Draft Version
Published in Proceedings of the ECMR 2011

Robots that change their world: Inferring Goals from Semantic Knowledge

Cipriano Galindo*, Javier González,
Juan-Antonio Fernández-Madrigal
Dept. of System Engineering and Automation
University of Málaga, Spain

Alessandro Saffiotti
AASS Mobile Robotics Lab
Örebro University, Sweden

***Abstract*—A growing body of literature shows that endowing a mobile robot with semantic knowledge, and with the ability to reason from this knowledge, can greatly increase its capabilities. In this paper, we explore a novel use of semantic knowledge: to encode information about how things should be, or *norms*, to allow the robot to infer deviations from these norms and to generate goals to correct these deviations. For instance, if a robot has semantic knowledge that perishable items must be kept in a refrigerator, and it observes a bottle of milk on a table, this robot will generate the goal to bring that bottle into a refrigerator. Our approach provides a mobile robot with a limited form of *goal autonomy*: the ability to derive its own goals to pursue generic aims. We illustrate our approach in a full mobile robot system that integrates a semantic map, a knowledge representation and reasoning system, a task planner, as well as standard perception and navigation routines.**

***Index Terms*—Semantic Maps, Mobile Robotics, Goal Autonomy, Knowledge Representation, Proactivity**

I. Introduction

Mobile robots intended for service and personal use are being increasingly endowed with the ability to represent and use semantic knowledge about the environment where they operate [13]. This knowledge encodes general information about the entities in the world and their relations, for instance: that a kitchen is a type of room which typically contains a refrigerator, a stove and a sink; that milk is a type of perishable food; and that perishable food is stored in a refrigerator. Once this knowledge is available to a robot, there are many ways in which it can be exploited to better understand the environment or plan actions [21], [18], [19], [10], [22], assuming of course that this knowledge is a faithful representation of the properties of the environment. There is, however, an interesting issue which has received less attention so far: what happens if this knowledge turns out to be in conflict with the robot's observations?

Suppose for concreteness that the robot observes a milk bottle laying on a table. This observation conflicts with the semantic knowledge that milk is stored in a refrigerator. The robot has three options to resolve this contradiction: (a) to update its semantic knowledge base, e.g., by creating a new subsclass of milk that is not perishable; (b) to question the validity of its perceptions, e.g., by looking for clues that may indicate that the observed object is not a milk bottle; or (c) to modify the environment, e.g., by bringing the milk into a refrigerator. While some work have addressed the first two options, the last one has not received much attention so far. Interestingly, the last option leverages an unique capability of robots: the ability to modify the physical environment. The goal of this paper is to investigate this option.

We propose a framework in which a mobile robot can exploit semantic knowledge to identify inconsistencies between the observed state of the environment and a set of general, declarative descriptions, or *norms*, and to generate *goals* to modify the state of the environment in such a way that these inconsistencies would disappear. When given to a planner, these goals lead to action plans that can be executed by the robot. This framework can be seen as a way to enable a robot to proactively generate new goals, based on the overall principle of maintaining the world consistent with the given declarative knowledge. In this light, our framework contributes to the robot's *goal autonomy*. Although behavioral autonomy has been widely addressed in the robotic arena by developing deliberative architectures and robust algorithms for planning and executing tasks under uncertainty, goal autonomy has received less attention, being explored in the last years in the theoretical field of multi-agents [8], [4] and implemented through *motivational architectures* [1], [7].

Our framework relies on a *hybrid semantic map*, which combines semantic knowledge based on description logics [2] with traditional robot maps [11], [21], [18]. Semantic maps have been already shown to increase the robot's behavioral autonomy, by improving their basic skills (planning, navigation, localization, etc.) with deduction abilities. For instance, if a robot is commanded to "fetch a milk bottle" but it ignores the target location, it can deduce that milk is supposed to be in fridges which, in turn, are located at kitchens. We now extend our previous works on these issues [11], [10] to also include partial goal autonomy through the proactive generation of goals based on the robot's internal semantic model.

