doors, one behind the other (see Figure 8b), the robot determines which door is in the same room where the robot is and the other doors are not taken into account. To know which door is in the same room as the robot is and which is not, we compute the distance between the robot and the doors. The door with the shortest distance to the robot is in the robot's room. If the robot detects one or more doors within the room, the $seeDoors()$ predicate is set with TRUE (see Figure 8c), otherwise the $seeNoDoors$ is set with TRUE (see Figure 8a). Another situation is when the robot is in front of a door and 1 meter away (see Figure 8d); in this case, the $seeDoors()$ predicate is set with TRUE and additionally, the $seeNextPlace(P)$ predicate is also set with TRUE.

Finally, the truth values of P_{SM} predicates are computed by using *Semantic Map* block, as explained next.

During navigation, a semantic map of the environment is created. Each visited room and each observed object are represented as vertices of a graph that describes the topological map (left side of Figure 9). Connectivity between rooms is represented by graph edges, which are defined through doors connecting the rooms. While this topological map is built, edges are created by connecting vertices of the topological map to vertices of the conceptual map (right side of Figure 9). Unlike other approaches (AYDEMIR et al., 2013; GALINDO; SAFFIOTTI, 2013), our map does not involve metric representation of the environment. Every inference and reasoning in WRRA occurs at the level of objects, rooms and relationships and properties thereof.

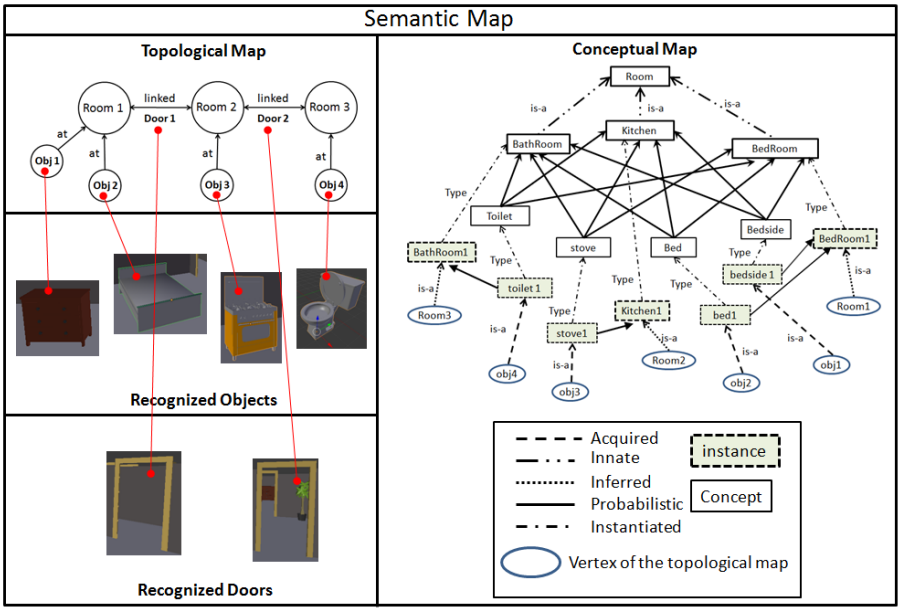
While *Semantic Map* block takes information that comes from the *Doors* block and *Place Inference* block to build a Semantic Map of the indoor environment, the robot uses the Semantic Map information to infer the P_{SM} predicates $seeNonVisitedDoor(D)$, $seeVisitedDoor(D)$, $allDoorsVisited()$, $notAllDoorsVisited()$, $notAllDoorsVisitedInNextPlace(P)$ and $allDoorsVisitedInNextPlace(P)$.

Figure 8: Figures 8a, 8b, 8c are semantic camera outputs for three possible robot's position within the environment, and Figure 8d is a robot's position when is near to a door. The arrow in Figure 8d indicates the direction, in which the robot sees.



Source: Author.

Figure 9: The semantic map built by WRRA.

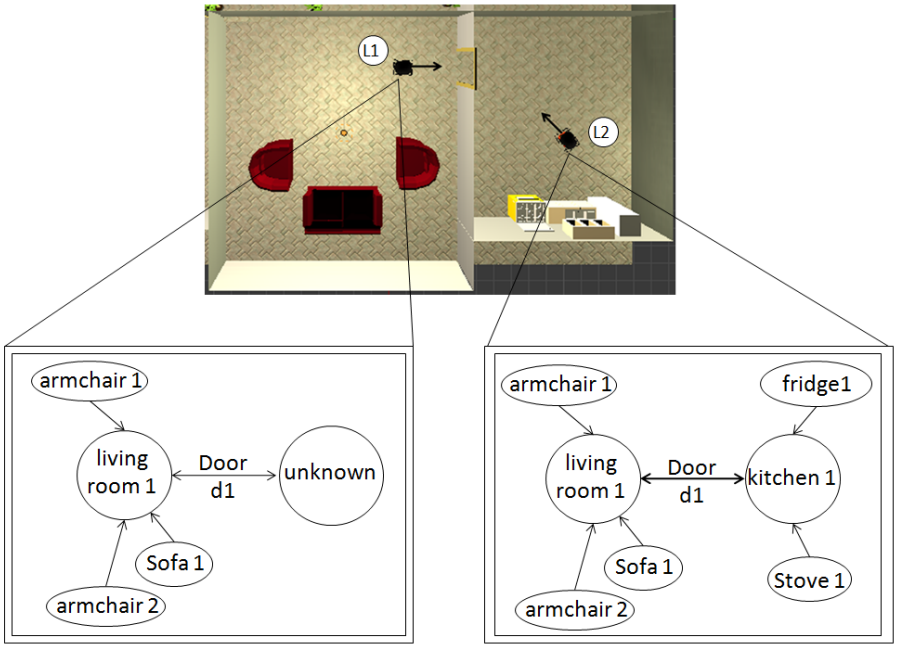


Source: Author.

Let us suppose that the robot is in location (L1) shown on the upper left side of Figure 10. Additionally, let us suppose that the robot shown in location (L1) has only recognized the room together with the objects and the doors present in that room; besides the result of the recognition, the Semantic map is shown on the lower left side of Figure 10. In this case, predicate $seeNonVisitedDoor(d_1)$ is set with TRUE, since the robot can see the door $\mathtt{d}_1$, but the robot has not gone through $\mathtt{d}_1$. The predicate $notAllDoorsVisited()$ is also set with TRUE, because in the room where the robot is, one door was found, but this door has not been visited, e.g. predicate $notAllDoorsVisited()$ compares the number of doors recognized with the number of doors visited; if there are more doors recognized than doors visited, the predicates is set with TRUE, otherwise the predicate is set with FALSE. Additionally, predicate $notAllDoorsVisitedInNextPlace(p_2)$ is set with TRUE, since although the robot knows door $\mathtt{d}_1$ that connects room $\mathtt{p}_1$ with room $\mathtt{p}_2$, the robot does not yet know if room $\mathtt{p}_2$ has more doors than that already recognized.

Another different case is when the robot is in location (L2) shown on the upper right side of Figure 10. In this location the robot has already recognized all about the two rooms $\mathtt{p}_1$ and $\mathtt{p}_2$, and the result of the recognition is the semantic map shown on the lower right side of Figure 10. In this case, predicate $seeVisitedDoor(d_1)$ is set with TRUE, because the robot already went through the door $\mathtt{d}_1$, and it knows the type of room there is on each side of door $\mathtt{d}_1$. Predicate $allDoorsVisited()$ is also set with TRUE, since the only door $\mathtt{d}_1$ in room $\mathtt{p}_2$ the robot has already visited. Another

Figure 10: Two possible robot's position (L1 and L2) are shown with its respective semantic maps after exploration.



Source: Author.

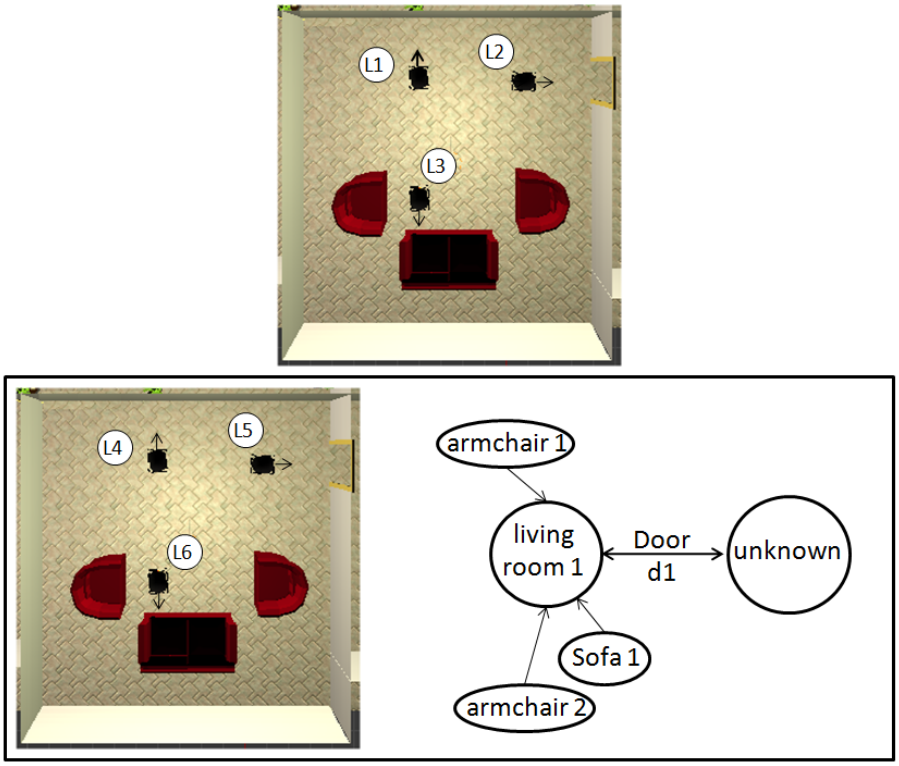
predicate set with TRUE is predicate $allDoorVisitedInNextPlace(p_1)$, because the robot knows from the semantic map that room $\mathtt{p}_1$ has only one door $\mathtt{d}_1$, and this the robot has already visited.

When the Perception module has already inferred all the predicates from the position in which the robot lies, the *Situation Inference* block computes the Perception module output by taking all the TRUE predicates which are then grouped in a set to determine a ground situation. In logic programming a ground situation is a set of ground atoms or ground predicates, where a ground predicate is formed by a predicate symbol together with its terms, and the terms are represented with constants. For example, predicate $seeVisitedDoor(d_1)$ is a ground predicate, better explained in section 3.1.2. Figure 11 shows 6 possible robot's locations within an indoor environment and we show below the ground situation inferred for each location.

Let us suppose that locations L1, L2 and L3 shown at the top of Figure 11 represent 3 robots in 3 possible initial position; the target room is the bathroom; the robots do not have yet a semantic map of the environment and the robot in L1 does not perceive any object, the robot in L2 perceives a door d_1 and the robot in L2 perceives two objects a sofa o_1 and armchair o_2. The ground situation for these three locations are:

- $s_1 = \mathtt{inUnknownRoom}() \wedge \mathtt{nonTargetRoom}(\mathtt{unknown1})$

Figure 11: Six locations of a robot within an indoor environment, the locations L1, L2, L3 represent 3 robots in 3 possible initial positions and the locations L4, L5, L6 represent 3 robots in 3 possible positions that the robot takes after having explored the environment, Additionally, it is shown a semantic map obtained from the exploration of the robots in L4, L5 and L6.



