Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 35 (2014) 1023 – 1032

Procedia Computer Science

18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems - KES2014

Automation in handling uncertainty in semantic-knowledge based robotic task-planning by using markov logic networks

Ahmed Al-Moadhen[a,b,*], Michael Packianather[a], Rossi Setchi[a], and Renxi Qiu[a]

[a]*School of Engineering, Cardiff University, Cardiff, CF24 3AA, UK*
[b]*College of Engineering, Karbala University, Karbala, Iraq*

Abstract

Generating plans in real world environments by mobile robot planner is a challenging task due to the uncertainty and environment dynamics. Therefore, task-planning should take in its consideration these issues when generating plans. Semantic knowledge domain has been proposed as a source of information for deriving implicit information and generating semantic plans. This paper extends the Semantic-Knowledge Based (SKB) plan generation to take into account the uncertainty in existing of objects, with their types and properties, and proposes a new approach to construct plans based on probabilistic values which are derived from Markov Logic Networks (MLN). An MLN module is established for probabilistic learning and inferencing together with semantic information to provide a basis for plausible learning and reasoning services in supporting of robot task-planning. In addition, an algorithm has been devised to construct MLN from semantic knowledge. By providing a means of modeling uncertainty in system architecture, task-planning serves as a supporting tool for robotic applications that can benefit from probabilistic inference within a semantic domain. This approach is illustrated using test scenarios run in a domestic environment using a mobile robot.

Keywords: Task-Planning; Semantic-Knowledge Based; Planning Under Uncertainty; Markov Logic Networks

1. Introduction

Plan generation by an autonomous robot planner acting in human environments is a complex task as it involves dealing with uncertainty and undeterministic situations in which a variable number of objects may be relevant to its

* Corresponding author. Tel.: +44-743-851-6890.
E-mail address: Al-MoadhenAA@cardiff.ac.uk.

doi:10.1016/j.procs.2014.08.188

tasks and these objects may be related in various ways. Uncertainty is a feature related to robots acting in real environments, and may sometimes cause failure in robot operation. To deal with unexpected planning contingencies, robotic task-planning employs probabilistic inference procedures, based on reasoning techniques or learned models, to make sure the generation of its plans do not deviate from their intended course of action[1]. Most approaches to plans generation are focusing on deterministic information about the robot environment, such as exact objects and their properties, and explicit actions' details (pre-conditions and post-conditions). For instance, in the kitchen, there is a fridge and oven and the explicit effect of grasping a glass of water would be that the robot is holding a glass of water.

In a real world environment this is not always realistic as planning with uncertainty is a complex process. Therefore, more advanced forms of probabilistic reasoning engaging with semantic knowledge should be proposed to derive probabilistic implicit information which is related to the existence of objects and predicting their types. For example, a robot moving into a living room could be expected to see at least a tv-set and a sofa. If the robot is entering a kitchen instead, it should have more probability to see a fridge and a sink. These probabilities are details that would add more complexity to the task planner, if reasoning about them has been left to it. Therefore, it is important to build a separate unit, from semantic-knowledge base and the action descriptions uses by the planner, which has the ability to learn from a knowledge base and then infer probable information about the robot environment to support its task-planner.

In this paper, the robot semantic-knowledge base is represented with Description Logic (DL)[2], which has the ability to infer the types of things and the automatic classification of things based on their classes and properties. This approach enables the robot to derive new statements from its existing knowledge. Pure description logics inference is completely deterministic, so it is often desirable to represent uncertain information in a way that can be more useful to robot planner.

Markov Logic Networks allow to draw probabilistic inferences that combine the expressiveness of first-order logics with the representation of uncertainty[3]. The influence of such relational structures involving variable sets of objects on the propositions relevant to the robot's tasks clearly cannot be accounted for a model that involves a fixed set of propositions. What is needed, therefore, is a unification of the principles of first-order logic, which makes objects and relations the main building blocks of the representation, and probabilistic graphical models, which enable reasoning in the face of uncertainty, within a single representation formalism. This approach allows, by combining the respective semantics, to obtain a language that possesses a level of expressiveness that is sufficient for robots to be equipped with the much-needed capability of reasoning about situations when they arise in real-world applications.

