Lección 15: Interface Hombre-Robot y Representación del Conocimiento

Jesús Savage

6 de mayo de 2020

Índice

Introducción

En esta lección se encontrarán cuales son los pasos que se tienen que hacer para crear un representación del conocimiento e interface hombre-robot.

A Robot in Every Home: Overview/The Robotic Future. Bill Gates, Scientific American (2008)

Knowledge Representation

In a hybrid robotics architecture, an expert system maintains a knowledge database representing the state of the world. The data of humans, objects, furniture, rooms, and robots are represented using template facts that contain several slots with information related to them. Basically there are five fact templates: Human, Object, Room, Furniture, and Robot.
The Human template contains all the information related of the humans that interact with the robot. Some of its slots are:

```
(Human
   (name identification)
   (room last-known-room-location)
   (zone last-known-zone-inside-room)
   (objects last-know-objects-asked)
   (pose last-known human's-pose)
   (locations possible-locations)
)
```

Knowledge Representation

The slot "pose" represents the x, y, and z coordinates, in meters, of the human with respect of the room coordinates. The slot "locations" represents the possible rooms where the person could be if she is not present in the last known room. The robot uses these rooms to search for the person.
For example, a human identified as the Mother is represented as:

```
(Human
  (name Mother)
  (room studio)
  (zone couch)
  (objects book)
  (pose 1.8 2.0 0.5)
  (locations main-bedroom kitchen living-room)
)
```

Knowledge Representation

The Object template contains the information related to the entities on which the robot can perform actions. Some of its slots are:

```
(Object
  (name identification)
  (room room-location)
  (zone zone-inside-room)
  (pose object's-pose)
  (possession to-whom-belongs)
  (use object's-use)
  (locations possible-locations)
  (attributes object's-attributes)
)
```

Knowledge Representation

Example:

```
(Object
  (name newspaper)
  (room outside)
  (zone outside-door)
  (pose 0.5 1.3 0.1)
  (possession nobody)
  (use information)
  (locations living kitchen)
  (attributes movable)
)
```

Knowledge Representation

The Furniture template contains the information related to the furniture in the rooms. Some of its slots are:

```
(Furniture
  (name identification)
  (room room-location)
  (zone zone-inside-room)
  (pose furniture's-pose)
  (use furniture-use)
  (locations possible-locations)
  (attributes object's-attributes)
)
```

Knowledge Representation

Example:

```
(Furniture
  (name kitchen-table)
  (room kitchen)
  (zone left-side)
  (pose 0.2 2.3 1.0)
  (use eating)
  (attributes fixed)
)
```

Knowledge Representation

The Rooms template contains all the information related to the rooms where the robot performs actions. Some of its slots are:

(Room
  (name identification)
  (location room's-location)
  (center-coordinates-center-of-the-room)
  (objects objects-inside-the-room)
  (furniture furniture-inside-the-room)
  (use actions-that-can-be-done)
  (humans humans-in-the-room)
)

Knowledge Representation

Example:

```
(Room
  (name kitchen)
  (location right main-entrance)
  (center-coordinates 2.3 3.0)
  (objects knife plate plates glasses food water apples)
  (furniture stove fridge kitchen-table chairs)
  (use breakfast dinning lunch cleaning eating)
  (humans Peter Mary)
)
```

Knowledge Representation

The Robot template contains all the information related to the robots.
Some of its slots are:

```
(Robot
  (name robot-identification)
  (room room-location)
  (zone zone-inside-room)
  (pose robot's-pose)
  (objects objects-carrying-the-robot)
)
```

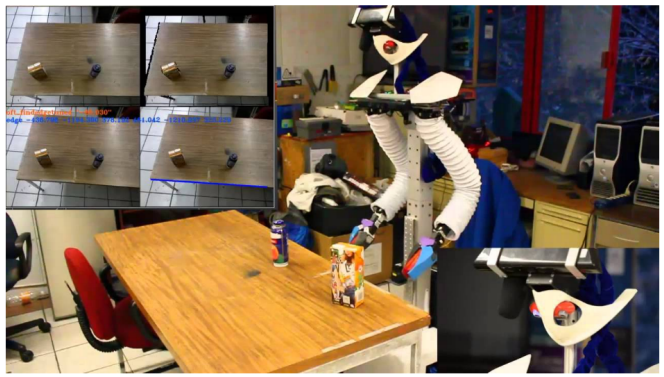
Knowledge Representation

Example:

```
(Robot
  (name Justina)
  (room living-room)
  (zone sofa)
  (objects none)
)
```

Knowledge Representation

After applying digital signal processing techniques, a symbolic representation of the data coming from the sensors is obtained. This symbolic representation is used to modify the knowledge representation by updating some of the five fact templates: Human, Object, Room, Furniture, and Robot.