More specifically, we consider a robot with the innate objective of keeping its environment in good order with respect to a given set of norms, encoded in a declarative way in its internal semantic representation. Incoherences between the sensed reality and the model, i.e., the observation of facts that violate a particular norm, will lead to the generation of the corresponding goal that, when planned and executed, will re-align the reality to the model, as in the milk bottle example discussed above. It should be emphasized that in this work we only focus on the goal inference mechanism: the

*Corresponding author. Department of System Engineering and Automation. University of Málaga, Campus de Teatinos. E-29071 Málaga, Spain. Email: cipriano@ctima.uma.es.

development of the required sensorial system, and the possible use of semantic knowledge in that context, are beyond the scope of this paper.

Our approach to goal autonomy can be seen as a case of normative goals applied to agents which act based on beliefs and intentions [8], [4]. However, normative goals are often considered as simple *if-then* rules triggered when particular stimuli are given in the environment [20], [1]. Other works have used the term *maintenance goals* to represent innate goals that are aimed to satisfy a particular state of the world over time, e.g., the battery level should be always over a certain value [3], [12]. Our approach substantially diverges from those works, since it is not based on procedural rules, i.e., motivation-action pairs, nor if-then rules. Instead, we rely on a declarative representation of the domain, using the LOOM description logic language [17], from which the robot *infers* what should be done according to the current factual information in order to maintain the consistency between its environment and its representation.

This manuscript is structured as follows. In the next section we present our semantic map. Section III formalizes the use of semantic knowledge for goal generation. In section IV a real experiment is described. Finally some conclusions and future work are outlined.

II. A Semantic Map for Mobile Robot Operation

The semantic map considered in this work, derived from [10], comprises two different but tightly interconnected parts: a *spatial box*, or S-Box, and a *terminological box*, or T-Box. Roughly speaking, the S-Box contains factual knowledge about the state of the environment and of the objects inside it, while the T-Box contains general semantic knowledge about that domain, giving meaning to the entities in the spatial box in terms of concepts and relations. For instance, the S-Box may represent that `Obj-3` is placed at `Area-2`, while the T-Box may represent that `Obj-3` is a stove which is a type of appliance. By combining the two sides, the semantic map can infer, for instance, that `Area-2` is a kitchen, since it contains a stove.

This structure is reminiscent of the structure of hybrid knowledge representation (KR) systems [2], which are now dominant in the KR community. Our semantic map extends the assertional component to be more than a list of facts about individuals by also associating these individuals to sensor-level information with a spatial structure — hence the name S-Box. Please refer to [10] for more detail.

Figure 1 shows a simple example of a semantic map of a home-like environment where both the S-Box and the T-Box have a hierarchical structure. The hierarchy in the T-Box is a direct consequence of the fact that the represented semantic knowledge forms a taxonomy. For the S-Box, the use of a hierarchical spatial representation is a convenient and common choice in the robotic literature [15], [9] for dealing efficiently with large-scale environments. Of course one could also consider a flat representation in the S-Box: in fact, in our framework, the S-Box can be substituted by any other spatial representation.

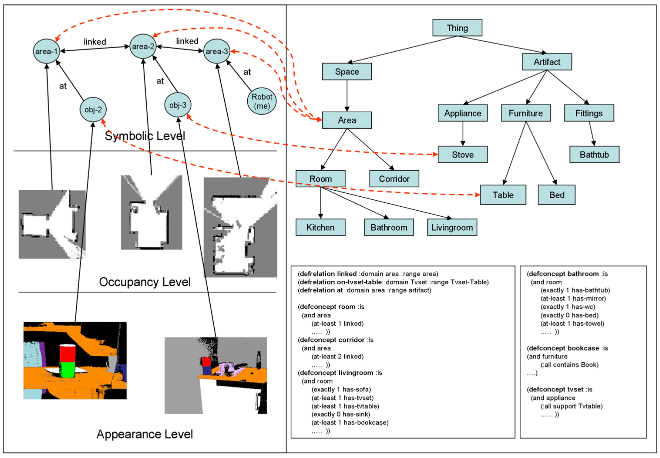

Fig. 1. An example of semantic map for a home-like environment. S-Box is on the left and T-Box on the right. See explanation in the text.

In the next section we exploit this semantic map, and more precisely the T-Box, for robot goal inference.

III. Inferring Goals from Semantics

The semantic map described above provides two different points of view of the robot workspace. On the one hand the spatial part (S-box) enables the robot to generate plans from basic skills, striving for behavioral autonomy. On the other hand the terminological part (T-box) provides an abstract model of the robot environment which includes general knowledge, e.g., *books are located on shelves*, which can be exploited for the automatic generation of robot goals.