The main contributions of this paper are (i) constructing MLN from semantic-knowledge base to support the robot task planner in uncertain situations about objects existence and their types, and (ii) devising an algorithm which has a role to create MLN from semantic-knowledge base. The MLN template will be learned by training data from SKB, then the learned MLN model is used to answer queries generated from the knowledge base and predict the existence and types of objects or places.

The rest of this paper is organized as follows. Section 2 describes related works in the field of planning with a focus on deterministic and undeterministic approaches. Section 3 briefly describes planning under deterministic conditions. Section 4 explores planning under uncertainty and introduces the proposed approach to address this issue. Section 5 outlines an example showing the applicability of the MLN model. Finally, Section 6 provides conclusions and suggestions for future work.

2. Related works

Planning under deterministic conditions suffers from a few number of issues compared to planning under undeterministic conditions. The planning domains can be generated by integrating semantic action models with common-sense knowledge base[4]. These domains are input into a robot planner to generate a suitable plan for a given tasks.

Semantic knowledge can be successfully used to support robot task-planning. Previous research[5] included defining a specific type of semantic maps, which integrate hierarchical spatial information and semantic knowledge. This approach is dependent on asserted information and does not take into consideration the uncertainty of robot operation. Another approach[6] had proposed a hierarchical task-planning that handled uncertainty in both the state of the world and the effect of actions. It introduced mechanisms to handle situations with incomplete and uncertain information about the state of the environment by using belief states to represent incomplete information about the state of the world, and actions are allowed to have stochastic outcomes.

Partially observable Markov decision processes (POMDPs)[7] provided a principled general framework for robot motion planning in uncertain and dynamic environments. Furthermore, POMDPs had been used in[8] for motion planning under uncertainty. Semantic information can be used as a tool to improve the task-planning in complex scenarios where other planners easily may find themselves in intractable situations[9]. The approach involved constructing a "semantic" plan composed of categories of objects, places, etc. that solved a "generalized" version of the requested task, and then using that plan for discarding irrelevant information in the definitive planning carried out on the symbolic instances of those elements (that correspond to physical elements of the world with which the robot can operate).

Further work in this area[10] included developing a formalism of symbolic model of the environment to solve the issues of processing large amounts of information in planning and being efficient in human–machine communication in a natural form through a human-inspired mechanism that structures knowledge in multiple hierarchies. Planning with a hierarchical model may be efficient even in cases where the lack of hierarchical information would make it intractable. However, this method was depended on deterministic information.

To deal with uncertainties, probabilistic methods can be used to link the system variables between each other and construct a network among these variables. Markov Logic Networks[3] and Bayesian Logic Networks[11] can be utilized to process probabilistic information to support many applications such as task-planning.

3. Semantic-Knowledge Based (SKB) Planning System

Semantic knowledge refers to knowledge about objects, their classes and properties, and the relationships between these objects. The representation that refers to this structure is called ontology. Fig. 1 shows the ontology of robot environment. For instance, a living room is a concept (class) refers to a room, which has at least a tv-set and a sofa. The entities tv-set and sofa are themselves defined as pieces of furniture.

It can be shown from previous research[4], the Semantic-knowledge base (SKB) can be used to support robot task-planning to generate semantic plans for a given tasks. Semantic knowledge is used in the process of generating symbolic plans, i.e. symbols are used to refer to physical objects which will be manipulated by actions. An example of such an action is grasp m where m is a symbol that refers to a milk box.

In this paper, the use of Description Logic (DL)[2] is proposed for representing and managing the semantic domain knowledge, since it provides a good compromise between representation power and reasoning tractability. An important characteristic of DL is their reasoning capabilities of inferring implicit knowledge from the explicitly represented knowledge.

3.1 Robot Environment Ontology

A large amount of knowledge about objects and their properties in a robot environment can assist the robot to learn about the structure of its environment. This knowledge can be represented as an ontology that contains classes of all the relevant types of objects and their properties and relations.