Natural Language Understanding

One way to represent a spoken command is by describing the relationships of objects mentioned in the input sentence. During this process the main event described in the sentence and participants are found. In this work the participants are any actors and recipients of the actions. The roles the participants play in the event are determined, as are the conditions under which the event took place. The key verb in the sentence can be used to associate the structure to be filled by the event participants, objects, actions, and the relationship between them.

Natural Language Understanding

The natural language understanding system allows a robot to translate voice commands into an unambiguous representation that helps an inference engine to do action and movement planning. The following considerations were taken into account to generate an unambiguous representation: generation of metadata of words and constituents. Extracting valuable information from the sentences requires syntactic, semantic analysis and queries of the state of the world. The syntactic structure of the statements indicates the way in which the words relate to one another. In particular our robot must interpret commands within known contexts.

Natural Language Understanding

The process of understanding natural language can be decompose into a number of steps:

1.- Input Signal .- Either speech or text coming from a keyboard. This input signal is transformed into basic units (words) for the next steps.

2.- Syntactic Analysis .- In this step the inputs words are tested if they are grouped according to grammatical rules, meaning that they form meaningful sentences.

3.- Semantic Analysis .- In this step the meaning of each word and sentence is assigned. This one of the most complicated part of the three steps, and unless is a very simple problem domain, it requires a big knowledge data base about the topic being discussed.

Natural Language Understanding

In the process of syntactic analysis for each recognized word should be labeled with a grammatical category, the most commons are: nouns, verbs, adjectives and adverbs. This type of grammatical categories are known as open categories since words change over time as well as new words appear. In the grammar category labeling there is a table with word and category pairs, but due to the ambiguous nature of natural language, a word can work in different categories depending on previous or subsequent words. To solve the ambiguity, an algorithm based on rules of the type is used: a preposition precedes a determinant and an adjective precedes a noun. In addition, there are rules of the form a preposition of place precedes a noun of the category place. These rules are used to determine what is the correct labeling considering all possible combinations of labels

Conceptual Dependency

The theory of conceptual dependence is a representation of the meaning of an idea developed by Roger Schank This theory establishes the meaning of a sentence as a graph of dependencies between objects and semantic roles. As there are many ways of expressing the same idea, it is considered unlikely that humans will keep memory of structures highly related to natural language. Schank considers it more likely that a standard form of knowledge will be developed, where all the possible paraphrases of a statement are mapped to a canonical form of meaning. This canonical form of representation must move away from natural language by avoiding the use of words or ambiguous syntactic structures. This technique finds the structure and meaning of a sentence in a single step using Conceptual Dependency (CD) primitives. CDs are especially useful when there is not a strict sentence grammar.

Conceptual Dependency

The theory of conceptual dependence has as its premise that an action is the basis of any proposition. All propositions that describe events are made up of conceptualizations, which are formed by an action, an actor and a set of roles that depend on the action. An action is defined as something that an actor can apply to an object. Schank proposes a finite set of primitive actions that are the basic units of meaning with which a complex idea can be constructed. These primitive actions differ from the grammatical categories since they are independent elements that can be used in combination with each other to express the idea underlying a statement. One of the main advantages of CDs is that they allow a rule base systems to be built which make inferences from a natural language system in the same way humans beings do. CDs facilitate the use of inference rules because many inferences are already contained in the representation itself. The CD representation uses conceptual primitives and not the actual words contained in the sentence. These primitives represent thoughts, actions, and the relationships between them.

Conceptual Dependency

Some of the more commonly used CD primitives are, as defined by Schank:

ATRANS: Transfer of ownership, possession, or control of an object. For example, possession, ownership or control. It requires an actor, an object and a container. With this primitive act it can be coded verbs like give, take, buy.
PTRANS: Transfer of the physical location of an object. It requires an actor, an object and a destination address. Encode verbs such as fly, go, walk, drive.
ATTEND: Concentrate on perceiving a sensory stimulus, focus a sense organ (e.g. find, look.)
MOVE: Movement of a body part by its owner (e.g. kick.)
GRASP: Grasping of an object by an actor (e.g. take.)

