Semantic Reasoning in Service Robots Using Expert Systems

Jesus Savage, Julio Cruz, Reynaldo Martell, Mauricio Matamoros, Hugo Estrada and Luis Contreras
Bio-Robotics Laboratory, School of Engineering Universidad Nacional Autónoma de México (UNAM)

***Abstract*— This paper presents the semantic module in a proposed architecture for service robots, referred as VIRBOT. In the case of natural language understanding, we show that by combining symbolic AI with digital signal processing techniques, state-of-the-art performance is obtained, and we present result obtained in international competitions, such as RoboCup and RockIn.**

***Keywords*—** *Service robots, Semantic Reasoning, Knowledge Representation, Expert Systems.*

I. Introduction

Spoken language is the preferred way of communication among human beings. People use speech to express their desires and intentions. Therefore, is of no surprise one will prefer addressing to an assistant using speech rather than any other communication way, regardless whether the assistant is another person or a robot.

It is desirable that robots understand people the way we speak. This is no trivial task. Human languages, are not constrained by strict syntactic rules. On the contrary, *natural languages* are flexible, rich in ambiguities, exceptions, and driven mostly by the intention of their speaker. In addition, the speaker often considers contextual information as known by the hearer, hence omits it. Therefore, a Natural Language Understanding (NLU) solution can't rely only in the lexicon and syntax, but reach to the semantic level, considering also acquired knowledge of the surroundings and the language itself.

All these elements are considered in our robot architecture, the VIrtual and Real roBOt sysTem (VIRBOT). In our system, the operation of a service robot is divided in four general layers: Input, Planning, Knowledge Management and Execution, where (each having several subsystems). Each layer combines traditional, reactive, and probabilistic techniques to solve the several tasks required from a service robot such as safe and robust autonomous navigation in dynamic environments, obstacle avoidance, object detection, recognition, and manipulation; people detection, recognition, and tracking; and, of course human-robot interaction via natural language. Therefore, techniques and approaches addressing NLU are core to our architecture.

Almost seventy years after the Turing test was first proposed, it is clear that computers are still far from processing language as humans do. Even though IBM Watson was taken for a human teenager, such feat involved finding an adequate reply for the given input. The same approach is not viable for a service robot. More than mapping answers, taking action and modifying the environment is required. Therefore, meaning must also be represented.

One of the main challenges in NLU is meaning representation. This work presents a semantic reasoning module used in a service robot to interpret spoken language and then execute the given task. Once a task is defined, a framework establishes the robot semantics as a series of instructions that allow the robot to perform relevant operations.

This paper is organized as follows. In Section II we discuss the related work. Then our semantic module is detailed in Section III, where the implemented natural language processing techniques are described, and in Section IV, where we show how to link the natural language representations with robot actions. Experimental results are presented in Section **??**, and Section VII concludes this work.

*This work was supported by PAPIIT-DGAPA UNAM under Grant IG-100818

II. Related Work

Natural Language is the language that we use to communicate with other people, that also can be used as an interface between humans and machines. Using natural language an user should be able to asks questions and give commands to computers in a natural way. The computer needs to understand the questions and commands to answer or perform them. But, what is understanding? The operational definition of understanding is when a system performs the actions that the user asks. In this definition is assumed that the actions chosen by the computer were the right ones.

Sometimes the system does not respond to every input, it is possible that it is modifying its internal structures and until some conditions happen it performs something that is noticeable externally. "To understand something is to transform it from one representation into another, where the second representation has been chosen to correspond to a set of available actions that could be performed and where the mapping has been designed so that for each event, an appropriate action will be performed", Rich [**?**] explains.

> *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.*

A. Syntactic Analysis

The second part of a natural language understanding system consists of syntactic analysis. Sequences of words are transformed into structures that show how the words relate each to other [**?**]. Some of the phrases may be rejected if they form a phrase that it is not allowed in the grammar. In order to begin to extract meaning from the sentence, it must be broken down into organizational patterns that a machine can understand [**?**]. This process is called parsing.