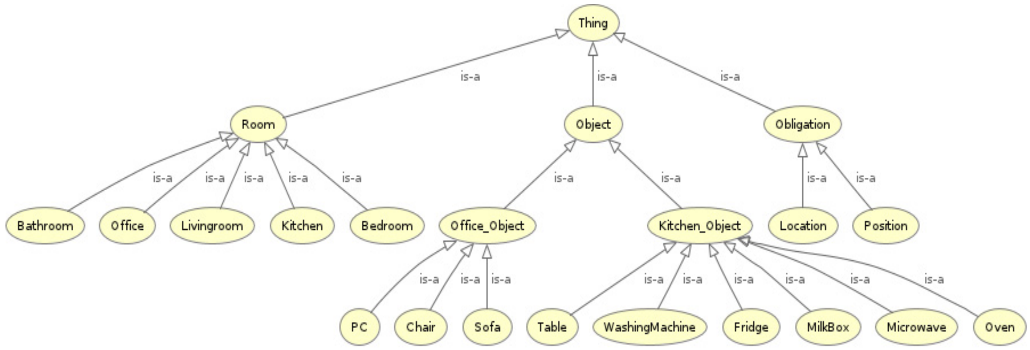

Fig. 1. Robot World Ontology

Description Logic (DL) is used as a formalism to represent the robot's knowledge about its world. The Web Ontology Language (OWL)[12] has knowledge representation format suitable to describe robot knowledge about its environment. It is used for storing Description Logic formulas in an XML-based file format. There are two types of knowledge in Description Logic. The first one is called terminological knowledge, the so-called TBOX, and the second one is called assertion knowledge, the ABOX. The TBOX contains definitions of concepts, for example the concepts Room, Action or Object. These concepts are arranged in a hierarchical way, which is called classification, using subclass definitions that describe, for instance, that a Table is a subclass of Furniture. The ABOX contains individuals which are instantiated from TBOX concepts, for example, a table1 is an instantiation of the concept Table. Individuals properties in knowledge base can be described by roles which reflect the relations between two individuals, and can also be used in class definitions to constraint the extent of a class to individuals having certain properties e.g. the concept GrasspingABottle can be described as a subclass of GraspSomething with the restriction that the objectActedOn has to be some instance of Bottle. The following examples (which are based on Fig. 1) describe the definition of atomic classes and defined classes. The roles are used to restrict the extension of some classes.

Atomic classes: Room, Object.
Defined classes: **Kitchen** is a **Room** and is*Containing* **KitchenObject**
KitchenObject is an **Object** and is*LocatedAt* **Kitchen**

3.2 Deterministic Planning System

The planning method described in[4] deals with deterministic knowledge which is stored in a semantic-knowledge base. It is used as static information by the planner to generate its plans. The next section extends this planning approach by supporting the planning system with probabilistic capabilities.

4. Planning under uncertainty

In this section, semantic-knowledge base is proposed to support robot planning system by handling the issues of uncertainty in existence of objects in a certain place or predicting the types of objects or places. For instance, to insert an action *move bedroom1 kitchen1* in the robot plan, it might be necessary to provide robot planner with information about the type of the next room that robot should go to. If the task requires the robot to fetch a milk box and there is no assertion about the milk box location, it is necessary for the planner to get support from the probabilistic module. If the probabilistic module answers with high probability that the most probable location for a milk box is the kitchen because it is one of kitchen objects, so the best next location in the move action is the kitchen. Planning involving uncertainty in predicting the types of objects or places is conducted in the space of weights associated with a formula that describes them.

In order to support the task planner efficiently, probabilistic models are needed to represent the probability distributions of uncertain information in the planning domain. There are several statistical relational models represented as extensions of undirected[3] or directed graphical models (Multi-Entity Bayesian Networks[13] or Bayesian Logic networks[14]. These models can make using of learning and inference methods which have been developed for their underlying representations. Markov Logic Networks are used in this paper as a statistical relational model to support the task-planning process. Fig. 2 represents the probabilistic planning system architecture proposed in this research.