Conceptual Dependency

PROPEL: The application of a physical force to an object. It requires an actor, an object and an address. Encode verbs like push, pull, kick, crash.

SPEAK: Production of sounds (e.g. say.)

EXPEL: Expel an object out of the body (e.g. cry.)

MTRANS: The transfer of an idea within or between animated entities (e.g. remember.)

MBUILD: Encode verbs like remember, see, tell, read (e.g. figure it out.)

INGEST: Enter one object into another (e.g. eat.)

Conceptual Dependency

Each primitive represents several verbs which have similar meaning. For instance give, buy, steal, and take have the same meaning, i.e., the transference of one object from one entity to another. Each primitive is represented by a set of rules and data structures.
Basically each primitive contains components:

1. **An Actor:** He is the one that perform the ACT.
2. **An ACT:** Performed by the actor, done to an object.
3. **An Object:** The action is performed on it.
4. **A Direction:** The location that an ACT is directed towards.
5. **A State:** The state that an object is in, and is represented using a knowledge base representation as facts in an expert system.
6. **An INSTRUMENT:** How the action is performed.
7. **TIME:** The time when the action is performed.

Conceptual Dependency

For example, when the system finds the main verb “give” in the phrase:

“Robot, please give this book to Mary”,

the system uses a lookup table, encoded in one of our expert-system rules, with this verb as key to obtain the ATRANS structure:

(ATRANS (ACTOR NIL) (OBJECT NIL) (FROM NIL) (TO NIL))

The empty slots (NIL) need to be filled finding the missing elements in the sentence.

Conceptual Dependency

The actor is the robot, the object is the book, etc., and the sentence is represented by the following CD:

(ATRANS (ACTOR Robot) (OBJECT book) (FROM book_s owner) (TO Mary))

The **book_s owner** is obtained from the knowledge base that is implemented as facts in the rule based system.

It is important to notice that the user could say more words in the sentence, such as: 'Hey Robot, please give this book to Mary, as soon as you can,' and the CD representation would be the same.
That is, there is a transformation of several possible sentences to one representation that is more suitable to be used by an action planner.

Conceptual Dependency

For sentences of stative conceptualization, **STATE**, that indicates the state of an OBJECT with some VALUE, there is no set of core states in CD, for example, "The milk is in the fridge", is represented by:

(Object (name milk) (room kitchen) (zone fridge))

Sentences that state the way actions are performed, like the following command given to the robot: "please go faster", can be encoded as conceptualizations that are attribute values statements:

(GO (OBJECT Robot) (VALUE faster))

Conceptual Dependency

CDs can also be used with multi-modal input. For instance, if the user says: "Put the newspaper over there", while pointing from the floor to the table top, separate CDs will be generated for the speech as well as gesture input with empty slots for the unknown information (assuming the newspaper was initially on the floor):

Speech: `(PTRANS (ACTOR Robot) (OBJECT Newspaper) (FROM NIL) (TO Over there))`

Gesture: `(ATTEND (ACTOR User) (OBJECT Hand) (FROM Floor) (TO Table top))`

Conceptual Dependency

Empty slots can be filled in by inspecting CDs generated by other modalities at the same time, and combining them to form a single representation of the desired command:
`(PTRANS (ACTOR Robot) (OBJECT Newspaper) (FROM Floor) (TO Table top))`

Conceptual Dependency and Keyword Speech Recognition System

In order to represent a continuous spoken sentence by CDs, some of its components that will represent it, are chosen according the results obtained by a keyword speech recognition system.
The initial structure issued to represent a sentence may contain holes (NILs) that need to be filled. These holes in the sentence's knowledge representation may represent some of the sentence's words that either are wrongly recognized or are not part of the recognition vocabulary. These knowledge holes will be filled by a mechanism in which an inference engine will use rules and context to look for the needed information.