Parsing techniques are based on the theory of formal languages, these are mathematical abstractions that can be used in modeling the syntax of natural languages. "A formal language is defined in terms of an alphabet and a grammar that determines the ways in which symbols of the alphabet may be combined into sentences", explains Tanimoto [**?**]

> *A particular grammar is specified by describing the following components:*
>
> 1) *An alphabet of terminal symbols. These appear in the sentences to be parsed.*

2) *An alphabet of nonterminal symbols, that are used during the process of generating intermediate strings, but they do not appear in the input sentences.*
3) *A start symbol that belongs to one of the nonterminal symbols.*
4) *A finite set of productions rules, each of which consists of a left-hand side string and a right-hand side string.*

To test if sentence was produced by the grammar the system starts with the start symbol and applies the production rules until the input sentence is produced.

B. Semantic Analysis

One of the main problems using natural language understanding is the representation of meaning. Natural language understanding is used in order that the computer interpret the language and then perform something [**?**]. Once the application is defined we have a framework for defining the semantics. "The computer's semantics are therefore computer instructions that allow it to carry out relevant operations", Steven C. Suddarth explains in his MS. thesis [**?**].

In some systems the semantic analysis is preceded by a syntactic analysis. In those systems the grammar constrains the sentences that are allowed.

A transition network (ATN) can be used for performing the semantic analysis while doing the syntactic analysis [**?**]. An augmented transition network is a state machine with nodes that represent words and arcs connecting the words with legal paths [**?**]. See fig X. The grammar of the system is represented by this state machine, allowing only meaningful sentences. Each arc is labeled by a condition that needs to be satisfied in order to traverse it, also to each arch there is an action associated to it. These actions can be implemented by procedures that modify the system's data base, and they make partial interpretation of the meaning contained in the sentence. The conditions can be a word, sentence, or another ATN than need to be satisfied. The ATN may be call recursively in one of this conditions.

The input to the ATN is a stream of written words. Each word will lead the state machine to different nodes in the ATN, producing different actions during this process.

One way to represent the meaning contained in a sentence is through relationships an objects. During this process the main event describing the sentence and participants are found, determining the roles they play in the event, an under which conditions the events took place. Sometimes finding the verb sets the possible role and conditions in which the actions may occur. Then it is possible to associated to some verbs frames that need to be filled, these may represent the participants and they relationships. Conceptual Dependency (CD) use this type of technique to represent the meaning contained in a sentence, this is explained in **??**.

C. Semantic Networks and Conceptual Dependencies

Semantic Networks are a declarative and graphic mean to "represent knowledge and support automated systems for reasoning about the knowledge". Introduced by John F. Sowa in 1987, semantic networks are graph structures in which nodes represent concepts and the directed links (drawn as arrows between the nodes) depict the relations between concepts.

Roger Schank adopted this approach, creating the Conceptual Dependencies theory in 1972. Hence, Conceptual Dependencies are a special case of Semantic Networks. Conceptual Dependencies (CD) departs from the assumption that there is a "representation of the conceptual base that underlies all natural languages"[1]. Different from other semantic networks, CD emphasizes the role of concepts and uses different types of arrows for different relations. Also, in CD verbs are located at the core of a graph and translated to one of a set of primitives. The rest of the graph is constructed surrounding the primitive and specifying the role of the concept regarding the action. CD are studied further in Section III-B.

CD have been used to significantly increase the accuracy of a speech recognition systems in the field of robotics. CD have enriched those systems by providing ambiguity resolution and increased robustness when using contextual data [1], [2], [3], [4].

However, one of the major setbacks of CD is that some sentences generate extremely complicated graphs. This makes necessary incorporate complex task planning to map verbs to sets of primitives. In addition, is often required to define new primitives that must be hand-coded.

D. Trending approaches in robotics

Most approaches found in robotics-related literature address the problem of spoken human-robot interaction as a mapping problem [5], [6]. In them, speech is commonly pre-processed with an Automated Speech Recognition engine (ASR) whose output is mapped to a function[6]. Methods to perform such mapping range from rule-based expert systems (nowadays being abandoned [7]) to Recurrent Neural Networks (RNN) [8]. Keyword Spotting and pattern matching along with Markovian approaches are the most popular ones, with deep-learning-based methods such as Deep Neural Networks (DNN) gaining momentum [8]. Those approaches are well known and have been broadly studied within the natural language processing and computational linguistics communities, so its know they have little to offer when it comes to NLU since information above the semantic level is discarded. Due to this limitation, such approaches won't be addressed any further.