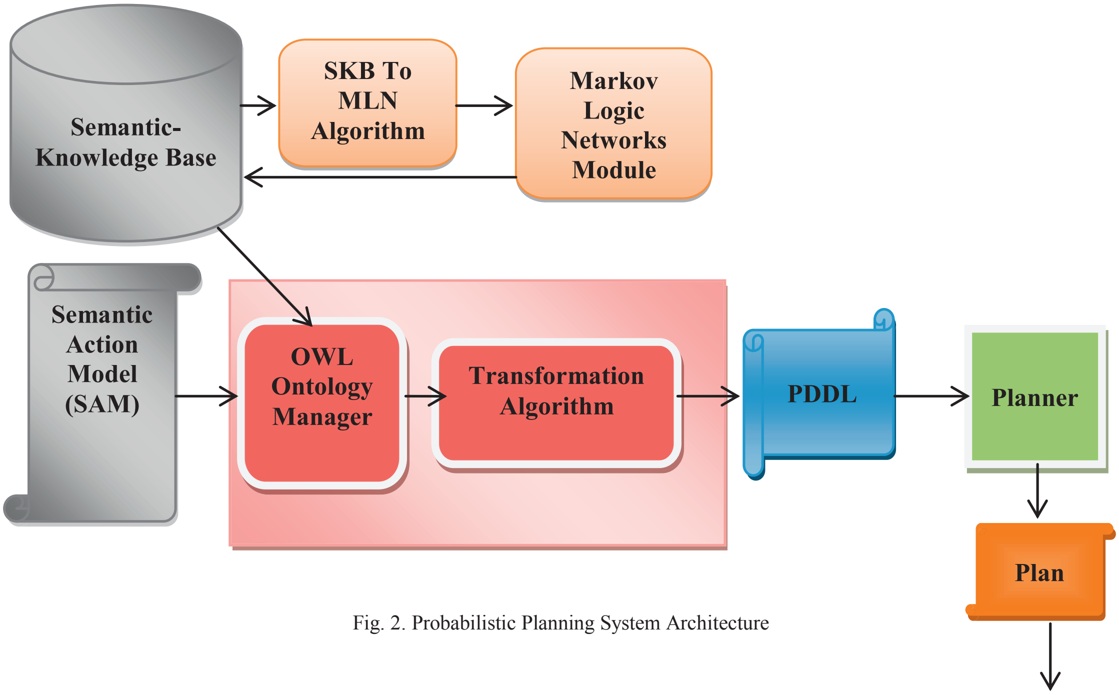

Fig. 2. Probabilistic Planning System Architecture

4.1 Markov Logic Networks (MLNs)

The formal definition of MLN L is given by a set of pairs $<F_i, w_i>$, where F_i is a formula in first-order logic and w_i is a real-valued weight. For each finite domain of constants D, an MLN L defines a *ground Markov network* $M_{L,D} = <X,G>$ as follows (see[11] for more details):

- X is a set of boolean variables. For each possible grounding of each predicate appearing in L, add a boolean variable (ground atom) to X.
- G is a set of weighted ground formulas, i.e. a set of pairs $<F_j', w_j'>$, where F_j' is a ground formula and w_j' is a real-valued weight.

The ground Markov network $M_{L,D}$ specifies a probability distribution over the set of possible worlds X as follows:

$$P(X = x) = \frac{1}{Z}\exp\left(\sum_{i=1}^{|L|} w_i n_i(x)\right) = \frac{1}{Z}\exp\left(\sum_{j=1}^{|G|} w'_j f'_j(x)\right) \tag{1}$$

$$Z = \sum_{x' \in X} \exp\left(\sum_i w_i n_i(x')\right) = \sum_{x' \in X} \exp\left(\sum_j w_j' f'_j(x)\right) \tag{2}$$

4.2 Learning MLN

The learning of a statistical relational model involves the construction of a model from observed training data. The structure of the model can either be known a priori, leaving only the parameters to be determined, as shown in[15], or can be part of the learning problem. One consequently differentiates parameter learning from the harder problem of structure learning. The first approach towards learning the structure of MLNs was presented by[15]. While structure learning is clearly important if Artificial Intelligent (AI) systems are to build up probabilistic models with as little human assistance as possible, realistic approaches will still rely on engineered structures for the most part. Parameter learning, therefore, is the most important aspect for knowledge engineers who typically qualitatively assess the properties of a distribution and indicate the dependencies between the variables but cannot quantitatively define the degree to which these variables depend on one another.

In a Markov logic network, the goal of parameter learning is to set the weights of the model's formulas such that they reflect observations that have been made about the particular part of the world the model is concerned with. The observations that were made are representative of the particular aspects of the world that are to be captured by the model, such that they allow the model to extract precisely the general principles. The observations used for learning can be stored in a training database that uses the same language as the model. Since MLNs use logical predicates, the database should thus contain the truth values of a number of ground atoms. The entities appearing in the training database implicitly define a set X of ground atoms. Any ground atoms in X whose truth values are not given in the training database are assumed to be false (closed world assumption). Under this assumption, the training database thus specifies a full assignment X = x.