Conceptual Dependency and Keyword Speech Recognition System

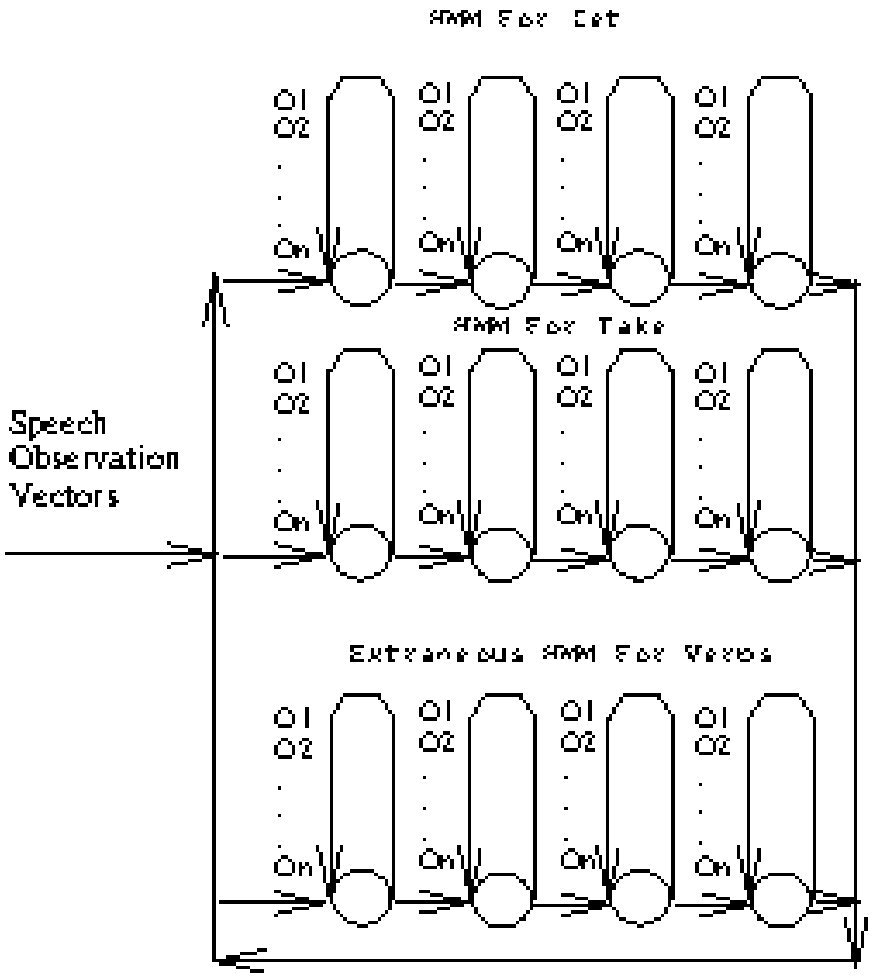
According to the conceptual dependency theory each phrase can be decomposed into several categories. Some of the categories are ACTORS, VERBS, OBJECTS, etc. Each category is represented by a connected network, in which the HMM models of each of the words that belong to a particular category are concatenated together. Also concatenated are the silence and extraneous models. The extraneous models are created using all the words that do not belong to that category. For example to create the extraneous model for the category VERBS we used all the words that are not verbs.

Conceptual Dependency and Keyword Speech Recognition System

For a given sentence we used the keyword spotting technique to find some of the members of each category embedded in it. This technique finds the VQ representation and it passes the index vectors found through the Vitterbi algorithm using each of the categories. The Vitterbi algorithm will give the best sequence of states that in turn represent the best possible words for each category.

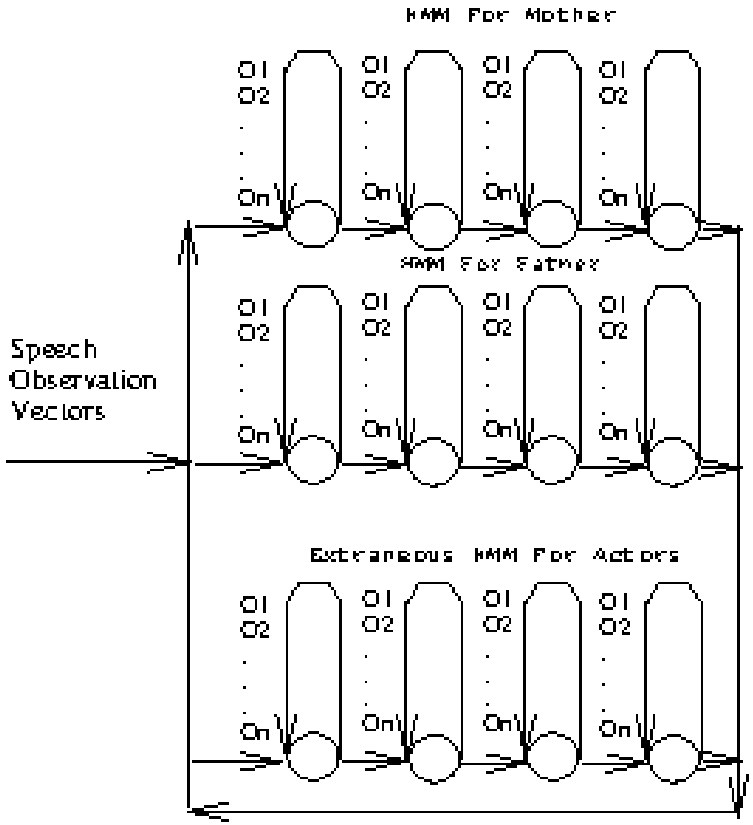
Conceptual Dependency and Keyword Speech Recognition System