Source: Author.

$\wedge$ notAllDoorsVisited() $\wedge$ seeNoDoors().

- s_2 = inUnknownRoom() $\wedge$ nonTargetRoom(unknown1) $\wedge$ notAllDoorsVisited() $\wedge$ seeDoors() $\wedge$ seeNextPlace(unknown2) $\wedge$ seeNonVisitedDoor(d_1) $\wedge$ notAllDoorsVisitedInNextPlace(unknown2).

- s_3 = inRoom(living − Room) $\wedge$ nonTargetRoom(living − Room) $\wedge$ notAllDoorsVisited() $\wedge$ seeNoDoors().

For example, the ground situation s_2 means that the robot is in an unknown place that is not the target room, the robot can observe the door d_1, which has not yet been visited and the robot knows that all doors in its room have not yet been visited. Additionally, the robot can see the next place through the door d_1, but the robot cannot identify the place. Finally, the robot knows that the place through the door d_1 possibly has more doors without visiting.

Let us now suppose that locations L4, L5 and L6 shown on the lower left side of Figure 11 represent 3 robots in 3 possible positions that the robot takes after having

made a recognition of the room and the result of the recognition is the semantic map shown on the lower right side of Figure 11. Additionally, the target is yet the bathroom and the robot in L4 does not see objects, the robot in L5 sees the door d_1 and the robot in L6 sees the objects sofa o_1 and armchair o_2. The ground situation for these three locations are:

- $s_4 = \texttt{inRoom}(\texttt{living}-\texttt{Room}) \wedge \texttt{nonTargetRoom}(\texttt{living}-\texttt{Room})$ $\wedge\, \texttt{notAllDoorsVisited}() \wedge \texttt{seeNoDoors}()$.
- $s_5 = \texttt{inRoom}(\texttt{living}-\texttt{Room}) \wedge \texttt{nonTargetRoom}(\texttt{living}-\texttt{Room})$ $\wedge\, \texttt{notAllDoorsVisited}() \wedge \texttt{seeDoors}() \wedge \texttt{seeNextPlace}(\texttt{unknown2})$ $\wedge\, \texttt{seeNonVisitedDoor}(\texttt{d}_1) \wedge \texttt{notAllDoorsVisitedInNextPlace}(\texttt{unknown2})$.
- $s_6 = \texttt{inRoom}(\texttt{living}-\texttt{Room}) \wedge \texttt{nonTargetRoom}(\texttt{living}-\texttt{Room})$ $\wedge\, \texttt{notAllDoorsVisited}() \wedge \texttt{seeNoDoors}()$.

3.1.2 REASONING AND LEARNING

The WRRA uses reinforcement learning (RL) to refine behavior through interactions with the environment (SUTTON; BARTO, 1998). Typical RL solutions learn from scratch; we instead employ two levels of RL (KOGA et al., 2013), where an abstract and a ground policy are learned simultaneously. The stochastic abstract policy learned in a source task is then used in new similar tasks.

3.1.2.1 MODELING

Our robot navigation problem is modeled as a Relational Markov Decision Process (RMDP), which is an extension of the Markov Decision Process (MDP) formal model (VAN OTTERLO, 2009). Unlike the MDP formalism, the RMDP model uses a *relational vocabulary* Σ to describe situations and behaviors. A relational vocabulary is defined by a set of constants C and predicates P_S and P_A such as $\Sigma = C \cup P_S \cup P_A$. The set of *constants* C represents the objects of the environment; the set of *predicates* P_S describes object properties and relations among them; and the set of *predicates* P_A represents behaviors (KOGA et al., 2013). For this work, the relational vocabulary is assumed finite, with n_C constants, n_S predicates in P_S and n_A predicates in P_A.

To better understand the RMDP, some concepts are necessary. If $t_1, ..., t_n$ are terms, which can be constants (lower-case letter) or variables (capital letter), besides p/n being a predicate symbol p with arity n ≥ 0, then p($t_1, ..., t_n$) is an atom. When an atom does

not contain any variable, it is called a ground atom. One more important concept is Herbrand base HB_Σ, which is the set of all possible ground atoms formed with the n_S predicates and n_C constants in a relational vocabulary Σ. Finally, an RMDP is defined as a tuple $\langle \Sigma, \mathcal{B}, \mathcal{S}, \mathcal{A}, T, R, \mathcal{G}, b^0 \rangle$, where:

- Σ is the finite relational vocabulary that specifies situations and behaviors as indicated below;
- $\mathcal{B}$ is the knowledge base of sentences in first-order logic, which defines constraints for the combinations of predicates and objects and it is specified by Σ;
- $\mathcal{S}$ is the set of (ground) situations, beside it is the set of all complete truth assignments for conjunctions of atoms in the Herbrand base $HB_{P_S \cup C}$ satisfying $\mathcal{B}$;
- $\mathcal{A}$ is the set of behaviors that is a subset of $HB_{P_A \cup C}$ satisfying $\mathcal{B}$;
- $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ is a *transition function* such that $T(s,a,s') = P(s_{t+1} = s' | s_t = s, a_t = a)$ is the probability of reaching situation s' at time $t+1$ when the agent is in situation s and executes behavior a at time t;
- $R : \mathcal{S} \rightarrow \mathbb{R}$ is a *reward function*, such that $r_t = R(s_t)$ is the reward received when the agent is in situation s at time t;
- $\mathcal{G} \subset \mathcal{S}$ is a set of goal situations. No transition leaves any goal situation, i.e., $T(s,a,s') = 0, \forall s \in \mathcal{G}, \forall a \in \mathcal{A}, \forall s' \in \mathcal{S}$;
- $b^0 : \mathcal{S} \rightarrow [0,1]$ is the initial situation probability distribution, such that $b^0(s)$ is the probability of situation s being the first one in an episode, i.e., $P(s_0 = s) = b^0(s)$.

In a RMDP, situations $s \in \mathcal{S}$ are represented as a conjunction of predicates describing properties of and relations among objects, such as: $s_1 = \texttt{inRoom}(\texttt{livingroom}) \wedge \texttt{nonTargetRoom}(\texttt{livingroom}) \wedge \texttt{seeNoDoors}() \wedge \texttt{notAllDoorsVisited}()$. Other formalisms are possible to represent decisions and transitions (MATEUS et al., 2001).

A conjunction is a *ground conjunction* if it contains only ground atoms (such as s_1 given in the example). In our discussion, each variable in a conjunction is implicitly assumed to be existentially quantified. An *abstract situation* σ (and *abstract behavior* α) is a conjunction with no ground atom. A relational representation enables us to aggregate situations and behaviors by using variables instead of constants in the predicate terms. For example, ground situation s_1 is covered by abstract situation σ by replacing *livingroom* with variable X; in this case, other situations could also be covered

by σ, e.g., s_2 = `inRoom(kitchen)` $\wedge$ `nonTargetRoom(kitchen)` $\wedge$ `seeNoDoors()` $\wedge$ `notAllDoorsVisited()`.

Denote by $\mathcal{S}_\sigma$ the set of ground situations $s \in \mathcal{S}$ covered by abstract situation σ. We assume that each ground situation s is abstracted to only one abstract situation σ. Similarly, we define $\mathcal{A}_\alpha(s)$ as the set of all ground behaviors $a \in \mathcal{A}$ covered by an abstract behavior α in ground situation s. We also define $\mathcal{S}_{ab}$ and $\mathcal{A}_{ab}$ as the set of all abstract situations and the set of all abstract behaviors in an RMDP, respectively. To simplify notation, we here use the assumption that if an atom does not appear in a ground sentence, the negated atom is assumed.

Our behaviors P_A are described by 4 predicates, which are shown in Table 2. Here we explain each of them:

- `goToDoor(D)` meaning that the robot should go to `D`; this behavior is available in situations with `seeNonVisitedDoor(D)` activated;
- `goToNextRoom(P)` meaning that the robot should go into the next place `P`; this behavior is available in situations with `seeNonVisitedDoor(D)` and `seeNextPlace(P)` activated or with `notAllDoorsVisitedInNextPlace(P)` activated;
- `findDoor()` meaning that the robot should search for doors in the room; this behavior is available in situations with `seeNoDoors()` activated;
- `exploration()` meaning that the robot should search for objects in the room; this behavior is available in situations with `inUnknownRoom()` activated.

Table 2: Behavior predicates proposed in our framework, where P is a room label and D is a door label .

Behavior predicates	
`goToDoor(D)`	`findDoor()`
`goToNextRoom(P)`	`exploration()`

Source: Author.

3.1.2.2 SOLVING THE RMDP

To solve an RMDP is to find an *optimal policy* π^* that maximizes a function R_t of future rewards. In Reinforcement Learning (RL) tasks the agent does not know the dynamics of the process and a series of RL algorithms can be used to find a policy (SUTTON; BARTO, 1998). In RL, the robot uses its experiences with the environment

to learn the policy. The robot performs trial-and-error interactions with its environment through its perception and action. During the interactions, the robot receives some information about its current environment situation s; the robot performs a behavior a that changes the environment situation and then the robot receives a numeric reward R, which it tries to maximize over time.

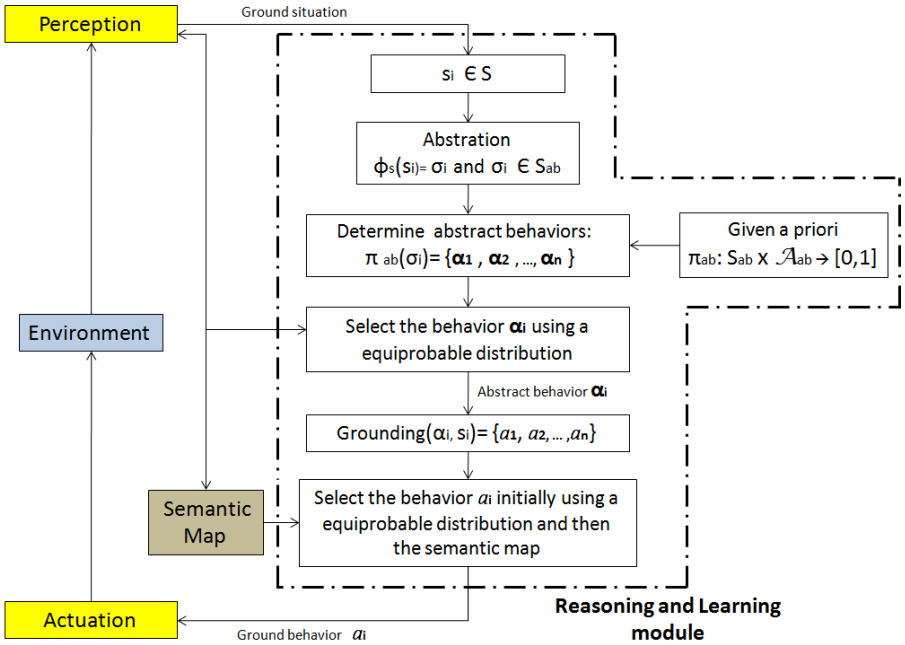
We propose the following scheme to apply an abstract policy in a ground problem (see Figure 12). To begin, we need to define two operations: *abstraction* and *grounding*. Abstraction is the translation from the ground level (perceived by the robot's sensors) to the abstract level by replacing constants with variables, $\phi_s : \mathcal{S} \rightarrow \mathcal{S}_{ab}$. For a ground situation s, the corresponding abstract situation σ is given by $\phi_s(s) = \sigma$ so that $s \in \mathcal{S}_\sigma$. An example is shown in Example 1:

Example 1 Consider the ground situations s_3 = `inRoom(bathroom)` $\wedge$ `nonTargetRoom(bathroom)` $\wedge$ `seeDoors()` $\wedge$ `seeNonVisitedDoor(d`$_1$`)`; and s_4 = `inRoom(livingroom)` $\wedge$ `nonTargetRoom(livingroom)` $\wedge$ `seeDoors()` $\wedge$ `seeNonVisitedDoor(d`$_2$`)`; and s_5 = `inRoom(diningroom)` $\wedge$ `nonTargetRoom(diningroom)` $\wedge$ `seeDoors()` $\wedge$ `seeNonVisitedDoor(d`$_1$`)`; Their respective abstract situation is: σ_1 = `inRoom(P)` $\wedge$ `nonTargetRoom(P)` $\wedge$ `seeDoors()` $\wedge$ `seeNonVisitedDoor(D)`; and therefore $s_3, s_4, s_5 \in \mathcal{S}_{\sigma_1}$.