Maximum Pseudo-Likelihood Learning: This problem can be addressed by maximum Pseudo-Likelihood.

$$P^*(X = x) = \prod_{k=1}^{|x|} P(X_k = x_k \mid MB_x(X_k)) \tag{3}$$

where X_k is a ground atom, x_k is X_k's truth value in x, and MB_x (X_k) is the assignment of X_k's Markov blanket in x[3]. The pseudo-likelihood approximates $P(X = x)$ by making strong independence assumptions, avoiding an expensive inference process.

Discriminative Learning. The most attractive feature of undirected models and therefore MLNs is that they can also be trained discriminatively, i.e. they can be trained to represent not a full-joint distribution but a conditional distribution. Assuming that there is a strict separation between observable and unobservable variables in the application at hand (e.g. a classification task, where only the classes are always unknown but everything else is given), discriminative training can yield models with superior performance. Methods for the discriminative training of MLNs were introduced by[16]. Fig. 3 explains the components of the MLN learning phase.

4.3 MLN as inference engine

The probabilistic semantics of Markov logic networks are defined via ground Markov random fields, so inference can essentially be handled by applying standard inference methods for Markov networks.

Markov chain Monte Carlo (MCMC) scheme[17] methods are essentially approximate, sampling-based methods where the individual samples are not drawn independently but taken from a Markov chain, i.e. a model describing sequences of states. Fig. 4 explains the components of MLN query phase.

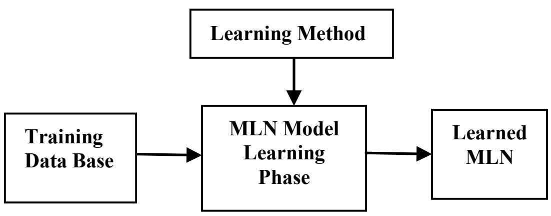

Fig. 3. MLN Learning Phase

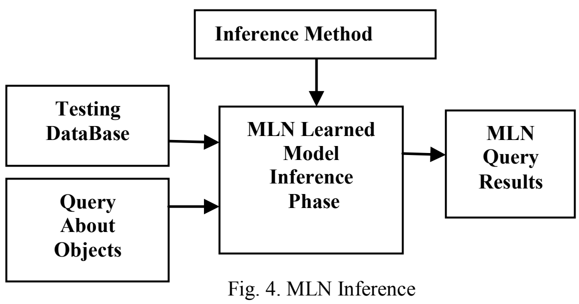

Fig. 4. MLN Inference (Query) Phase

4.4 SKB To MLN Algorithm

The algorithm proposed to create MLN from SKB is as follows:

Algorithm1:
Input: SKB Classes (C), Roles (R)
Output: MLN
P = null
F = null
For every c in C
    P = P + Create predicate (c)
For every r in R
    F = F + Create formula (r)
MLN = combine (P + F)

4.5 Example

Suppose that the planner has to insert the movement action *move r2 r1* (where r2 is a living-room and r1 may be a bedroom or an office), so there are two possible values to r1. Then r1 value depends on the result of the query that is returned from the learned MLN model. If the result shows that r1 could be a bedroom with probability of 0.2 or an office with probability of 0.8, then the move action should be *move r2 r1* with r1 = office.

5. Testing scenario

The test scenario involves performing a navigation task in a house environment. This house consists of 4 rooms named k1, k2, o1 and o2 as shown in Fig. 5. These rooms have not been identified yet and the robot depends on the MLN module to identify them. The semantic-knowledge base contains, among other things, the following classes and roles definitions (see Fig. 1):

Classes
Room, Object, Kitchen
Office, KitchenObject, OfficeObject

Rules
Kitchen is Room and Kitchen isContaining KitchenObject
KitchenObject is Object and KitchenObject isLocatedAt Kitchen
Office is Room and Office isContaining OfficeObject
OfficeObject is Object and OfficeObject isLocatedAt Office