Here, lexicon and syntax altogether allow to reach information-richer levels, targeting semantics, pragmatics and beyond. Once more, RNN and deep-learning (DL) approaches have a prominent role in research [8] but its use in production systems is scarce. Is very common that stochastic methods benefit from the use of rules which are reportedly obfuscated using euphemisms to emphasize the machine-learning aspect [7]. Such approaches are of little interest when it comes to semantic reasoning since all semantics is concealed within the intricate mechanisms of the stochastic model. Henceforth, they won't be studied here.

Other research benefits from the use of semantic parsers for human-robot interaction. Semantic parsers map natural language sentences into formal representations of their meaning. Therefore, the output of the natural language has to be compatible with the knowledge representation used to address the problem such as λ-calculus or Java expressions.

Thomason et al. uses the University of Washington Semantic Parsing Framework to map natural language sentences to λ-calculus representation [9]. Their approach allowed them to implement an agent fro delivering items to an specific room [9]. It can also learn new lexicon, including misspelled words [9].

Similar in its mathematical approach, [10] addresses the problem of arranging blocks within the working space using natural language sentences [10]. An important constrain is that the robot decision process has to be as fast as possible [10]. To achieve that goal, the authors perform probabilistic inference over semantic graphs built based on the robot's knowledge of its working space and abstract concepts such as cardinality, ordinality, and spacial relationships [10].

Closer to our approach, Bastianelli et all [11] use Semantic Frames to map verbs to physical actions. The free-speech transcription provided by Google speech API is analyzed with Standford Core NLP and semantically labeled using the Babel platform [11]. This last step performs frame prediction, which selects the adequate action; boundary detection, that recognizes the arguments required by the frame within the sentence; and the argument classification that labels those arguments [11]. Such frames are quite similar to our Conceptual Dependencies (CD) in the sense that verbs are core, and then mapped to actions (primitives of action in CD) but, in our understanding, stopping at the first level of categorization, while CD goes deeper when assembling the conceptual graph.

Another difference strives

III. HUMAN-ROBOT INTERFACE

The Human-Robot Interface subsystem in the VIRBOT architecture has three modules: Natural Language Understanding (NLU), Speech Generation and Robot Facial Expressions. The NLU module finds a symbolic representation of spoken commands given to the robot; it consists of a speech recognition system coupled with Conceptual Dependency techniques [12].

One of the goals of our research is to find an appropriated representation of the spoken commands that can be used by the actions planner. For the speech recognition system, it is used the Microsoft Speech SDK engine [13]; one of the advantages of this speech recognition system is that it accepts continuous speech without training, it is freely available and provides the source code so it can be modified as needed. It allows the use of grammars, that are specified using XML notation, which constrains the sentences that can be uttered, reducing the number of recognition errors. Almost every speech recognition system currently developed has the problem of word insertion, i.e. words incorrectly added by the speech recognition system; these words may cause a robot to fail to perform the requested command. Then, the system should allow the robot to recognize the required commands even if these errors exists.

A. Natural Language Understanding

One way to represent a spoken command is by describing the relationships between objects in the input sentence; during this process the main event described in the sentence and participants are found. In this work, the participant is any actor and recipient of the actions, and the roles that the participants play in the event are deterministic, as are the conditions under which the event takes 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.

The VIRBOT's NLU system allows a robot to translate voice commands into an unambiguous representation that helps an inference engine to do action and movement planning. Th generation of metadata of words and constituents has been considered to generate an unambiguous representation, where extracting valuable information from the sentences requires syntactic, semantic analysis and queries of the state of the world. The statements' syntactic structure indicates the way in which the words relate to one another; in particular, our robot must interpret commands within known contexts. In the syntactic analysis process, each recognized word should be labeled within a grammatical category, namely, nouns, verbs, adjectives and adverbs. This type of grammatical categories are known as *open categories* since the 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 in natural language, a word can fit in different categories, depending on previous or subsequent words. To solve the ambiguity, an algorithm based on rules of the type is used (e.g. a preposition precedes a determinant and an adjective precedes a noun). In addition, there are rules of the form (e.g. a preposition of place precedes a noun of the category place). These rules are used to determine what is a correct labeling considering all possible combinations of labels.