First we give an informal description of the proposed mechanism for goal generation. Then, section III-B formalizes our approach under description logic. Finally, section III-C illustrates the process with two intuitive examples.

A. Informal Description

In the field of knowledge representation, semantic knowledge is usually interpreted as being *descriptive* of a specific domain: for example, the item of knowledge "beds are located in bedrooms" is used to partially describe beds. This knowledge is most useful to infer implicit properties from a few observed facts. For example, if the robot perceives a bed in a room it can infer that the room is a bedroom; conversely, if it is commanded to find a bed it can restrict its search to bedrooms. Galindo *et al.* [10] offer examples of applications of these inferences in the robotic domain.

Interestingly, semantic knowledge can also be interpreted as being *normative*: under this interpretation, the above item of knowledge is prescribing where a bed must be located. The difference becomes apparent when considering how a robot should react to an observation that contradicts this knowledge. Consider again the milk box example in the Introduction, and the three possible options to resolve the contradiction discussed there. Options (a) (update the model) and (b) (update the perceived state) correspond to modifying the robot's beliefs to recover from a contradiction, and are related to execution monitoring and uncertain state estimation. These options has been explored previously [11], [6]. The third option (c) (update the world) involves goal generation, and it is the one addressed here.

Informally, our approach defines a subset of concepts and relations stored in the T-Box as *normative*, i.e. they are involved in norms that should be fulfilled, by defining a special class `normative-concept` and a special relation `normative-relation`. Items of knowledge to be treated as normative will derive from them.

For instance, we can define that the normative concept `towel` should be related to the concept `bathroom` through the *normative* relation `place` — that is, towels *should* be located in a bathroom.

When a given instance violates a norm in the T-Box, the system derives the items of knowledge involved in the norm, and hence the goal that should be posted in order to satisfy that norm. In our example, suppose that an instance of a towel is perceived in a room which is not a bathroom. Then the given definition of a `towel` is violated — a circumstance that can be detected by most knowledge representation systems, including the LOOM [17] system used in our experiments. Since the above definition of towel is normative, the system yields a goal to satisfy the constraint, that is, to make the `place` of this towel be an instance of a `bathroom`. If the robot knows that, let say, `room-3` is a bathroom, this means that the goal "bring the towel to `room-3`" is generated.

B. Description Logic Representation for Inferring Normative Goals

Let $\mathfrak{I}$ be a description logic interpretation on a particular domain $\mathcal{D}$. Let $\wp$ define a set of disjoint concepts $\wp = \{P_1, \ldots P_n\}$, i.e., $\forall a, a \sqsubseteq P_i \Rightarrow \nexists j, j \neq i, a \sqsubseteq P_j$, where $x \sqsubseteq y$ denotes that x is subsumed by concept y.

Let N_r be called a *normative relation*, a function defined as:

$$N_r : N_C \rightarrow \wp$$

where N_C represents the so-called *normative concepts*, that is, concepts which ought to be properly related to those from $\wp$. N_r actually defines the norms to be kept. Normative relations are defined as one-to-one function as $\forall b \sqsubseteq N_C \Rightarrow \dot{\exists} P_j \in \wp, b \rightarrow [FILLS : N_r\ P_j]$.

The N_C set is further divided into two disjoint sets: the set $\triangle$ of all normative concepts that fulfill the imposed norms, and the set $\overline{\triangle}$ of those that fail to fulfill some of the norms (see figure 2).

Within this structure of the domain, constraint violations are automatically inferred when instances of the defined partitions are deduced to belong to a number of disjoint concepts. Let see an example:

Let C a normative concept (and therefore $C \sqsubseteq \triangle$ by definition) which is related to the P_i concept through the normative relation N_r. That is,

$$\forall c \sqsubseteq C, c \rightarrow [FILLS : N_r\ x], x \sqsubseteq P_i$$

If in a given interpretation $\mathfrak{I}$, $\exists k \sqsubseteq C, k \rightarrow [FILLS : N_r\ y], y \sqsubseteq P_j \in \wp, P_j \neq P_i \Rightarrow \mathfrak{I} \models y \sqsubseteq P_j \wedge y \sqsubseteq P_i \Rightarrow$