Hidden Markov Model for Verbs:

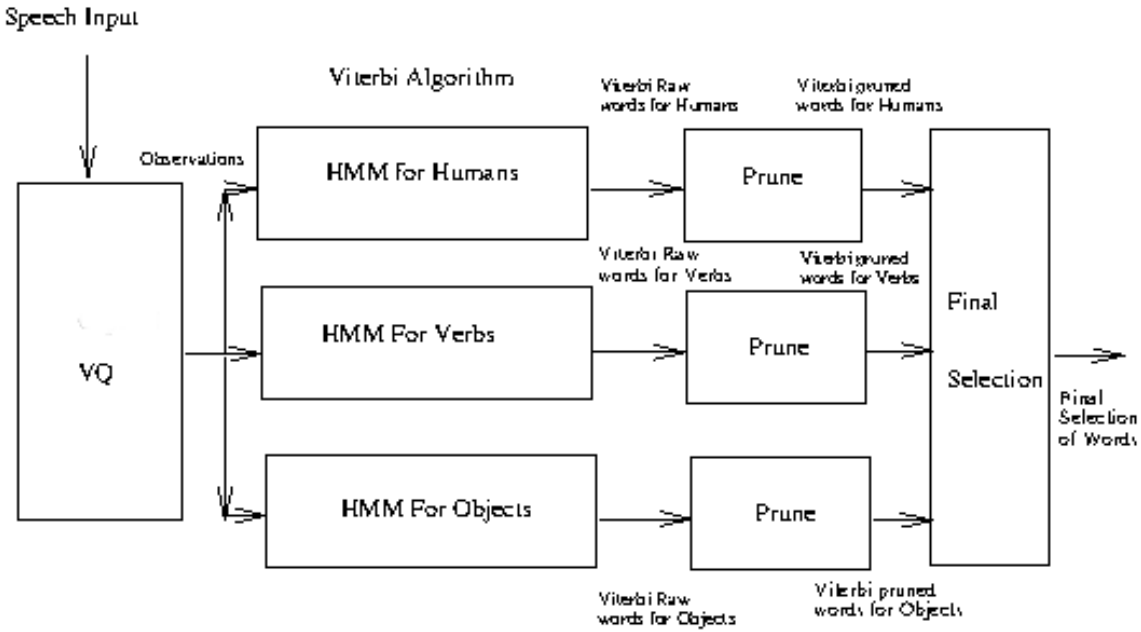


Conceptual Dependency and Keyword Speech Recognition System

Hidden Markov Model for Humans:



Conceptual Dependency and Keyword Speech Recognition System



Conceptual Dependency and Keyword Speech Recognition System

Find the elements in the provided information that will fired some of the CD primitives, ie, find the verb or question words.

Find the components of the CD in the information provided. If there is more than one word that may fill some of the slots, issue a clone of the CD primitive, and fill the elements with the repeated words.
At the end of this process we have a set of possible representations of the meaning of the sentence, and we have a decision rule to choose the one that best represent its, according to the context.

Context Representation

Context plays an important role for understanding the information provided in a conversation or talk.
The expert system maintains a data base that represents the state of the world. The expert system receives from the DSP and statistical module information from the best M words/sentences recognized. Starting with the best first word/sentence it will try to interpret it according to its data base.
If this information do not correspond or do not make any sense given a particular context, due to a bad recognition from the DSP and statistical module, then the expert system will start looking on the information of the best second word, and so on until it finds the word/sentence that make more sense.