1) Constituents of a sentence: When the words are labeled with their corresponding grammatical category, it is possible to find syntactic structures in the sentence that behave as a single unit or phrase, called *constituent*, as follows:

1) *Nominal phrase (Noun Phrase, NP):* includes proper names such as "John", pronouns like "she" or phrases like "the intelligent robot".
2) *Prepositional phrase (Prepositional Phrase, PP):* as "from the kitchen".
3) *Verbal Phrase (Verb Phrase, VP):* like "arrived on time".

Frequently, the order in which the constituents are presented can vary without altering the meaning of the sentence. For example, the prepositional phrase: "Before I get home", has different words positions in the following examples without affecting the meaning of the sentence:

"Before I get home, I'd like you to turn on the air conditioner". "I'd like you to turn on the air conditioner before I get home".

2) Separation of sentences: The speech commands that are given to the robot in most cases are composed of several sentences and they are separated by a connector and or a comma, like in the following commands:

1) *Get into the office* ***and*** *look for the soup.*
2) *Get into the kitchen,* ***look*** *for a person,* ***and*** *answer a question.*
3) *Navigate to the bedroom,* ***find*** *a person,* ***and*** *follow her to the kitchen.*

For example, the voice command "Find a person in the kitchen and answer a question" is broken down into sentences and constituents as shown in table I.

TABLE I: Constituents of a given voice command

Sentence	Words
Sentence 1	
	1 "Verb" Find
	2 "Noun phrase" a person
	3 "Noun phrase" in the kitchen
Connector	
	4 "Sentence 1" and "Prayer 2"
Sentence 2	
	5 "Verb phrase" answer a question

Then the constituents are simplified by replacing them with isolated keywords and the WORDS list is obtained with the words and the SYNTAX list with its grammatical category as follows:

1. The command is replaced by known isolated keywords, as shown in table II.

TABLE II: Isolated keywords detection

Words	Keywords
1	find
2	person
3	kitchen
4	and
5	answer
6	a question

2. After labeling the words, a list with its corresponding grammatical category is obtained, as it is illustrated in table III.

TABLE III: Grammatical category extraction

Words	Grammatical category
1 find	verb
2 person	noun
3 kitchen	noun
4 and	seq
5 answer	verb
6 a question	noun

Once the words in the sentence are grouped and labeled, the next step in is to look for interpretation patterns that describe semantic roles compatible with the structures found.

3) Role association: A semantic role refers to a noun phrase that fulfills a specific purpose with respect to the action or state that describes the main verb of the statement. The complete description of the event can be modeled as a function with parameters that correspond to semantic roles in the event that describes the verb, such as actor, object, instrument, destination place, start time, etc.

The interpretation of a variety of statements that describe a single event can be raised using fragments of the sentence that are

related to some parameter that describes the event. That is why we propose interpretation patterns that enumerate both the fragments of the statement associated with events and semantic roles and the functions that generate expressions of meaning, commands and verbal interactions. The language accepted by the interpreter depends, at its highest level of abstraction, on the set of patterns that define the assignment of meanings and commands to the action planner.

4) Interpretation patterns: The translation of natural language into a formal language useful for a robot is expressed through interpretation patterns. The interpretation patterns consist of output expressions that use the semantic roles of the verb and the description of those roles. The role's description is composed of syntactic information by defining the constituents of the instance it can come from and the semantic information by stating the categories of compatible objects. In addition, the description includes a list of keywords that must be contained in the constituent.

In the interpretation process, the system generates two outputs: conceptual dependence (expressing the non-linguistic meaning of the statement) and a verbal confirmation (a paraphrase in natural language that is repeated to the user to confirm that it has been interpreted as the intended speak).

The specification of the meaning of sentences that use a verb requires multiple patterns to cover all cases of interest in a task. The algorithm takes the set of stored interpretation patterns and looks for them in the semantic roles compatible with the structures found in the input sentence. Each interpretation attempt is scored with a value between 0 and 1, depending on the number of semantic roles and the number of words in the sentence that are not used. The algorithm returns the pattern associated to with the highest role score.