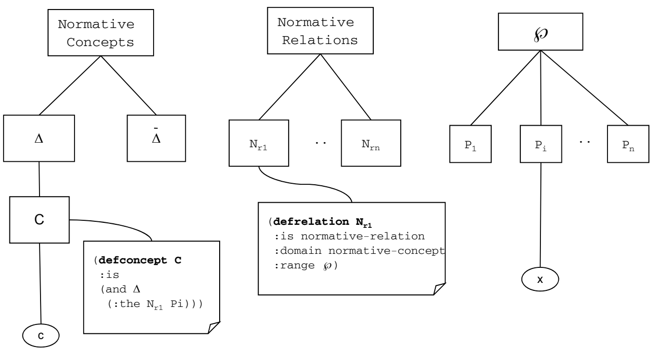

Fig. 2. Knowledge representation for detecting inconsistencies. Boxes represent concepts while instances are represented as circles. The concept C is defined as a normative concept related to P_i through the normative relation N_{r1}. See explanation in the text.

(Incoherent y). That is, if the normative relation is not met for a particular instance of a normative concept, the filler of such an instance, in this case y, becomes incoherent. Moreover, since k is defined as $k \sqsubseteq C \sqsubseteq \triangle$, it is also inferred that $k \sqsubseteq \overline{\triangle}$, which also makes k incoherent.

Goal Inference. Given an incoherent instance of a normative concept, $k \sqsubseteq C$ and the normative relation $N_r, N_r(k) = x, x \sqsubseteq P_i \in \wp$, the inferred goal to recover the system from the incoherence is:

```
(exists ?z (Pi z) (Nr k z))
```

That is, in the goal state, there should exist an instance of P_i related to k through the normative relation N_r[1].

C. Sample Scenarios

In this section we describe two illustrative examples.

1) Milk should be inside fridges: Consider a home assistant robot taking care of an apartment. Among other norms, the robot might watch milk bottles so they are always kept inside the fridge (see an implementation in section IV).

The semantic map for this scenario will entail information about the different rooms, i.e. kitchen, livingroom, bathroom, etc., the objects found inside, i.e. tables, chairs, fridges, books, bottles, etc, and their relations. Following the formalization given in III-B, part of the description of this scenario includes the partition of different places where bottles of milk could be found, e.g. $\wp = \{fridge, table, shelf\}$, being `milk-bottle` a normative concept, i.e. $milk-bottle \sqsubseteq \triangle$, (see figure 3).

Note that this definition implicitly provides certain restrictions that any bottle of milk should fulfill. Precisely, `milk-bottle` is assumed to be a beverage which has to meet at least one norm imposed by a normative relation, since it is subsumed by the `fulfilling-norm` concept.

Through the definitions given in figure 3, the expression `(:the place fridge)` indicates that every bottle of milk ought to be located in one location that must be

[1] It is not necessary to add the negation of $(N_r\ k\ z)$ to the goal state, since the N_r function is defined as one-to-one.

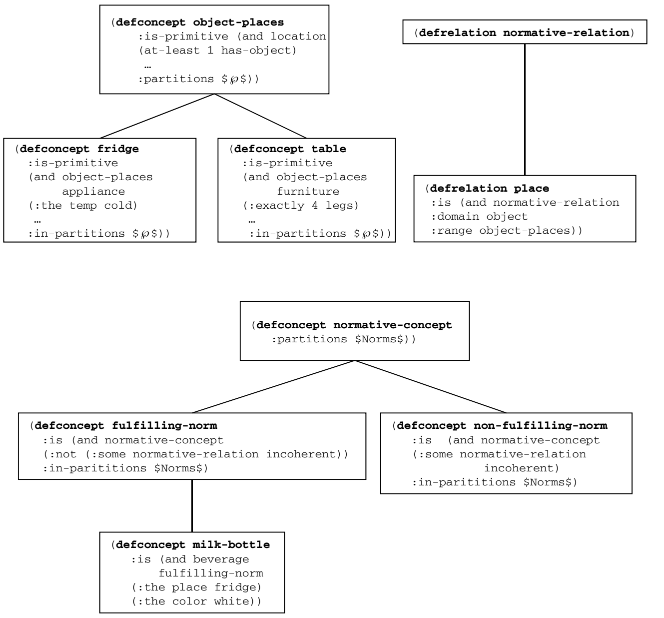

Fig. 3. Part of the domain definition for the "milk inside fridges" example. For clarity sake, *fulfilling-norm* is used instead of $\triangle$ and *non-fulfilling-norm* instead of $\overline{\triangle}$.

a fridge. Notice that in this example, the other restriction `(:the color white)` is not defined as normative relation, and thus, if it is not fulfilled in the scenario it will be simply deduced that the object is not a bottle of milk and no incoherences or robot goals will be generated.