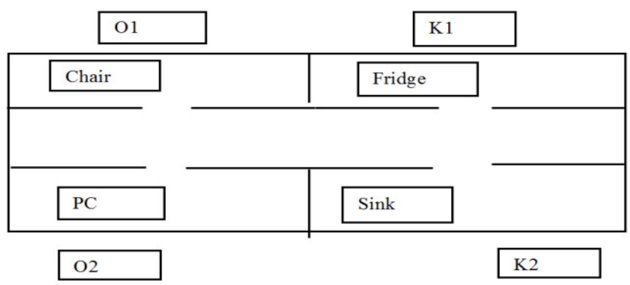

Fig. 5. Robot Environment

These definitions are then translated from SKB to MLN (using Algorithm1) to create the following MLN model:

```
Room(place), Object(entity), Kitchen(place), Office(place), KitchenObject(entity), OfficeObject(entity)
Containing(place!, entity), LocatedAt(entity, place!)
Room(x) ^ KitchenObject(y) ^ isLocatedAt(y, x) => Kitchen(x)
Room(x) ^ OfficeObject(y) ^ isLocatedAt(y, x) => Office(x)
Office(x) ^ Containing(x, y) => OfficeObject(y)
Kitchen(x) ^ Containing(x, y) => KitchenObject(y)
```

This model becomes as a template and can be trained by using training data e.g.:

```
Room(R1), Room(R2), Room(R3), Room(R4),
Kitchen(R1), Kitchen(R4),
Office(R2), Office(R3),
KitchenObject(Oven), KitchenObject(Fridge), KitchenObject(MilkBox),
OfficeObject(Pc), OfficeObject(Chair),
isContaining(R1, Oven), isContaining(R1, Fridge), isContaining(R2, Pc), isContaining(R3, Chair), isContaining(R4, MilkBox),
isLocatedAt(Oven, R1), isLocatedAt(Fridge, R1), isLocatedAt(Pc, R2), isLocatedAt(Chair, R3), isLocatedAt(MilkBox, R4)
```

This paper uses the ProbCog[18] Toolbox to train and query the MLN model. By applying this information to the model in Fig. 3, the resulting learned MLN is represented as follows (only a part of the MLN model is displayed due to the limited space):

```
1.42655  Room(x) ^ KitchenObject(y) ^ isLocatedAt(y,x) => Kitchen(x)
1.38352  Room(x) ^ OfficeObject(y) ^ isLocatedAt(y,x) => Office(x)
1.41033  Office(x) ^ isContaining(x,y) => OfficeObject(y)
1.39625  Kitchen(x) ^ isContaining(x,y) => KitchenObject(y)
-0.185971  Kitchen(a1)
 0.185143  Office(a1)
```

Next, this learned MLN model is used to predict and infer the probability of objects existence and their types. For example, if the testing data is:

```
Room(K1),Room(K2),Room(O1),Room(O2),KitchenObject(Table),KitchenObject(Microwave),KitchenObject(WashingMachine),Of
ficeObject(Chair),OfficeObject(Sofa),OfficeObject(PC),isContaining(K1,Table),isContaining(O1,Chair),isContaining(O2,PC),isCon
taining(K2,Microwave),isContaining(K2,WashingMachine),isContaining(O2,Sofa),isLocatedAt(Table,K1),isLocatedAt(Chair,O1),is
LocatedAt(PC,O2),isLocatedAt(Microwave,K2),isLocatedAt(WashingMachine,K2),isLocatedAt(Sofa,O2).
```

and by inputting it to the model in Fig. 4 and selecting MCMC inference method[17], then the result is shown in table 1:

Table 1. Places and their probabilities

No.	Things	Probability
1	Kitchen(K1)	0.791971
2	Kitchen(K2)	0.779972
3	Kitchen(O1)	0.172033
4	Kitchen(O2)	0.154035
5	Office(K1)	0.225027
6	Office(K2)	0.239026
7	Office(O1)	0.834967
8	Office(O2)	0.823968

From table1, it is important to calculate the standard deviation to show the best selection of the Kitchen or Office places. The mean value for Kitchen probability is 0.474502, the variance is 0.097071 and the standard deviation is 0.311562, as shown in Fig. 6.

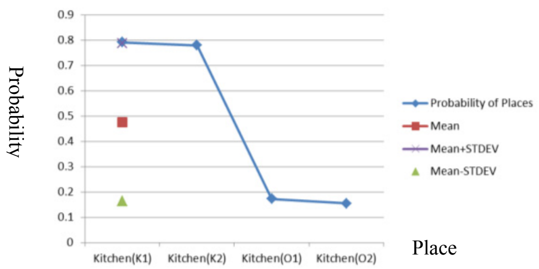

Fig. 6. Places Probability, Mean and Standard Deviation

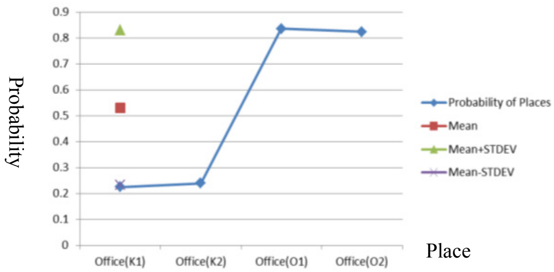

Fig. 7. Office Places Probability, Mean and Standard Deviation

So, that K1 and K2 are more probable to be kitchen than office. The same steps can be done with office place and Fig. 7 can explain that.