B. Conceptual Dependency

The theory of conceptual dependence is a representation of the meaning of an idea developed by Roger Schank and presented for the first time in his doctoral thesis "A Conceptual Dependency Representation for a Compute-Oriented Semantics" [12]. 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. 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.

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.)
`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.)

Each action primitive represents several verbs which have similar meaning. For instance give, buy, and take have the same representation, i.e., the transference of an object from one entity to another. Each primitive is represented by a set of rules and a data structure containing the following categories, in which the sentence components are classified:

An Actor: The entity that performs the ACT.
An ACT: Performed by the actor, done to an object.
An Object: The entity the action is performed on.
A Direction: The location that an ACT is directed towards.
A State: The state that an object is in, and is represented using a knowledge base representation as facts in an expert system.

For instance the phrase: "Robot, please give this book to Mary", when the verb give is found in the sentence an ATRANS structure is issued.

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

The empty slots (NIL) need to be filled finding the missing elements in the sentence. The actor is the robot, the object is the book, etc, and it is represented by the following CD:

```
(ATRANS
   (ACTOR Robot)        (OBJECT book)
   (FROM book's owner)  (TO Mary) )
```

It is important to notice that the user could say more words in the sentence, like "**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 a one representation that is more suitable to be used by an actions planner.

The user's spoken input is converted into a CD representation using a two step process. The CDs are formed first by finding the main verb in the spoken sentence and choosing the CD primitive associated with that verb. Once the CD primitive has been chosen the other components of the sentence are used to fill the CD structure.

The primitive acts mentioned above serve to organize the process of inference of the meaning of the sentence. However, there is disagreement about the choice of the set of primitive acts because it is difficult to demonstrate that all ideas and their paraphrases can be expressed using a finite set of primitive acts. By means of an implementation of Shank's Conceptual Dependency theory, the meaning of a natural language sentence is extracted and represented with a set of primitives. This technique finds the structure and meaning of a sentence in a single step. 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, and are subtle of being processed with ease in an inference machine.

Conceptual dependencies can also be used with multi-modal input [14]. For instance, if the user said "Put the newspaper over there", while pointing from the floor to the table top, separate CDs will be generated for the speech and 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))`

Empty slots can be filled by examining CDs generated by other modalities at the same time, and combining then to form a single representation of the desired command:

```
(PTRANS
(ACTOR Robot) (OBJECT Newspaper)
(FROM Floor)  (TO Table top)    )
```

The final CD encode the users commands to the robot. These structures facilitates the inference process, but this inference problem is not solved, what it was done is to reduce the number of verbs into a small number of items, from which inferences can be done. CDs can be use for representing simple actions. It is also well suited for representing commands or simple questions, but it is not very useful for representing complex sentences. In our system the conceptual dependencies technique was implemented in an expert system, as it is explained in III-D.

C. The use of Pronouns

The use of pronouns play an important role in our system, because using them allows us to have a broad range of operations without increasing the number of sentences to be recognized.

Terry Winograd and et (ref x) develop a language-understanding system that was able to to carry out actions and answer questions about a simple scene containing a table, hand, several blocks, a box, pyramids and an arm Robot that was able to perform operation with the objects. The user introduced the commands through a keyboard and he saw the operation in a computer monitor. This system was able to recognize the pronouns embedded in the sentences, and use them to perform its operation. Winograd used a detail representation of the world model, its represents the current states of the objects and it has knowledge procedures for changing the state and making deductions of it.

Winograd explains that one of the problems of using pronouns is that sometimes is difficult to deduce to what objects they are making reference. For instances if some one says "I drooped the vase in the folding table, and I broke it", without further information we do not if he broke the table or the vase. But if in the next sentence he ask for a linen and a broom to clean water and debris we would be certain that he broke the vase. This type of deduction are difficult to implement, and Winograd explains that his system can only make primitive deductions of simple sentences that involve pronouns.