Grounding is the translation from the abstract level to the ground level, $a = grounding(\alpha, s)$. Clearly only ground states are sensed and visited by the robot, and only ground actions can be actually applied. Learning and reasoning must proceed by processing, at time t, the (ground) experience $\langle s_t, a_t, r_t, s_{t+1}, a_{t+1} \rangle$, which is related to the tuple $\langle \sigma_t, \alpha_t, r_t, \sigma_{t+1}, \alpha_{t+1} \rangle$.

Consider now that a stochastic abstract policy is defined as $\pi_{ab} : \mathcal{S}_{ab} \times \mathcal{A}_{ab} \rightarrow [0, 1]$. After the abstract situation $\sigma = \phi_s(s)$ is derived from the observed ground situation s (see Figure 12), we start up our reasoning and learning process using a stochastic abstract policy tuned and generalized that yields probabilities $\pi_{ab}(\sigma, \alpha_k) = P(\alpha_k|\sigma)$ for all $\alpha_k \in \mathcal{A}_{ab}$, which can be obtained from different approaches such as those presented by Matos et al. (2011) that propose the inductive algorithm ND-TILDE, which produces nondeterministic abstract policies represented by a first order logical decision tree, which is induced from several experiences using a optimal policy; or the approach presented by Koga et al. (2013) that proposes the algorithm AbsProb-RL, which produce a stochastic abstract policy through direct search on the space of policies. Given the stochastic abstract policy, for each abstract situation σ we can have more than one abstract behavior α, thus we select an abstract behavior $\alpha_k \in \mathcal{A}_{ab}$ according to the proba-

Figure 12: Overview of our reasoning and learning module.



Source: Author.

bilities yield by the abstract policy. Then, the grounding operation $a = grounding(\alpha_k, s)$ is executed and it is possible that given the abstract behavior α_k for the ground situation s more than one ground behavior a may be possible. Therefore, a ground behavior is randomly selected. As the robot's knowledge about the new environment increases, due to its perception and action in the environment, the robot creates and improves a semantic map, which places restrictions on the actions defined by the policy initially received (see Figure 12), adapting it to the new environment and to the new task, which allows transforming a policy that is initially stochastic and abstract into a deterministic and ground policy.

For example, considering the robot in the location L1 shown in Figure 13. Let us suppose that L1 is the initial position of the robot, in which the robot can only observe two door d_1 and d_2 and it does not know anything about the new environment. Following the scheme proposed for our Reasoning and Learning module shown in Figure 12, the ground situation received from the Perception module is:

$s_1 = \texttt{inUnknownRoom}() \wedge \texttt{nonTargetRoom}(\texttt{unknown}) \wedge \texttt{notAllDoorsVisited}() \wedge \texttt{seeDoors}() \wedge \texttt{seeNonVisitedDoor}(\texttt{d}_1) \wedge \texttt{seeNonVisitedDoor}(\texttt{d}_2)$.

The abstract situation $\sigma_1 = \phi_s(s_1)$ is:

$\sigma_1 = \texttt{inUnknownRoom}() \wedge \texttt{nonTargetRoom}(\texttt{P}) \wedge \texttt{notAllDoorsVisited}() \wedge \texttt{seeDoors}() \wedge \texttt{seeNonVisitedDoor}(\texttt{D}) \wedge \texttt{seeNonVisitedDoor}(\texttt{D})$.

For this abstract situation two abstract behaviors are possible: α_1 = `exploration`() and α_2 = `goToDoor`(`D`) both with probability of 1/2. Let us suppose that is randomly selected the abstract behavior α_2 = `goToDoor`(`D`). Then the grounding operation $\{a_1, a_2\} = grounding(\alpha_2, s_1)$ is executed and two ground behaviors are obtained for the ground situation s_1: the ground behavior `goTodoor`($\mathtt{d}_1$) and the ground behavior `goTodoor`($\mathtt{d}_2$). Let us suppose that the robot randomly selects the ground behavior `goTodoor`($\mathtt{d}_2$), which is sent to the Actuation module.

Consider now the robot in the location L2 shown in Figure 13 and let us suppose that the robot takes the position in the location L2 after having explored the environment, and constructed the semantic map shown on the lower right side of Figure 13, the ground situation received from the Perception module is:

s_2 = `inRoom`(`hallway1`) $\wedge$ `nonTargetRoom`(`hallway1`) $\wedge$ `seeDoors`() $\wedge$ `notAllDoorsVisited`() $\wedge$ `seeVisitedDoor`($\mathtt{d}_1$) $\wedge$ `seeNonVisitedDoor`($\mathtt{d}_2$).

The abstract situation $\sigma_2 = \phi_s(s_2)$ is:

σ_2 = `inRoom`(`P`) $\wedge$ `nonTargetRoom`(`P`) $\wedge$ `notAllDoorsVisited`() $\wedge$ `seeDoors`() $\wedge$ `seeVisitedDoor`(`D`) $\wedge$ `seeNonVisitedDoor`(`D`).

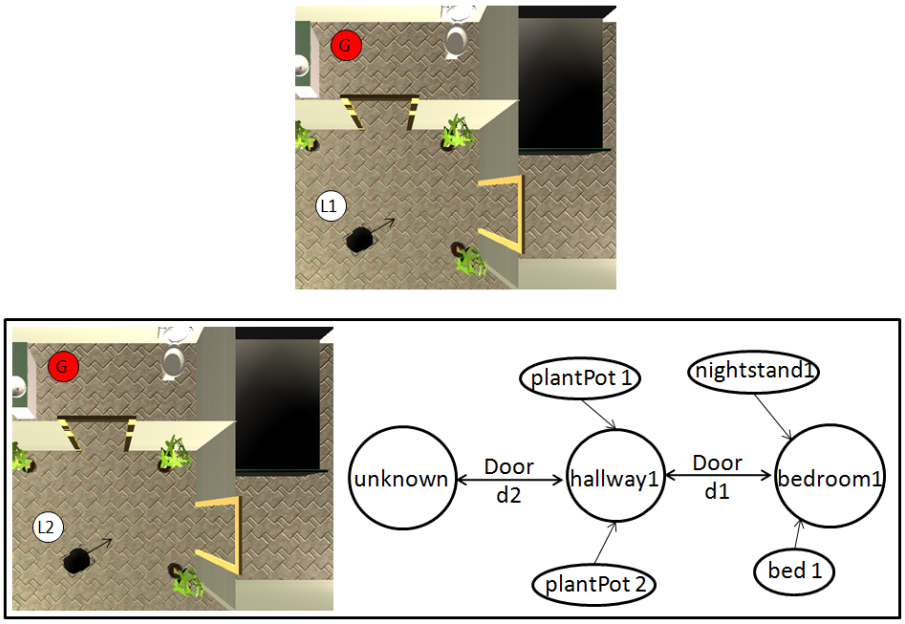
Given the restrictions of the semantic map, the robot has only one abstract behavior α_1 = `goToDoor`(`D`) with probability 1. Then the grounding operation $a_1 = grounding(\alpha_1, s_2)$ is executed and only one ground behavior is obtained for ground situation s_2. This ground behavior is $a_1 = goToDoor(d_2)$ since the robot in s_2 has already visited door d_1. Finally this new ground behavior a_1 is sent to the Actuation module.

A Stochastic ground policy is defined as $\pi(s,a) \rightarrow [0,1]$ with $a \in \mathcal{A}(s)$, where $\mathcal{A}(s)$ is the set of feasible ground behaviors in situation s. Therefore $\pi(s,a) = \frac{1}{|\mathcal{A}(s)|}$ in the beginning. As the robot explores the environment, it constructs a semantic map, which is used to transform the stochastic policy into a deterministic policy that allows solving the task given to the robot in an optimal way. Over time, the robot only uses this deterministic policy until it is faced with a new task. When the robot has already constructed a semantic map of the environment, planning techniques, instead of learning, can be used to solve new tasks.

3.1.3 ACTUATION

The Actuation module is built by using the framework Robot Operation System (ROS) (QUIGLEY et al., 2009) that is a open source tool that provides hardware ab-

Figure 13: Example of the reasoning and learning process, in which the robot needs to reach the target room "bathroom" (red circle) . Location L1 represents a robot in its initial position, when it does not know anything about the new environment. Location L2 represents a robot that has already explored its new environment and constructed a semantic map of the places explored.



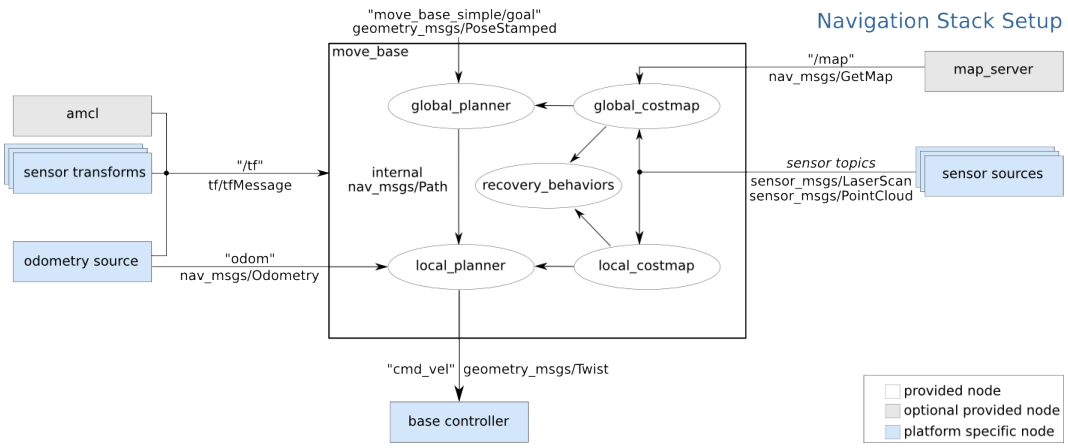
Source: Author.

straction, device drivers, libraries, visualizers, message-passing, package management, and more, for creating robot applications. The most important ROS libraries used in this module are the ACTIONLIB (MARDER-EPPSTEIN; PRADEEP, 2010) and MOVE-BASE (MARDER-EPPSTEIN, 2010c). The ACTIONLIB library allows constructing simple actions such as a translation of 2m or a rotation of 30°, while the MOVE_BASE allows finding a path without obstacles between two points.