Let us now consider the following situation in which the information gathered by the robot contradicts the definitions in the domain:

```
{(table t)(milk-bottle mb)(fridge f)
(place mb t)}
```

Under this interpretation, LOOM infers that the instance `t` should be a `fridge` since there is a bottle of milk placed on it. Such an inference produces an incoherence in the model given that the instance `t` is deduced to belong to two concepts, i.e. `table` and `fridge`, which have been defined as members of a partition. In this situation `t` is marked by LOOM as "incoherent".

Moreover, it is also deduced that the instance `mb`, initially defined as $mb \sqsubseteq \triangle$, also belongs to $\overline{\triangle}$ since the normative relation `(:the place fridge)` is filled with an incoherent instance. Again the system detect that `mb` belongs to two concepts defined in a partition and thus, it is also marked as "incoherent". The result is that the instances involved in the violated norm are detected and marked as incoherent. By checking the domain definition of such an incoherent instances, the follwoing goal is deduced: [2]

```
(exist ?x (fridge ?x) (place mb ?x))
```

That is, the robot has to put the bottle of milk represented by `mb` inside any object `?x` which is known to be a fridge. Since in the robot's domain there is a single fridge `f`, the above goal

[2]This goal is expressed in the goal language of the planner used in our experiment (see below), which is a subset of FOL.

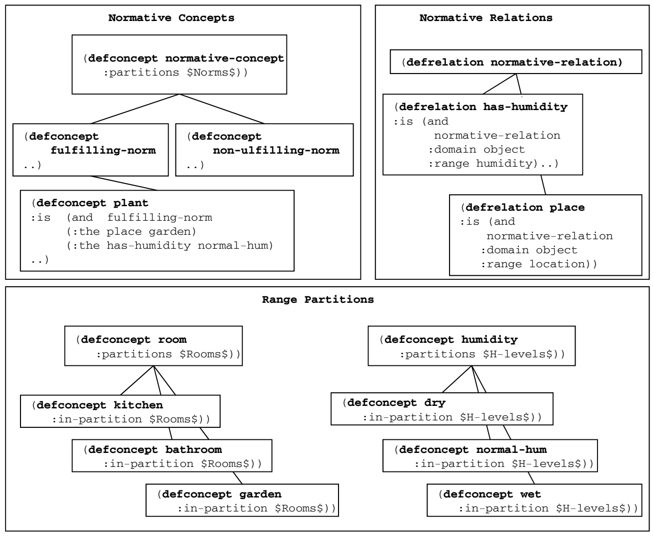

Fig. 4. General scheme for representing multiple norms. A particular partition has to be defined for each normative relation.

is instantiated as `(place mb f)`.

2) Plant should be properly watered: We now describe a more general example in which two norms are imposed on the same normative concept.

Consider we impose that "plants should be placed at the garden and have a normal humidity level". In this case we need two normative relations `place` and `has-humidity` and two partitions of concepts representing the possible, disjoint values for such relations. Figure 4 depicts part of the T-Box for this example.

Let us consider the following situation:

```
{(kitchen k)(bathroom b)(garden g)
(plant p)(place p k)(humidity-value dry)
(has-humidity p dry)}
```

As in the previous example, the process for detecting norms violation checks for incoherent instances. In this case instances `k` and `dry` become incoherent since they are deduced to belong to `{kitchen,garden}` and `{dry,normal-hum}` respectively. Besides, the instance `p` is also incoherent and therefore the following goal is generated:

```
(and
 (exist ?x (garden ?x)(place p ?x))
 (exist ?y (normal-hum ?y)(has-humidity p ?y)))
```

IV. AN ILLUSTRATIVE EXPERIMENT

We now illustrate the applicability of our goal generation technique to a real robotic application by showing an illustrative experiment run in a home environment. The experiment is inspired by the "milk" scenario in Sec. III above. In this experiment, a distributed network of sensors is used to update the state of the environment stored in the semantic map. Our algorithm is then run to detect violations of semantic constraints, and to generate goals which are passed to a planning and execution system in the robot.