From this result, a decision can be taken to specify the type of rooms (in uncertain situations) according to the objects inside them. After this procedure, the robot task planner can use this information to generate its plans, so the uncertainty is handled in robot task planner.

6. Conclusions and future works

This paper developes a new method to enable the robot to deal with uncertainty. A template of the MLN model is created based on information stored in the semantic-knowledge base. Then this model is trained in order to get learned MLN, which has the ability to answer any query that is generated from the SKB and can infer (in a probabilistic way) the types and the existence of objects or places in the robot environment.

Future work includes obtaining more training data in order to cover most of the situations that may face the robot operations and using it to learn the model and recover from situations when the domain parameters are not deterministic. The use of other types of statistical relational model such as Bayesian Logic Netwoks in supporting of the task planner will also be considered.

References

1. Bouguerra A, Karlsson L, Saffiotti A. *Handling Uncertainty in Semantic-Knowledge Based Execution Monitoring*. Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems San Diego, CA, USA, 2007.
2. Baader F, Calvanese D, McGuinness D L, Nardi D, Patel-Schneider P F. *The Description Logic Handbook: Theory, Implementation and Applications*. Second Edition, 2007.
3. M. Richardson and P. Domingos. *Markov logic networks*. Machine Learning 2006; 62: 107-136.
4. Al-Moadhen A, Qiu R, Packianather M, Ji Z, Setchi R. *Integrating Robot Task Planner with Common-sense Knowledge Base to Improve the Efficiency of Planning*. Procedia Computer Science 2013; 22: 211–220.
5. Galindo C, Fernández-Madrigal JA, González J, Saffiotti A. *Robot Task Planning Using Semantic Maps*. Robotics and Autonomous Systems 2008; 56(11): 955–966.
6. Bouguerra A, Karlsson L. *Hierarchical task planning under uncertainty*. In 3rd Italian Workshop on Planning and Scheduling, 2004.
7. Papadimitriou C, Tsitsiklis J. *The complexity of Markov decision processes*. Mathematics of Operations Research 1987; 12(3).
8.Ong SCW, Png SW, Hsu D, Lee WS. *Planning under Uncertainty for Robotic Tasks with Mixed Observability*. International Journal of Robotics Research 2010; 29(8): 1053–1068.
9. Galindo C, Fernández-Madrigal J.-A, González J, Saffiotti A. *Using semantic information for improving efficiency of robot task planning*. ICRA-07 Workshop on Semantic Information in Robotics 2007, Italy.
10. Galindo C, Fernández-Madrigal J.-A, González J. *Interactive Task Planning through Multiple Abstraction: Application to Assistant Robotics*. ECAI 2004, 1015-1016.

11. Jain D. *Knowledge engineering with markov logic networks: A review*. Proceedings of the Third Workshop on Dynamics of Knowledge and Belief, 2011.
12. *OWL Web Ontology Language Reference*. [Online]. Available: http://www.w3.org/TR/owl-ref/. [Accessed: 15-Dec-2013].
13. Laskey KB. *MEBN: A language for first-order Bayesian knowledge bases*. Artificial Intelligence 2008; 172(2–3): 140–178.
14. Jain D, Waldherr S, Beetz M. *Bayesian Logic Networks. Technical report*. Technische Universität München 2009.
15. Kok S, Domingos P. *Learning the structure of Markov logic networks*. Proceedings of the 22nd international conference on Machine learning 2005, 441- 448.
16. Singla P, Domingos P. *Discriminative training of Markov logic networks*. Proceedings of the 20th national conference on Artificial intelligence 2005; 2:868-873.
17. Koller D, Friedman N. *Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning)*. MIT Press, 2009.
18. *Intelligent Autonomous Systems - ProbCog Toolbox*. [Online]. Available: http://ias.in.tum.de/software/probcog. [Accessed: 21-Mar-2014].