This module is based on the 2D NAVIGATION STACK of ROS (MARDER-EPPSTEIN, 2010b) that sends the robot safe velocity commands, which are processed by using information from odometry, sensor streams such as laser scan or point cloud, static map (metric) and the desired goal. A detailed description of the ROS Navigation Stack is shown in Figure 14. This Navigation Stack has some inputs such as an external map (metric) input, which can be generated a priori with a SLAM algorithm or another algorithm and can be read from the disk using the *map_server* ROS library; an Adaptive Monte Carlo Localization (AMCL) system, besides some blocks such as a LOCAL PLANNER and a GLOBAL PLANNER, a LOCAL COSTMAP, a GLOBAL COSTMAP, and the MOVE_BASE ROS library.

However, as we are interested in a relational navigation and not in a metric one, some inputs such as the external metric map and the AMCL system are not used in our

Figure 14: The ROS Navigation Stack.



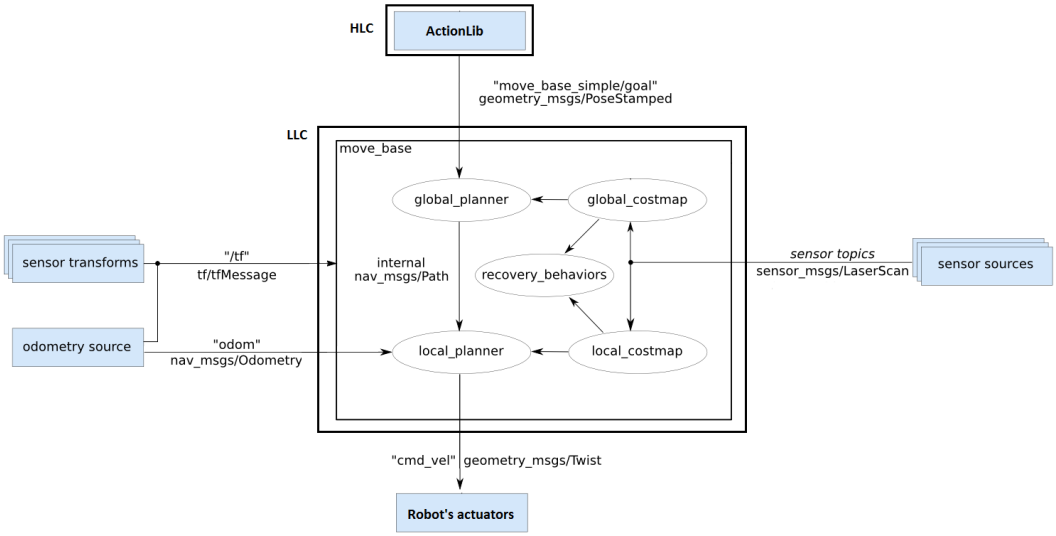
Source: Reproduced from (MARDER-EPPSTEIN, 2010b).

Actuation module. Therefore, this module uses the ROS Navigation Stack, but with some changes as shown in Figure 15.

The Actuation module is divided into a High Level Control (HLC) and a Low Level Control (LLC) (see Figure 15). HLC receives a behavior selected by the Reasoning and Learning module. Then the behavior is divided into simple actions by using the ACTIONLIB ROS library, which is a standardized interface client-server (see Figure 16) that allows sending a simple action to the LLC and monitoring the incremental progress of a simple action as well as the final result.

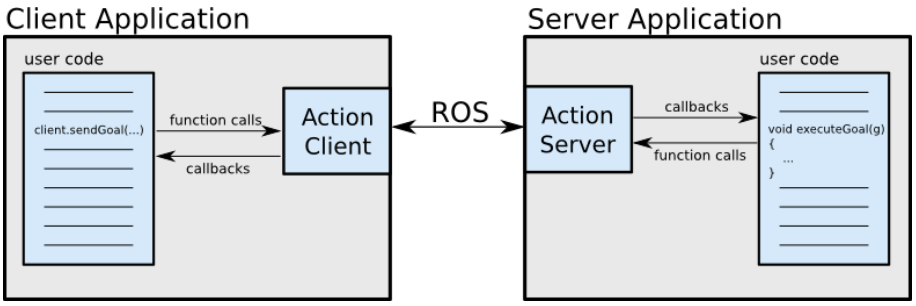
For better understanding how the HLC block works, let us suppose that the HLC receives ground behavior $a_1 = findDoor()$, the objective of which is to rotate the robot around its z-axis until it finds a door. This ground behavior a_1 is divided into some simple actions, which are 10° rotations. During each rotation, the HLC block verifies whether the robot found a door. If the robot found a door, ground behavior a_1 finishes; otherwise, the HCL block sends another simple action (to rotate 10°). Let us now suppose that the HLC receives another ground behavior $a_2 = goToNextRoom(unknown)$, aiming to lead the robot from its current room to another room that the robot can see through a door in front of it. This ground behavior a_2 has two requirements for applying it; first, the robot needs to stand approximately 1 meter away from the door and second, the robot must be facing the door. When the robot is 1 meter away from the door but it is not in front of the door, ground behavior a_2 can be divided into two simple actions: to rotate the robot around its z-axis until it is in front of the door and to move the robot forward 2 meters. The latter simple action moves the robot to another room. On the other hand, if the robot meets the two requirements (it is 1 meter away to the door and is facing the door), ground behavior a_2 has only one simple action: to

Figure 15: Overview of our Actuation module.



Source: Reproduced and modified from (MARDER-EPPSTEIN, 2010b).

Figure 16: The *ActionClient* and *ActionServer* communicate via a "ROS Action Protocol", which is built on top of ROS messages. The client and server then provide a simple API for users to request goals (on the client side) or to execute goals (on the server side) via function calls and callbacks.



Source: Reproduced from (MARDER-EPPSTEIN; PRADEEP, 2010).

move the robot forward 2 meters. Finally, each of the simple actions is sent to the LLC block by the HLC block.

Example 2 Just as an example of sending a simple action to the LLC block by the HLC block using the ACTIONLIB ROS library, consider a forward translation of 2 meters (a simple action of the behavior $a_2 = goToNextRoom(unknown)$), which is a simple action of translating the robot 2 meters in its x-axis. The code below shows how the simple action is executed from a simple client developed in python language:

```
self.move_base = actionlib.SimpleActionClient("move_base", MoveBaseAction)
```

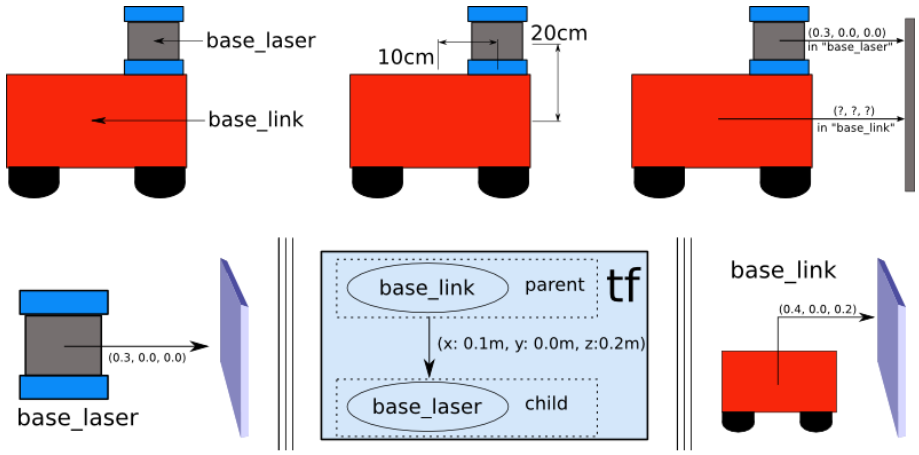
```
self.move_base.wait_for_server(self.rospy.Duration(10.0))
rospy.loginfo("Starting simple action")

goal = MoveBaseGoal()
goal.target_pose.header.frame_id = 'base_link'
goal.target_pose.pose.position.x = 2  # meters
goal.target_pose.pose.position.y = 0
goal.target_pose.pose.orientation.z = 0
goal.target_pose.pose.orientation.w = 1

# Send the pose to the MoveBaseAction server
self.move_base.send_goal(goal)

state = self.move_base.get_state()
if state == GoalStatus.SUCCEEDED:
    rospy.loginfo("Simple Action succeeded!")
```

Each simple action is received by the LLC block, which uses the MOVE_BASE ROS library to attempt to execute each simple action with the robot. Besides each simple action, the MOVE_BASE library also receives sensor information such as: odometry, laser scan, and sensor transforms. The latter information is the transform tree (tf) of a robot that defines translation offsets and rotation offsets between different coordinates frames and allows relating all the sensor measurements with the center point of the robot's base (see Figure 17).

Since the Actuation module does not use an external metric map, the NAVIGATION STACK is configured to use a *rolling window parameter*. This parameter keeps the robot in the center of a square, while it navigates throughout the indoor environment. The robot only cares about obstacles within the square local area.

Within the MOVE_BASE library several blocks are used to accomplish the simple actions (see Figure 15). Two costmap blocks are constructed and updated by the MOVE_BASE library, the GLOBAL COSTMAP and the LOCAL COSTMAP blocks. Both costmaps are formed by 4 layers, the Static map layer, the Obstacle layer, the Voxels layer and the Inflation Layer (LU et al., 2014). Since the Actuation module does not use either an initial static map or point cloud information as input, the two costmap blocks only use the Obstacle layer and the Inflation Layer (see Figure 18). The first one takes data from the laser sensor, which is collected in a two-dimensional grid. The read data are considered occupied spaces and the space between the sensor and the sensor measurements is considered free space; meanwhile, the Inflation layer inserts a buffer zone around each obstacle found in the Obstacle layer, which is considered a lethal obstacle

Figure 17: Example of the transform tree (tf) of a robot, which has two bases, the base_laser and the base_link. The first one represents the center point of the laser sensor and the second one represents the center point of the robot's base. In the tf of this robot, the base_link is the parent node since the other parts such as sensors are added to the robot. Therefore the base_laser is a child node. The edge between the two nodes in the tf represents translation offsets and rotation offsets between the different coordinates frames of the two bases. Finally, the data received by the base_laser are converted into the base_link.

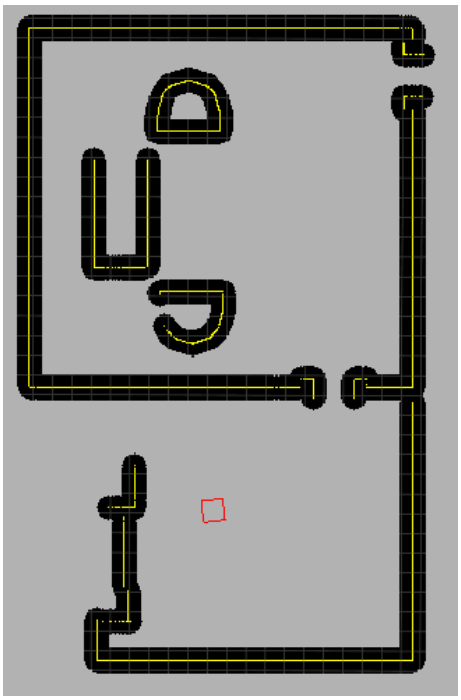


Source: Reproduced from (MARDER-EPPSTEIN, 2010a).

that has a lethal cost, while the buffer zone has a small non-lethal cost. Furthermore, the Local costmap is the short-term memory that uses the Obstacle and Inflated layer to temporally store the local data present in the rolling window (see Figure 19c), while the Global costmap is the long-term memory that uses the layers to store and to update the obstacle data during the navigation (see Figure 19b).