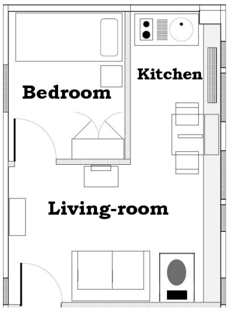

Fig. 5. The test environment. Left: layout. Right: the robot Astrid.

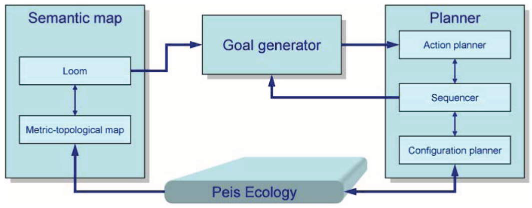

Fig. 6. Sketch of the software architecture used in our experiments. Only the modules and connections relevant to goal generation are shown.

A. Physical setup

We have used a physical test-bed facility, called the PEIS-Home [23], that looks like a bachelor apartment of about $25\,m^2$ and consists of a living-room, a bedroom and a small kitchen — see Fig. 5. The PEIS-Home is equipped with a communication infrastructure, and with a number of sensing and actuating devices, including a few mobile robots. Relevant to the experiments reported here are:

- a refrigerator equipped with a computer, some gas sensors, a motorized door, and an RFID tag reader;
- an RFID tag reader mounted under the kitchen table;
- a set of RFID tagged objects, including a milk cartoon;
- a set of webcams mounted on the ceiling; and
- Astrid, a PeopleBot mobile robot equipped with a laser scanner, a PTZ camera, and a simple gripper.

A few provisions have been introduced to simplify execution. In particular, since Astrid does not have a manipulator able to pick-up an object and place it somewhere else, these operations have been performed with the assistance of a human who puts the object in and out from the Astrid's gripper. These simplifications are acceptable here, since the purpose of our experiments is not to validate the execution system but to illustrate our goal generation algorithm in the context of a full robotic application.

B. Software setup

The software system used in our experiment is schematically shown in Fig. 6. The block named "PEIS Ecology" contains all the robotic components and devices distributed in the PEIS-Home. These are integrated through a specific middleware, called the PEIS-Middleware, that allows to dynamically activate and connect them in different ways in order to perform different tasks [5]. A set of activations and connections is called a *configuration* of the PEIS Ecology. For instance, the configuration in which the ceiling cameras are connected via an object recognition to the navigation controller onboard Astrid can be used to let the robot reach a given object.

The semantic map is based on a simple metric-topological map attached to the LOOM knowledge representation system [17]. Newly observed facts are asserted in LOOM using the `tell` primitive. The goal generation system interacts with LOOM as described in Sec. III above. Newly generated goals are passed to the planning system. This consists of three parts: an action planner, called PTLplan [14], that generates a sequence of actions to satisfy the goal; a sequencer, that selects those actions one by one; and a configuration planner [16], that generates the configuration needed to perform each action. When the current plan is completed, the goal generation system is re-activated.

C. Execution

Before the execution started, the semantic map contained a metric-topological map of the PEIS-Home, and the considered semantic knowledge in LOOM . In particular, the following statement was included in the LOOM knowledge base

```
(defconcept MilkBox :is
    (:and Container FulfillingNorm
          (:the place Fridge) ))
```

This encodes the normative constraint that any instance of the normative class `MilkBox` must have a single filler for the `place` relation, and that this filler must be of the class `Fridge`.