Context Representation

If this process did not work for all M words then the expert system may require this module to check again for the missing information. As a last resort it may ask the user to repeat the phrase or to provide more information.
For example, if the user says **Robot, bring the milk** and the DSP and statistical module found the following best sentences:

1.- Robot, bring the mail
2.- Robot, bring the milk
3.- Robot, bring the ink

Context Representation

As we can see the DSP and statistical module did a wrong recognition because the first sentence is incorrect, but how the expert system decides if is it correct or not?
It will decide by looking to the context surrounding the sentences. Let say that user is in the kitchen and making the breakfast. Then from this context is more probable that the second sentence is the correct one, that is the user is asking for the milk and not for the mail. Here the DSP and statistical module failed to distinguish milk from mail.

Action Planner

When the mechanism for generating a new plan starts with a spoken command, the representation of this command should be made in a way that planning can be achieved to solve the requirement in the command. One objective of action planning is to find a sequence of physical operations to achieve the desired goal. In our architecture, after receiving the CD representation, the Planner layer attempts to plan actions according to the situation presented.

Action Planner

An action planner takes concepts from space-state search planning and hierarchical task networks. This type of planners has the following basic components:

1. A representation of the environment's state where the robot operates.
2. A set of basic operators that the robot can perform.
3. A number of truth relationships, axioms, or rules for performance.

Together, the representation of the environment, the operators, and the axioms or rules define a space-state.

Action Planner

1. For the representation of the environment we use facts templates using the expert system.
2. The basic operators that the robot can execute are: finding objects, grasping objects, moving from on place to another, finding humans, localize itself, open doors, open drawers, etc.
3. The planning rules consider different situations present in the environment, so that the robot can act accordingly. So, depending on the current situation and the currently goal, a plan is generated.

The plan specification is done through rules that represent a hierarchical structure of tasks, and each task can use several planning rules. Each planning rule once it is activated the planning rule uses operators to solve a specific problem.

Action Planner

For example, if the robot needs to pick up a specific object, then the following rule is activated if the precondition is satisfied:

Rule Pick-Object {
  Preconditions:
    Pick object X
    Manipulator empty
    Not exist another object Y above object X
  Then:
    Grasp(object X)
}

Action Planner

If the preconditions are not satisfied, then the following rule could be activated:

Rule Pick-Object-Above {
  Preconditions:
    Pick object X
    Manipulator empty
    Object Y is above object X
  Then:
    Move object Y to another place
}

Action Planner

The truth relationships, axioms, and rules for performance are encoded using the rule base system CLIPS. CLIPS transforms internally the planning rules into a tree structure and uses depth-first search to find a solution.

For example, when the user says "Give an apple to John", once the speech recognition system recognizes these words, the following CD is generated:

(ATRANS (ACTOR Robot) (OBJECT apple)
(FROM Kitchen) (TO John))

Action Planner

The ATRANS primitive requires several CD primitives to be issued:
1. The robot goes first to the kitchen and this action can be represented by the following primitive:

(PTRANS (ACTOR Robot) (OBJECT Robot)
(FROM Robot's-Place) (TO Kitchen))

which means that the robot is moving itself.

2. The robot needs to find the apple. This is represented by the following primitive:

(ATTEND (ACTOR Robot) (OBJECT apple)
(FROM Kitchen) (TO Vision-System))

Action Planner

3. The robot needs to pick up the apple:

(GRASP (ACTOR robot)(OBJECT cucumber)
(TO robot's-manipulator)(FROM kitchen))

4. The robot goes to the location where John is. This information is in the template associated with John, in the room slot (the dining room).

(PTRANS (ACTOR Robot)(OBJECT Robot)
(FROM kitchen)(TO dining-room))

5. The robot needs to find John:

(ATTEND (ACTOR Robot) (OBJECT John)
(FROM Kitchen) (TO Vision-System))

Action Planner

This is the plan represented using CD primitives. A sequence of ordered high-level tasks is determined, where the parameters corresponding to each type of action are specified. For the first `PTRANS` primitive, the motion planner, using a topological network of the free navigation space, finds the best global path between the robot's location and the kitchen. Such path planning is made using the A* algorithm. Then the robot navigates to the kitchen by reaching each node in the path. If there are unknown obstacles not considered by the planner, the robot avoids such obstacles using reactive behaviors. Once the robot is in the kitchen, with the CD primitive `ATTEND` the robot activates a behavior to *find the apple*. Once the robot finds the apple, it grasps the apple and continues with the plan until it delivers the apple to John.