So far we have explained how LLC block uses the laser data to construct a 2D occupancy grid of the environment that separates free spaces from the occupied spaces (obstacles). After that, this grid is used to compute the costs of the spaces in a 2D costmap, which can be Local or Global. For better understanding how the costmaps are used by the LLC block, we take example 2 that sends to the LLC block the simple action "move the robot forward 2 meters", which is part of the ground behavior $a_2 = goToNextRoom(unknown)$. Les us suppose that the LLC block receives this simple action and the robot is located in the environment as shown in Figure 19a, after having explored the current place. Therefore, the Actuation module needs to construct a path that allows the robot to execute successfully the simple action. But before constructing the path, the robot needs to know what the obstacles in its way are. The costmaps allow knowing the obstacles in the the robot's path. For the robot's location shown in figure 19a, a partial Global costmap is shown in figure 19b and a Local costmap is shown in figure 19c. Now that the robot knows the obstacles in its surroundings, it can construct the path to apply the simple action sent to the LLC block in example 2.

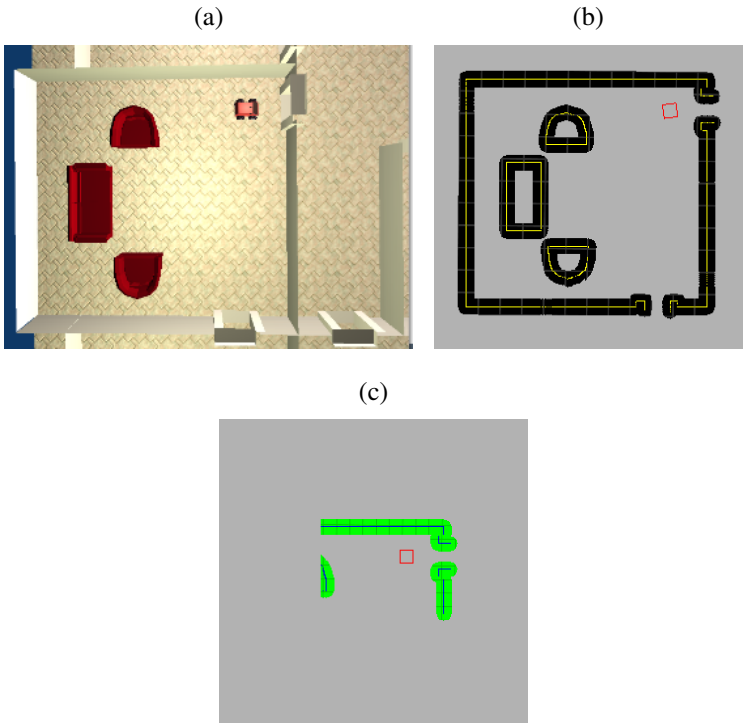
Figure 18: Example of a costmap, the red square represents the robot, the black lines represent the inflation layer and the yellow lines represent the obstacle layer.



Source: Author.

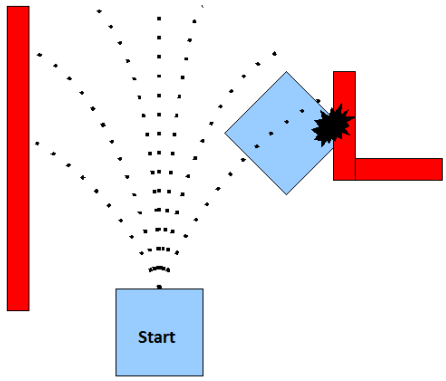
In order for the LLC block to construct a path between the current robot's pose and a goal pose, the MOVE_BASE library links the GLOBAL PLANNER and the LOCAL PLANNER (MARDER-EPPSTEIN et al., 2010). The first one takes the GLOBAL COSTMAP information, the goal pose of the simple action and the robot's location, which will be always the pose [0,0,0] for the Actuation module, and then it uses the Dijkstra's algorithm to create a high level path for the robot to reach the goal of the simple action. This GLOBAL PLANNER does not take into account the kinematics or the dynamics of the robot. The second planner takes the LOCAL COSTMAP information together with the kinematics and dynamics of the robot to produce safe velocity commands to move the robot towards a goal by attempting to follow the path created by the GLOBAL PLANNER as closely as possible. To produce the velocity commands, the LOCAL PLANNER uses the Dynamic Window Approach (DWA) technique (FOX et al., 1997), which samples the robot's control space in d_x, d_y, and d_{theta} velocities; for each sample, a forward simulation is performed that creates a trajectory (see Figure 20) with a score associated, which depends on some parameters such as the proximity to the obstacles, the proximity to the global path and others. The trajectory with the highest score is selected and the velocity commands that compose the trajectory are sent to the robot. While the robot receives the velocity commands, it sends to the LLC block the odometry information, which allows knowing how many meters the robot moved and how many degrees the

Figure 19: An example of a global costmap (b) of the indoor environment shown in (a), after the robot explored its current place. An example of a local costmap (c) in the current location of the robot in the environment in (a).



Source: Author.

Figure 20: Kinematic trajectories generated by the Local planner for the robot to get from a start to a goal location. The Local planner predicts for each trajectory what would happen if the robot takes each trajectory.

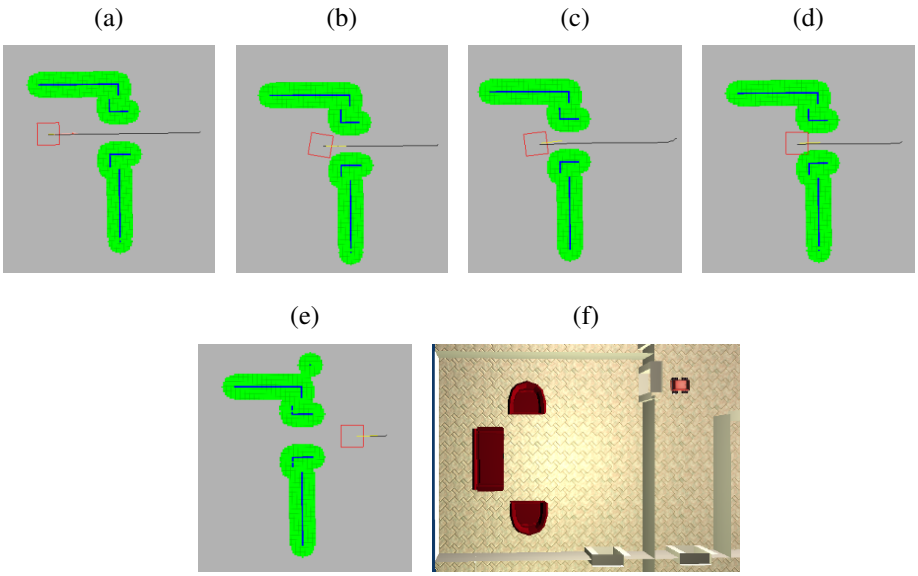


Source: Reproduced from (MARDER-EPPSTEIN; PERKO, 2010).

robot rotated. Finally, when the simple action is successfully performed, another simple action that composes the ground behavior is sent to the LLC block or if all the simple actions of the ground behavior have been performed, the behavior finishes.

Taking again example 2 that sends a simple action to the LLC block and using the Global costmap shown in figure 19b, the Global planner constructs a global path to achieve the simple action, this global path is shown in figure 21a (black line). Given that the robot only has three degrees of freedom, which we represent as t_x (translation in x-axis), t_y (translation in y-axis) and r_z (rotation in z-axis), a robot's pose can be represented by the vector $[t_x, t_y, r_z]$. Therefore, the start of the global path is the zero pose of the robot [0,0,0] and the end of the path is the goal pose that is the pose desired with the simple action, which in this case is [2,0,0]. The global path is a set of poses between the zero pose and the goal pose the robot can follow to execute the simple action. But this global path only has a set of poses that the robot needs to reach and it does not have the velocities that the robot should execute to reach each pose. Therefore, the Local planner uses the path constructed by the Global planner in figure 21a and the Local costmaps shown in figure 19c to produce a local path with the velocity commands that allows attempting to follow the global path. These velocity commands are represented by the vector $[d_x, d_y, d_{theta}]$ that are the translational velocity in x-axis and y-axis and the rotational velocity in z-axis repetitively. Some example of the local path are shown in figures 21b, 21c, 21d, and 21e that shows a yellow line (local path) that tries to follow the black line (global path). While the Local planner attempts to follow the global path, the Local planner receives from the robot's sensor the odometry information used by the Local planner to control the velocity commands. Finally, the robot finishes the simple action when it reaches the goal pose $[2,0,0]$ and, in turn,

Figure 21: An example of a global path (a) to achieve the simple action of example 2. Additionally (b),(c),(d), and (e) are examples of the local paths for the same simple action, and (f) is the final pose of the robot after executing the ground behavior $a_2 = goToNextRoom(unknown)$.



Source: Author.

the robot finishes the ground behavior $a_2 = goToNextRoom(unknown)$ as shown in Figure 21f.

4 EXPERIMENTS

In this chapter we are going to describe the simulated indoor environment used in this work. Furthermore we are going to present a case study that describes in detail the effectiveness of our framework.

4.1 IMPLEMENTATION AND DISCUSSION

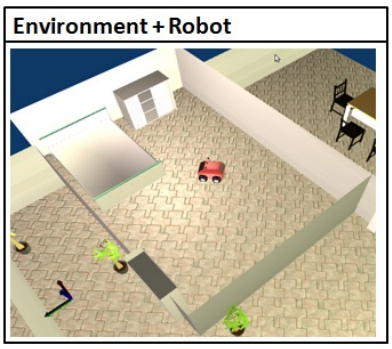
We now describe our implementation and experiments. The robot we consider is a wheeled base equipped with 5 sensors: 3 semantic cameras, 1 odometer and 1 laser range scanner. We ran tests in a simulated environment. This simulated scenario (see Figure 22) was designed with the open source tool for 3D creation, BLENDER[1], with which we have created a 3D CAD representation of the house and the objects it contains, including the robot (an ATRV 4-wheeled base); the representation was integrated into the Modular Open Robots Simulation Engine MORSE (ECHEVERRIA et al., 2012) and the Robot Operating System (ROS)[2]. The environment is a house that has eight types of rooms: 1 hallway (with some potted plants), 1 kitchen (with 1 stove, 1 fridge, and a dishwasher), 1 living room (with 1 sofa and 2 armchairs), 1 bathroom (with 1 toilet and 1 sink), 3 bedrooms (with 1 bed and 1 nightstand), and 1 dining room (with 1 table and 6 chairs).

The semantic camera is, in essence, a sensor that allows to recognize objects that the robot sees and the relative position between the robot and objects viewed. The robot was equipped with two semantic cameras that recognize general objects and one semantic camera that recognizes only doors. The odometer and the laser scanner are used in the Navigation module.

For a better understanding of how the architecture WRRA works, how is its integration with the information of the semantic web and the technologies employed, we describe a simple case study executed in the simulated scenario, where we have

[1] http://www.blender.org/
[2] http://wiki.ros.org/

Figure 22: Simulated house.