An RFID tag has been attached to a milk box, containing an encoding of the following information:

```
id: mb-22
type: MilkBox
color: white-green
size: 1-liter
```

At start, the milk is put on the kitchen table, called `table-1` in the map. The RFID tag reader under the table detects the new tag, and reports the information that `mb-22` is a `MilkBox` and it is at `table-1` — see Fig. 7. This information is entered into LOOM by:

```
(tell (MilkBox mb-22))
(tell (place mb-22 table-1))
```

As discussed in Sec. III, this information renders both the instances `mb-22` and `table-1` incoherent. The goal generation algorithm identifies `mb-22` as the normative instance. The algorithm then searches through all the relations that constrain `mb-22` to find a violated normative one, and it finds `place`. Since this relation should be filled by an instance of `Fridge`, it generates the following goal:

```
(exists ?x (and (Fridge ?x)(place mb-22 ?x)))
```

PTLplan uses the knowledge in the semantic map, together with its domain knowledge about the available actions, to generate the following action plan (simplified):

```
((MOVE astrid table-1) (PICKUP astrid mb-22)
 (OPEN fridge-1) (MOVE astrid fridge-1)
 (PLACE astrid mb-22) (CLOSE fridge-1))
```

where the variable `?x` has been instantiated by `fridge-1`.

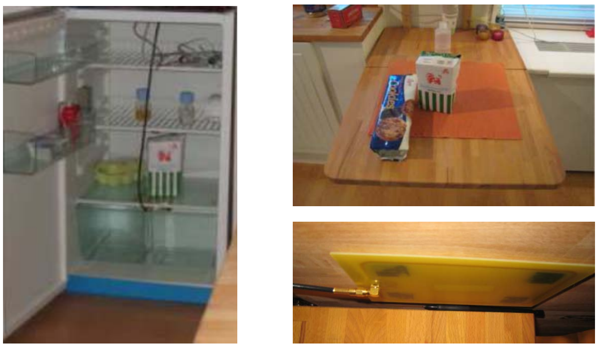

Fig. 7. RFID tagged objects and RFID tag readers used in our experiments. Left: in the fridge. Right: in the kitchen table.

The sequencer passes each action in turn to the configuration planner, which connects and activates the devices in the PEIS Ecology needed to execute it. For example, the first two actions only require devices which are on-board Astrid, while the third action requires the activation of the fridge door device. (The details of this "ecological" execution are not relevant here: see [16] for a comprehensive account.) As mentioned above, the `PICKUP` and `PLACE` actions were performed with the help of a human.

After the milk is removed from the table, the RFID tag reader under the table detects its absence and it signals it to the semantic map. When the milk is placed into the fridge, it is detected by the reader in the fridge. Corresponding to these two events, the following assertions are made in LOOM :

```
(forget (place mb-22 table-1))
(tell (place mb-22 fridge-1))
```

After execution is completed, the sequencer re-activates the goal generator. Since the place of `mb-22` is now an instance of a fridge, no incoherence is detected and no new goal is generated.

V. DISCUSSION AND CONCLUSIONS

One of the most promising uses of semantic knowledge in a robotic system is to resolve situations of conflict of ambiguity by reasoning about the cause of the problem and its possible solutions. This paper has explored an often neglected aspect of this use: recognizing and correcting situation in the world that do not comply with the given semantic model, by generating appropriate goals for the robot. A distinctive feature of our approach is the normative model is provided in a declarative way, rather than by exhaustive violation-action rules. Experiments carried out on a real mobile robot demonstrate the conceptual viability of this approach.

The work reported here is a first step in an interesting direction, and many extensions can and should be considered. For instance, in our work we assume that the robot should *always* enforce consistency with the semantic knowledge. However, there are cases where norm violation might be allowed. Going back to the milk example, it would be reasonable to allow that the milk bottle stays out of the fridge for some amount of time while the user is having breakfast. We speculate that our scheme for automatic goal generation can be extended to also cope with this and other issues.

REFERENCES

[1] R.C. Arkin, M. Fujita, T. Takagi, and R. Hasegawa. An ethological and emotional basis for human-robot interaction. *Robotics and Autonomous Systems, 42(3-4).*, 42(3-4), 2003.

[2] F. Baader, D. Calvanese, D.L.D McGuinness, and D. Nardi, editors. *The Description Logic Handbook*. Cambridge University Press, 2007.

[3] Chitta Baral, Thomas Eiter, Marcus Bjäreland, and Mutsumi Nakamura. Maintenance goals of agents in a dynamic environment: Formulation and policy construction. *Artif. Intell.*, 172(12-13):1429–1469, 2008.