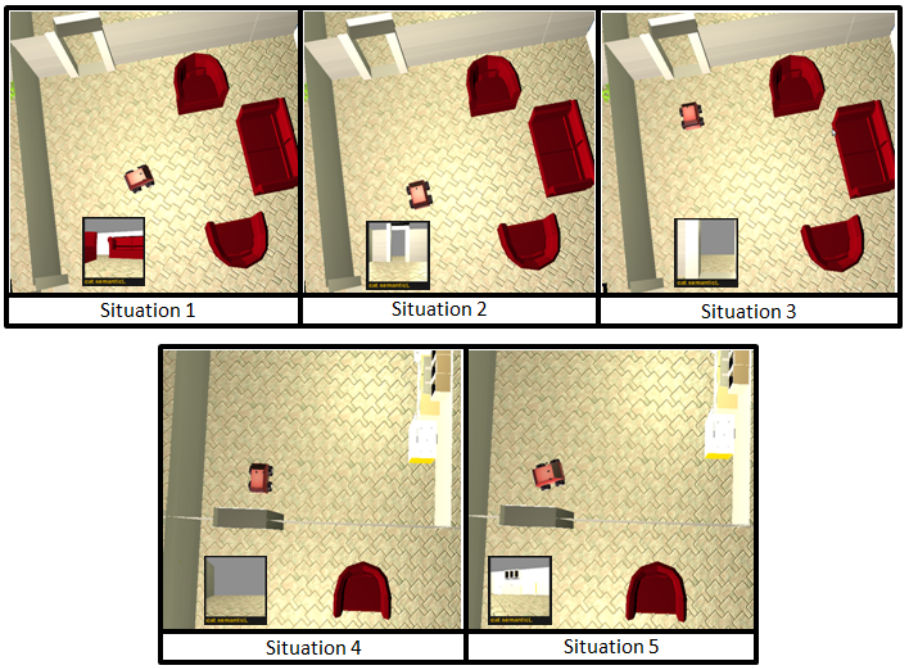
Source: Author.

great flexibility in defining tasks and measuring behavior. WRRA was implemented using the Python programming language and was integrated with the ROS framework. Initially, the robot knows nothing about the house and has only a generic abstract policy that defines mappings from abstract situations into abstract behaviors, such as $\pi_{abs}(\texttt{inRoom(X)} \wedge \texttt{nonTargetRoom(X)} \wedge \texttt{seeNoDoors()} \wedge \texttt{notAllDoorsVisited()}) = \texttt{findDoor()}$, Indicating that if the robot is in a certain room that is not the target room and it did not detect any door in this room and the list of doors visited by it is empty, then the robot must find a door. The robot is given the goal of reaching the home kitchen. Then, the robot perceives situations, and selects the appropriate behavior for each situation and performs it, until the target room is reached. Figure 23 describes a sequence of five situations faced by the robot.

Situation 1: The Perception module collects information from the environment using the robot semantic cameras. From the position where the robot is, two objects are detected: $\texttt{obj}_1 = \texttt{sofa}$ and $\texttt{obj}_2 = \texttt{armchair}$. Then the Place Inference submodule performs a query to the integrated ROS library KnowRob-OMICS, which estimates the most likely room where the robot is taking into account the objects detected by the robot:

```
prolog = json_prolog.Prolog()
for obj in objets:
    q="bayes_probability_given(knowrob:'OmicsLocations',
    Room,knowrob:'"+obj+"',Pr)"
    query = prolog.query(q)
    for solution in query.solutions():
        room=str(solution['Room'])[37::]
        places.place[room].append(solution['Pr'])
query.finish()
```

Figure 23: Sequence of situations faced by the mobile robot in case study.



Source: Author.

As the queries are implemented using Python and the KnowRob-OMICS is implemented using the PROLOG logic programming language, the WRRA uses the ROS library JSON-PROLOG [3] to send the queries from Python code to PROLOG. When the KnowRob-OMICS is queried, it returns the following information for each recognized object:

```
Room = 'knowrob.owl#Kitchen'
Pr = 0.1031101853182014 ;
Room = 'knowrob.owl#DiningRoom'
Pr = 0.12185749173969258 ;
Room = 'knowrob.owl#BedRoom'
Pr = 0.12185749173969258 ;
Room = 'knowrob.owl#LivingRoom'
Pr = 0.3655724752190777 ;
Room = 'knowrob.owl#Hallway'
Pr = 0.14893693434851316 ;
Room = 'knowrob.owl#BathRoom'
Pr = 0.1386654216348226 ;
```

[3] http://wiki.ros.org/json_prolog

This information gives the probability that each room is the current location of the robot, given only one recognized object. Figure 24 shows some probabilities that a room type has certain object type. These probabilities come from the ROS library KnowRob-OMICS that uses the Lidstone's law, which redistributes the probability mass assigned to the seen tuples to the unseen tuples like (bed,bathroom). The Lidstone's law uses a parameter $\lambda < 1$ and when the parameter $\lambda \to 1$ much probability is distributed to unseen tuples (GUPTA; KOCHENDERFER, 2004). In our experiments, we set $\lambda = 0.5$.

Then the Place Inference submodule uses the probabilistic reasoning process explained in Section 3.1.1 to infer the most likely place where the robot is by taking into account all objects recognized by the robot. In $situation_1$ of the case study reported here, the place inferred as the most likely place is $\mathtt{p}_1 = \mathtt{livingroom}$.

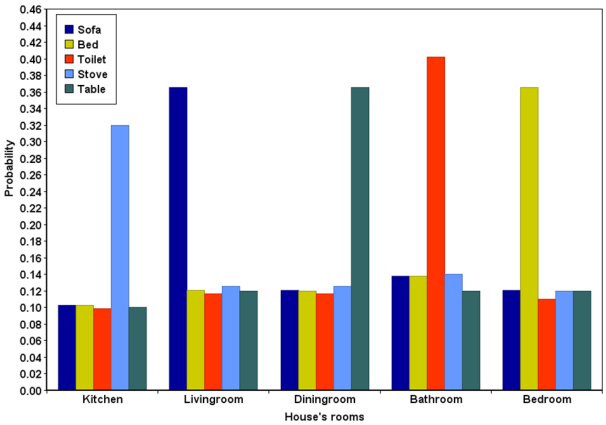
When objects and doors are detected and the place is inferred, the semantic map is updated. For this instance the map is updated with the objects $\mathtt{obj}_1$ and $\mathtt{obj}_2$ and the place $\mathtt{p}_1$ by building the relations $\mathtt{obj}_1$ at $\mathtt{p}_1$ and $\mathtt{obj}_2$ at $\mathtt{p}_1$. Next, the Inference Situation submodule receives information about detected doors, the inferred place and the updated semantic map. With this information, the truth values of the predicates are calculated and the conjunction of predicates with truth value TRUE forms a situation description using the SMACH library [4], which is a ROS-independent Python library to build hierarchical state machines. Finally that inferred situation is the output of the Perception module that for this case is the $situation_1 = \mathtt{inRoom}(\mathtt{livingroom}) \wedge \mathtt{nonTargetRoom}(\mathtt{livingroom}) \wedge \mathtt{seeNoDoors}() \wedge \mathtt{notAllDoorsVisited}()$.

Then $situation_1$ is sent as input to the Reasoning and Learning module, and it is transformed in its corresponding abstract situation. The abstract policy is then used to define the output to the Actuation module: $\pi_{abs}(\mathtt{inRoom}(\mathtt{X}) \wedge \mathtt{nonTargetRoom}(\mathtt{X}) \wedge \mathtt{seeNoDoors}() \wedge \mathtt{notAllDoorsVisited}()) = \mathtt{findDoor}()$ (see subsection 3.1.2). In the current version the output of this module is one of four behaviors: $\mathtt{goToDoor}(\mathtt{D})$ meaning that the robot should go to $\mathtt{D}$; $\mathtt{goToNextRoom}(\mathtt{P})$ meaning that the robot should go into the next place $\mathtt{P}$; $\mathtt{findDoor}()$ meaning that the robot should search for doors in the room; $\mathtt{exploration}()$ meaning that the robot should search for objects in the room.

Finally, in the Actuation module, the HLC submodule uses the ActionLIB ROS library [5] to decompose each behavior into a sequence of simple actions, which are in turn translated by the LLC submodule to respective velocities of translation and

[4] http://wiki.ros.org/smach
[5] http://wiki.ros.org/actionlib

Figure 24: Probabilities of room given observed object, by KnowRob-OMICS.



Source: Author.

rotation by using the Move Base ROS library [6], which allows the robot to reach a certain target pose. The Move Base library in turn uses other ROS libraries to avoid obstacles during navigation and to build a local path for the execution of each simple action sent by the HLC. Usually the Move Base library works with a static metric map of the environment, but in our case the Move Base library was set to work without it, since WRRA only reasons in relational level. The robot, controlled by the actuation module, searches for a door in the environment. At the moment its semantics camera detects a door, a new situation is defined.

Situation 2: When the robot perceives a door (in this case, it sees door $\mathtt{d}_1$), the Perception module updates the semantic map with the door $\mathtt{d}_1$ and the relation $\mathtt{d}_1$ at $\mathtt{p}_1$. So a new ground situation is determined by the Perception module: $situation_2 = \mathtt{inRoom(livingroom)} \wedge \mathtt{nonTargetRoom(livingroom)} \wedge \mathtt{seeDoors()} \wedge \mathtt{notAllDoorsVisited()} \wedge \mathtt{seeNonVisitedDoor(d_1)}$. This situation is turned into its corresponding abstract situation in the Reasoning and Learning module, and the abstract policy gives: $\pi_{abs}(\mathtt{inRoom(X)} \wedge \mathtt{nonTargetRoom(X)} \wedge \mathtt{seeDoors()} \wedge \mathtt{notAllDoorsVisited()} \wedge \mathtt{seeNonVisitedDoor(Y)}) = \mathtt{goToDoor(Y)}$. In this case, the grounding of behavior $\mathtt{goToDoor(Y)}$ gives $\mathtt{goToDoor(d_1)}$. Then, the Navigation module operates properly, using sensory information that gives the relative position of the robot to $\mathtt{d}_1$ and to obstacles, in order to drive the robot to the front door.

Situation 3: In this situation the robot knows it still is in the living room, it still sees the door $\mathtt{d}_1$ that has not been visited, and now it can see the adjacent room $\mathtt{p}_2$ through door $\mathtt{d}_1$. The map is updated by building the relation $\mathtt{p}_2$ connected to $\mathtt{p}_1$ through $\mathtt{d}_1$. This situation is:
$situation_3 = \mathtt{inRoom(livingroom)} \wedge \mathtt{nonTargetRoom(livingroom)} \wedge \mathtt{seeDoors()} \wedge$

[6] http://wiki.ros.org/move_base

$\texttt{notAllDoorsVisited}() \wedge \texttt{seeNonVisitedDoor}(\texttt{d}_1) \wedge \texttt{seeNextPlace}(\texttt{p}_2)$.
Given the abstraction of *situation*$_3$, the behavior indicated by the abstract policy is $goToNextPlace(Z)$ and the grounding of it results in $\texttt{goToNextPlace}(\texttt{p}_2)$. In this case, the Navigation module drives the robot through the door and the robot reaches the adjacent room $\texttt{p}_2$.

Situation 4: As the robot has not observed any object in this new room, then $\texttt{p}_2 = \texttt{unknown}$. In this case, the map does not need to be updated and the only predicate with truth-value TRUE is `inUnknownRoom`(). Then the Reasoning and Learning module outputs the behavior `exploration`(), meaning that the robot must explore the room, looking for objects.

Situation 5: Finally, the robot observes objects $obj_3 = \texttt{stove}$ and $obj_4 = \texttt{fridge}$ that allow inferring that it is in the $\texttt{p}_2 = \texttt{kitchen}$. Then the map is updated by bulding the relations obj_3 at $\texttt{p}_2$, obj_4 at $\texttt{p}_2$. As the kitchen is the target room, the episode ends and the task is fulfilled.

Finally, we ran some simple tests in a real environment (see Figure 25a) by using the real robot Pioneer 2DX, which was equipped with 3 web cameras. Besides, we stuck in some walls and objects some Quick Response (QR) codes with a red frame around the QR codes (see Figure 25b and 25c) . The idea with the tests within the real environment was to prove the use of QR-codes along with the web cameras to obtain functionality similar to a semantic camera. The process used to run the experiments in the real environment was; fist, the QR-codes were created by using the *qr-tools* library [7] (figure 25e shows an example) as we show below in a python language.

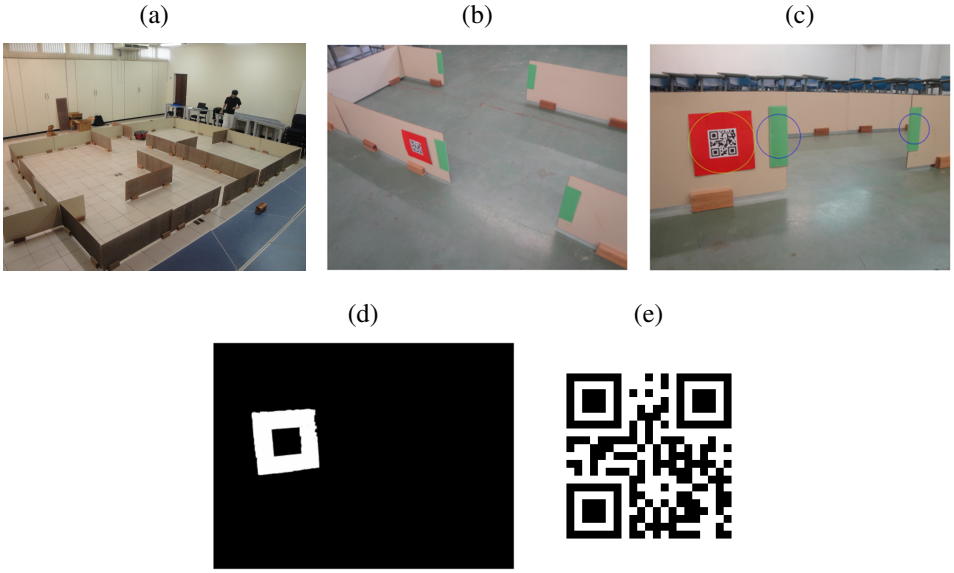
```
from qrtools import QR

code = QR(data=u"bed", pixel_size=10)
code.encode()
```

Then, we took a image from a web camera and applied the image processing technique *blob finder* (see Figure 25d), which separates regions in the image that differ in properties as color, with the areas surrounding those regions (we only wanted to detect red regions within the image); after finding the regions (the red frames around the QR codes), we took the QR-codes, then, we decode the QR-codes by using the *qr-tools* library as we show below in a python language.

```
from qrtools import QR
```

[7] https://launchpad.net/qr-tools

Figure 25: Figure (a) is the real environment used in the experiments. Figures (b) and (c) show some QR coder stuck in the walls. Figure (d) is the result of the image processing technique *blob finder* used to find the red frames that around the QR codes and figure (e) shows a QR-code that encodes the word *bed*.



Source: Author.

```
code = QR(filename=u"/home/walter/Escritorio/qr/bathroom.png")
if myCode.decode():
  print code.data
```

The results of the experiments in a real environment showed that the QR codes can be used as tool to identify objects with its respective semantic label without having to use complex image processing techniques. These results are important to future works that intend to implement the full WRRA architecture with a real robot.

5 CONCLUSIONS

In this dissertation we explored the use of semantic web resources to conduct probabilistic relational learning and reasoning in robot navigation. We presented an architecture (WRRA) for robot navigation that employs the KnowRob system and its knowledge base of probabilistic description logic sentences, together with relational reinforcement learning. The resulting framework shows how to use, in practice, semantic web resources that can deal with uncertainty.

We implemented the WRRA in a simulated environment, then in a real robot. The fact that the WRRA operates with semantic information, both in its knowledge base, and in its inputs and outputs, simplifies the whole process and leads to effective qualitative navigation. Moreover, the acquired knowledge base can be transferred to other scenarios. Our experiments indicate that indoor navigation can actually benefit from such a framework.

Our experiments showed that by combining the data from robots' sensors with semantic information and common sense knowledge database. The robot can face the problem of partial observability within an indoor environment since the combination of these data sources allows working in a higher level of abstraction, which, in turn, allows inferring information that the robot's sensor alone cannot obtain.

Several avenues are open to future work:

- **New predicates**. In our project, we propose 13 predicates to describe the environment in a qualitative relational way, but a new set of predicates can be added to study how these new predicates can improve the robot location.
- **New information sources such as human intervention**. We construct the predicates by using data from robot's sensors and semantic web resources, but human interventions can be used to teach the robot other information that permits the robot to construct new predicates.
- **Extensive tests with the real robot**. The WRRA framework was not extensively tested in a real robot. Therefore, tests can be performed by using RFID sensors

that allow the exact identification of an object by using a tag and a reader. The tag is stuck in the object and it has some information about the object while the reader is attached to the robot, which can read the information the tag has by using radio frequency.

- **To propose or to improve the behaviors** our approach proposes 4 behaviors, which can be improved or other behaviors can be added.

BIBLIOGRAPHY

AYDEMIR, A.; GÖBELBECKER, M.; PRONOBIS, A.; SJÖÖ, K.; JENSFELT, P. Plan-based object search and exploration using semantic spatial knowledge in the real world. In: *European Conference on Mobile Robot (ECMR)*. [S.l.: s.n.], 2011. p. 13–18.

AYDEMIR, A.; PRONOBIS, A.; GOBELBECKER, M.; JENSFELT, P. Active visual object search in unknown environments using uncertain semantics. *IEEE Transactions on Robotics*, IEEE, v. 29, n. 4, p. 986–1002, 2013.

AYDEMIR, A.; SJOO, K.; FOLKESSON, J.; PRONOBIS, A.; JENSFELT, P. Search in the real world: Active visual object search based on spatial relations. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.]: IEEE, 2011. p. 2818–2824.

BAADER, F. *The description logic handbook: theory, implementation, and applications*. [S.l.]: Cambridge University, 2003.

BEETZ, M.; JAIN, D.; MOSENLECHNER, L.; TENORTH, M.; KUNZE, L.; BLODOW, N.; PANGERCIC, D. Cognition-enabled autonomous robot control for the realization of home chore task intelligence. *Proceedings of the IEEE*, IEEE, v. 100, n. 8, p. 2454–2471, 2012.

BEETZ, M.; KLANK, U.; KRESSE, I.; MALDONADO, A.; MOSENLECHNER, L.; PANGERCIC, D.; RUHR, T.; TENORTH, M. Robotic roommates making pancakes. In: *IEEE-RAS International Conference on Humanoid Robots*. [S.l.]: IEEE, 2011. p. 529–536.

ECHEVERRIA, G.; LEMAIGNAN, S.; DEGROOTE, A.; LACROIX, S.; KARG, M.; KOCH, P.; LESIRE, C.; STINCKWICH, S. Simulating complex robotic scenarios with morse. In: *Simulation, Modeling, and Programming for Autonomous Robots*. [S.l.]: Springer, 2012. v. 7628, p. 197–208.

FOX, D.; BURGARD, W.; THRUN, S. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, v. 4, n. 1, p. 23–33, 1997.

GALINDO, C.; FERNÁNDEZ-MADRIGAL, J.-A.; GONZÁLEZ, J.; SAFFIOTTI, A. Robot task planning using semantic maps. *Robotics and Autonomous Systems*, Elsevier, v. 56, n. 11, p. 955–966, 2008.

GALINDO, C.; GONZÁLEZ, J.; FERNÁNDEZ-MADRIGAL, J.-A.; SAFFIOTTI, A. Robots that change their world: Inferring goals from semantic knowledge. In: *European Conference on Mobile Robot (ECMR)*. [S.l.: s.n.], 2011. p. 1–6.

GALINDO, C.; SAFFIOTTI, A. Inferring robot goals from violations of semantic knowledge. *Robotics and Autonomous Systems*, Elsevier, v. 61, n. 10, p. 1131–1143, 2013.

GALINDO, C.; SAFFIOTTI, A.; CORADESCHI, S.; BUSCHKA, P.; FERNANDEZ-MADRIGAL, J.-A.; GONZÁLEZ, J. Multi-hierarchical semantic maps for mobile robotics. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.]: IEEE, 2005. p. 2278–2283.

GÖBELBECKER, M.; HANHEIDE, M.; GRETTON, C.; HAWES, N.; PRONOBIS, A.; AYDEMIR, A.; SJÖÖ, K.; ZENDER, H. Dora: A robot that plans and acts under uncertainty. In: *35th German Conference on Artificial Intelligence*. [s.n.], 2012. p. 12. Accessed: february 2014. Available at: <http://www.cs.bham.ac.uk/~nah/bibtex/papers/gobelbeckeretal2012.pdf>.

GUPTA, R.; KOCHENDERFER, M. J. Common sense data acquisition for indoor mobile robots. In: *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. [S.l.: s.n.], 2004. p. 605–610.

HANHEIDE, M.; GRETTON, C.; DEARDEN, R.; HAWES, N.; WYATT, J.; PRONOBIS, A.; AYDEMIR, A.; GÖBELBECKER, M.; ZENDER, H. Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour. In: *Proceedings-International Joint Conference on Artificial Intelligence (IJCAI)*. [S.l.: s.n.], 2011. v. 22, n. 3, p. 2442.

HEARST, M. A.; DUMAIS, S.; OSMAN, E.; PLATT, J.; SCHOLKOPF, B. Support vector machines. *IEEE Intelligent Systems and their Applications*, IEEE, v. 13, n. 4, p. 18–28, 1998.

KLENK, M.; HAWES, N.; LOCKWOOD, K. Representing and reasoning about spatial regions defined by context. In: *AAAI Fall Symposium on Advances in Cognitive Systems*. [S.l.: s.n.], 2011.

KLINOV, P. *Practical reasoning in probabilistic description logic*. Phd Thesis (PhD Thesis) — The University of Manchester, Manchester, UK, 2011.

KOGA, M. L.; SILVA, V. F. da; COSTA, A. H. R. Online learning of abstract stochastic policies with monte carlo. In: *Proceedings of the 7th Workshop de Tecnologia Adaptativa (WTA)*. [S.l.: s.n.], 2013.

KOGA, M. L.; SILVA, V. F. da; COSTA, A. H. R. Stochastic abstract policies: Generalizing knowledge to improve reinforcement learning. *IEEE Transactions on Cybernetics*, PP, n. 99, 2014.

KOGA, M. L.; SILVA, V. F. da; COZMAN, F. G.; COSTA, A. H. R. Speeding-up reinforcement learning through abstraction and transfer learning. In: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '13)*. [S.l.: s.n.], 2013. p. 119–126.

KOSTAVELIS, I.; GASTERATOS, A. Learning spatially semantic representations for cognitive robot navigation. *Robotics and Autonomous Systems*, Elsevier, v. 61, n. 12, p. 1460–1475, 2013.

KUNZE, L.; TENORTH, M.; BEETZ, M. Putting people's common sense into knowledge bases of household robots. In: *KI 2010: Advances in Artificial Intelligence*. [S.l.]: Springer, 2010. v. 6359, p. 151–159.

LENAT, D. B.; GUHA, R. V.; PITTMAN, K.; PRATT, D.; SHEPHERD, M. Cyc: toward programs with common sense. *Communications of the ACM*, ACM, v. 33, n. 8, p. 30–49, 1990.