[4] Guido Boella and Rossana Damiano. An architecture for normative reactive agents. In *Proceedings of the 5th Pacific Rim International Workshop on Multi Agents*, pages 1–17, London, 2002. Springer.

[5] M. Bordignon, J. Rashid, M. Broxvall, and A. Saffiotti. Seamless integration of robots and tiny embedded devices in a peis-ecology. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). San Diego, CA*, 2007.

[6] A. Bouguerra, L. Karlsson, and A. Saffiotti. Semantic knowledge-based execution monitoring for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation, Rome, Italy*, pages 3693–3698, 2007.

[7] Alexandra M. Coddington, Maria Fox, Jonathan Gough, Derek Long, and Ivan Serina. Madbot: A motivated and goal directed robot. In *Proc., The Twentieth National Conference on Artificial Intelligence, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 1680–1681, 2005.

[8] Mehdi Dastani and Leendert W. N. van der Torre. What is a normative goal?: Towards goal-based normative agent architectures. In *Regulated Agent-Based Social Systems, First International Workshop, RASTA 2002, Bologna, Italy, July 16, 2002*, pages 210–227, 2002.

[9] C. Galindo, J.A. Fernandez-Madrigal, and J. Gonzalez. *Multiple Abstraction Hierarchies for Mobile Robot Operation in Large Environments*. Studies in Computational Intelligence, Vol. 68. Springer, 2007.

[10] C. Galindo, J.A. Fernandez-Madrigal, J. Gonzalez, and A. Saffiotti. Robot task planning using semantic maps. *Robotics and Autonomous Systems*, 56(11):955–966, 2008.

[11] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernandez-Madrigal, and J. Gonzalez. Multi-hierarchical semantic maps for mobile robotics. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3492–3497. IROS, Edmonton, Alberta (Canada), 2005.

[12] Koen V. Hindriks and M. Birna van Riemsdijk. Satisfying maintenance goals. In *Declarative Agent Languages and Technologies V, 5th International Workshop, DALT 2007, Honolulu, HI, USA, May 14, 2007*, pages 86–103, 2007.

[13] J. Hertzberg and A. Saffiotti, editors. Special issue on semantic knowledge in robotics. *Robotics and Autonomous Systems*, 56(11), 2008.

[14] L. Karlsson. Conditional progressive planning under uncertainty. In *Proc of the Int Joint Conf on Artificial Intell. (IJCAI)*, pages 431–438, Seattle, USA, 2001.

[15] B.J. Kuiper. *Modeling Spatial Knowledge*, chapter Advances in Spatial Reasoning, Vol. 2, pages 171–198. The University of Chicago Press, 1990.

[16] R. Lundh, L. Karlsson, and A. Saffiotti. Plan-based configuration of an ecology of robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), Rome, Italy*, pages 64–70, 2007.

[17] R. MacGregor and R. Bates. The loom knowledge representation language. Technical report, DTIC Research Report ADA183415, 1987.

[18] D. Meger, P. Forssen, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. Little, D. Lowe, and B. Dow. Curious george: An attentive semantic robot. In *IROS Workshop: From sensors to human spatial concepts*, pages 390–404, 2007.

[19] O.M. Mozos, P. Jensfelt, H. Zender, M. Kruijff, and W. Burgard. From labels to semantics: An integrated system for conceptual spatial representations of indoor environments for mobile robots. In *ICRA Workshop: Semantic Information in Robotics*, 2007.

[20] Timothy J. Norman and Derek Long. Alarms: An implementation of motivated agency. In *Intelligent Agents II, Agent Theories, Architectures, and Languages, IJCAI '95, Workshop (ATAL), Montreal, Canada, August 19-20*, pages 219–234, 1995.

[21] A. Nüchter, O. Wulf, K. Lingemann, J. Hertzberg, B. Wagner, and H. Surmann. 3D mapping with semantic knowledge. In *RoboCup Int. Symp.*, pages 335–346, 2005.

[22] A. Ranganathan and F. Dellaert. Semantic modeling of places using objects. In *Robotics: Science and Systems Conf.*, 2007.

[23] A. Saffiotti, M. Broxvall, M. Gritti, K. LeBlanc, R. Lundh, J. Rashid, B.S. Seo, and Y.J. Cho. The peis-ecology project: vision and results. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). Nice, France*, 2008.