LIU, H.; SINGH, P. *OMCSNet: A commonsense inference toolkit*. [S.l.], 2003. Available at: <http://web.media.mit.edu/~push/OMCSNet.pdf>.

LIU, H.; SINGH, P. Conceptnet-a practical commonsense reasoning tool-kit. *BT Technology Journal*, Springer, v. 22, n. 4, p. 211–226, 2004.

LU, D. V.; HERSHBERGER, D.; SMART, W. D. Layered costmaps for context-sensitive navigation. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [s.n.], 2014. Accessed: February 2014. Available at: <http://www.cse.wustl.edu/~dvl1/publications/LayeredCostmaps.pdf>.

LUKASIEWICZ, T. Expressive probabilistic description logics. *Artificial Intelligence*, Elsevier, v. 172, n. 6, p. 852–883, 2008.

LUKASIEWICZ, T.; STRACCIA, U. Managing uncertainty and vagueness in description logics for the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, Elsevier, v. 6, n. 4, p. 291–308, 2008.

MARDER-EPPSTEIN, E. *ROS tutorial: Setting up your robot using tf*. [S.l.], 2010. Accessed: June 2014. Available at: <http://wiki.ros.org/navigation/Tutorials/RobotSetup/TF>.

MARDER-EPPSTEIN, E. *ROS tutorial: The 2D navigation stack*. [S.l.], 2010. Accessed: June 2014. Available at: <http://wiki.ros.org/navigation>.

MARDER-EPPSTEIN, E. *ROS tutorial: The move base package*. [S.l.], 2010. Accessed: June 2014. Available at: <http://wiki.ros.org/move_base>.

MARDER-EPPSTEIN, E.; BERGER, E.; FOOTE, T.; GERKEY, B.; KONOLIGE, K. The office marathon: Robust navigation in an indoor office environment. In: *International Conference on Robotics and Automation (ICRA)*. [s.n.], 2010. Accessed: February 2014. Available at: <https://www.willowgarage.com/sites/default/files/icra2010_marder-eppstein.pdf>.

MARDER-EPPSTEIN, E.; PERKO, E. *ROS tutorial: The base local planer package*. [S.l.], 2010. Accessed: June 2014. Available at: <http://wiki.ros.org/base_local_planner>.

MARDER-EPPSTEIN, E.; PRADEEP, V. *ROS tutorial: The actionlib stack*. [S.l.], 2010. Accessed: June 2014. Available at: <http://wiki.ros.org/actionlib>.

MATEUS, P.; PACHECO, A.; PINTO, J.; SERNADAS, A.; SERNADAS, C. Probabilistic situation calculus. *Annals of Mathematics and Artificial Intelligence*, Springer, v. 32, n. 1-4, p. 393–431, 2001.

MATOS, T.; BERGAMO, Y. P.; SILVA, V. F. da; COZMAN, F. G.; COSTA, A. H. R. Simultaneous abstract and concrete reinforcement learning. In: *Proceedings of the 9th Symposium on Abstraction, Reformulation and Approximation (SARA '11)*. [S.l.]: AAAI Press, 2011. p. 82–89.

MILLER, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, ACM, v. 38, n. 11, p. 39–41, 1995.

MOZOS, O. M.; MIZUTANI, H.; JUNG, H.; KURAZUME, R.; HASEGAWA, T. Categorization of indoor places by combining local binary pattern histograms of range and reflectance data from laser range finders. *Advanced Robotics*, Taylor & Francis, v. 27, n. 18, 2013.

PANGERCIC, D.; PITZER, B.; TENORTH, M.; BEETZ, M. Semantic object maps for robotic housework-representation, acquisition and use. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.]: IEEE, 2012. p. 4644–4651.

PARK, J.-T.; SONG, J.-B. Hybrid semantic mapping using door information. In: *IEEE International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. [S.l.]: IEEE, 2011. p. 128–130.

PATEL-SCHNEIDER, P. F.; ABRAHAMS, M.; RESNICK, L. A.; MCGUINNESS, D. L.; BORGIDA, A. Neoclassic reference manual: Version 1.0. *AT&T Labs Research, Artificial Intelligence Principles Research Department*, 1996.

PEARL, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. [S.l.]: Morgan Kaufmann, 1988.

POLASTRO, R.; CORRÊA, F.; COZMAN, F.; JR, J. O. Semantic mapping with a probabilistic description logic. In: *Advances in Artificial Intelligence–SBIA*. [S.l.]: Springer, 2011. v. 6404, p. 62–71.

PRONOBIS, A. *Semantic Mapping with Mobile Robots*. Phd Thesis (PhD Thesis) — KTH Royal Institute of Technology, Stockholm, Sweden, 2011.

PRONOBIS, A.; JENSFELT, P. Hierarchical multi-modal place categorization. In: *European Conference on Mobile Robot (ECMR)*. [S.l.: s.n.], 2011. p. 159–164.

PRONOBIS, A.; JENSFELT, P. Large-scale semantic mapping and reasoning with heterogeneous modalities. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.]: IEEE, 2012. p. 3515–3522.

PRONOBIS, A.; MOZOS, O. M.; CAPUTO, B.; JENSFELT, P. Multi-modal semantic place classification. *The International Journal of Robotics Research*, SAGE Publications, 2009.

PRONOBIS, A.; SJÖÖ, K.; AYDEMIR, A.; BISHOP, A. N.; JENSFELT, P. Representing spatial knowledge in mobile cognitive systems. In: *International Conference on Intelligent Autonomous Systems (IAS)*. [S.l.: s.n.], 2010.

QUIGLEY, M.; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.; WHEELER, R.; NG, A. Y. Ros: an open-source robot operating system. In: *ICRA Workshop on Open Source Software*. [S.l.: s.n.], 2009. v. 3, n. 3.2, p. 5.

RANGANATHAN, A. Pliss: Detecting and labeling places using online change-point detection. In: *Robotics: Science and Systems*. [S.l.: s.n.], 2010.

RANGANATHAN, A.; LIM, J. Visual place categorization in maps. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.]: IEEE, 2011. p. 3982–3989.

RIAZUELO, L.; TENORTH, M.; MARCO, D. D.; SALAS, M.; MÖSENLECHNER, L.; KUNZE, L.; BEETZ, M.; TARDOS, J.; MONTANO, L.; MONTIEL, J. Roboearth web-enabled and knowledge-based active perception. In: *IROS Workshop on AI-based Robotics*. [S.l.: s.n.], 2013.

ROTTMANN, A.; MOZOS, O. M.; STACHNISS, C.; BURGARD, W. Semantic place classification of indoor environments with mobile robots using boosting. In: *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. [S.l.: s.n.], 2005. v. 5, p. 1306–1311.

RUSSELL, B. C.; TORRALBA, A.; MURPHY, K. P.; FREEMAN, W. T. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, Springer, v. 77, n. 1-3, p. 157–173, 2008.

SAITO, M.; CHEN, H.; OKADA, K.; INABA, M.; KUNZE, L.; BEETZ, M. Semantic object search in large-scale indoor environments. In: *Proceedings of IROS 2012 Workshop on active Semantic Perception and Object Search in the Real World*. [S.l.: s.n.], 2011.

SCHUSTER, M. J.; JAIN, D.; TENORTH, M.; BEETZ, M. Learning organizational principles in human environments. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.]: IEEE, 2012. p. 3867–3874.

SINGH, P.; LIN, T.; MUELLER, E. T.; LIM, G.; PERKINS, T.; ZHU, W. L. Open mind common sense: Knowledge acquisition from the general public. In: *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*. [S.l.]: Springer, 2002. v. 2519, p. 1223–1237.

SINGH, P.; WILLIAMS, W. Lifenet: a propositional model of ordinary human activity. In: *Proceedings of the Workshop on Distributed and Collaborative Knowledge Capture (DC-KCAP) at KCAP*. [S.l.: s.n.], 2003. v. 2003, n. 7.

SUTTON, R. S.; BARTO, A. G. *Introduction to reinforcement learning*. [S.l.]: MIT Press, 1998.

TENORTH, M.; BEETZ, M. Knowrob: A knowledge processing infrastructure for cognition-enabled robots. *The International Journal of Robotics Research*, SAGE Publications, v. 32, n. 5, p. 566–590, 2013.

TENORTH, M.; KLANK, U.; PANGERCIC, D.; BEETZ, M. Web-enabled robots. *IEEE Robotics & Automation Magazine*, IEEE, v. 18, n. 2, p. 58–68, 2011.

TOPP, E. A.; HUETTENRAUCH, H.; CHRISTENSEN, H. I.; EKLUNDH, K. S. Bringing together human and robotic environment representations-a pilot study. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.]: IEEE, 2006. p. 4946–4952.

VAN OTTERLO, M. Reinforcement learning for relational MDPs. In: *Proceedings of the 13th Machine Learning Conference of Belgium and the Netherlands*. [S.l.]: Benelearn, 2004. p. 138–145.

VAN OTTERLO, M. *The logic of adaptive behavior: knowledge representation and algorithms for adaptive sequential decision making under uncertainty in first-order and relational domains*. Amsterdam: IOS Press, 2009.

VASUDEVAN, S.; GÄCHTER, S.; NGUYEN, V.; SIEGWART, R. Cognitive maps for mobile robots-an object based approach. *Robotics and Autonomous Systems*, Elsevier, v. 55, n. 5, p. 359–371, 2007.

VASUDEVAN, S.; SIEGWART, R. Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robotics and Autonomous Systems*, Elsevier, v. 56, n. 6, p. 522–537, 2008.

VISWANATHAN, P.; MEGER, D.; SOUTHEY, T.; LITTLE, J. J.; MACKWORTH, A. Automated spatial-semantic modeling with applications to place labeling and informed search. In: *Canadian Conference on Computer and Robot Vision (CRV)*. [S.l.]: IEEE, 2009. p. 284–291.

VISWANATHAN, P.; SOUTHEY, T.; LITTLE, J.; MACKWORTH, A. Automated place classification using object detection. In: *Canadian Conference on Computer and Robot Vision (CRV)*. [S.l.]: IEEE, 2010. p. 324–330.

WANG, T.; CHEN, Q. Object semantic map representation for indoor mobile robots. In: *IEEE International Conference on System Science and Engineering (ICSSE)*. [S.l.]: IEEE, 2011. p. 309–313.

WU, H.; TIAN, G.-h.; LI, Y.; ZHOU, F.-y.; DUAN, P. Spatial semantic hybrid map building and application of mobile service robot. *Robotics and Autonomous Systems*, Elsevier, v. 62, n. 6, p. 923–941, 2014.

WU, J.; CHRISTENSEN, H. I.; REHG, J. M. Visual place categorization: Problem, dataset, and algorithm. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.]: IEEE, 2009. p. 4763–4770.

WU, J.; REHG, J. M. Where am i: Place instance and category recognition using spatial pact. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.]: IEEE, 2008. p. 1–8.

ZARAGOZA, J. H.; MORALES, E. F. A two-stage relational reinforcement learning with continuous actions for real service robots. In: *MICAI : Advances in Artificial Intelligence*. [S.l.]: Springer, 2009. v. 5845, p. 337–348.

ZENDER, H.; MOZOS, O. M.; JENSFELT, P.; KRUIJFF, G.-J.; BURGARD, W. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, Elsevier, v. 56, n. 6, p. 493–502, 2008.

ZIVKOVIC, Z.; BOOIJ, O.; KRÖSE, B. From images to rooms. *Robotics and Autonomous Systems*, Elsevier, v. 55, n. 5, p. 411–418, 2007.