Our system also uses very primitive type of deductions. After a sentence is recognized and executed we save the object, place or actor referenced by it. Then if in the next sentence appears a pronoun it is substituted by the previous reference. For instances when the user says "Where is the newspaper", in the next sentence the object newspaper is used as reference. This use of pronouns allows us to reduce the number of sentences to be recognized while increasing the number of operation to be perform, because instead of having a sentence for an action with each object, we just have it with pronouns. From the previous example the user may say "Robot, give it to the Father", instead of "Robot, give the newspaper to the Father".

D. Expert Systems

Much of the human problem solving or cognition can be expressed by IF THEN type production rules. Each rule corresponds to a modular collection of knowledge call chunk. The chunks are organized in loose arrangement with links to related chunk of knowledge, reasoning could be done using rules. Each rule is formed by a left side that needs to be satisfied (Facts,) and by a right side that produce the appropriate response (Actions.)

```
IF Facts THEN Actions
```

When an action is issued by a rule it may become a fact for other rules, creating links to other rules. A system may use thousands of rules to solve a problem, thus it is necessary a special mechanism that will select which rules will be fired according to the presented facts. That mechanism is an Expert System "Engine". The Inference Engine makes inferences by deciding which rules are satisfied by facts, prioritize the satisfied rules, and executes the rule with the highest priority. In our system we use the open expert system CLIPS, that was designed by NASA with the specific purposes of high portability, low cost, and easy integration. It is designed to allow artificial intelligence research, development, and delivery on conventional computers. CLIPS provides a cohesive tool for handling a wide variety of knowledge with support for three different programming paradigms: rule-based, object-oriented, and procedural. In the VIRBOT system an expert system maintains a knowledge data base that represents the state of the world. The data of the humans interacting with the robot, of the objects and the locations is represented using facts that contain several slots with information related with them. Each one has an identification; a location that specifies in which room has been located by the robot; an action that specifies what the actor is doing; and the slots x and y specifies the actor's position in the room, etc. The robot is considered an actor too.

```
(robot
(id robot) (location living-room)
(x 1.20) (y 1.35) (angle 0.35) )
(Mother
(id Mother) (location studio)
(action studying) (x 250) (y 300) )
```

The entities on which the actors can perform actions have a similar representation, for example:

```
(newspaper
   (location outside-door)
   (x 320) (y 550) (id newspaper) )
(milk
   (location fridge)
   (x 135) (y 500) (id milk) )
```

IV. Action Planner

After receiving the CD representation from the Human/Robot interface the Perception subsystem perceives a new situation that needs to be validated by the Situation Validation system. This validates the situation by the information provided by the Knowledge Management Layer. The Planner subsystem takes as an input the output of the Activation of Goals system and tries to take care of the situation presented.

A. Single Task Planner

The Robot is able to perform operations like grasping an object, moving itself from on place to another, finding humans, etc. Then the objective of action planning is to find a sequence of physical operations to achieve the desired goal. These operations can be represented by a state-space graph.

B. Planning using space-state search and hierarchical task networks

The task planning adopts concepts from space-state search planning and hierarchical task networks, like the ones used in classical STRIPS-like planners, so depending on the current situation and the currently active tasks, each task can decompose in one plan or another.

The plan specification is done through facts that represent a hierarchical structure of tasks, and each task can have several planning rules. The planning rules would be useful for considering different situations, present in the environment, so the robot can act accordingly. The mechanism to generate a new plan it starts occasionally with a spoken command, the representation of this spoken command should be made in other that planning can be achieved to solve the requirement in it.

For example when the user says "**Robot, find a person in the kitchen and answer a question**", after the speech recognition system recognizes these words the robots asks for verbal confirmation: "**Do you want me to find a person in the kitchen and answer a question?**"

If the answer is positive, the following CDs are generated:

```
(ATTEND
       (ACTOR Robot)   (OBJECT Person)
       (FROM Kitchen)  (TO Vision-System))
(SPEAK
       (ACTOR Robot)   (OBJECT Answer-Question) )
```

All the information required for the actions planner to perform its operation are contained in the CDs and the knowledge data base. The `ATTEND` primitive requires that 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) )
```

That means that the Robot is moving itself. Then with these three CD primitives a set of ordered high-level tasks, in which the parameters corresponding to each type of action are specified. Within the action planner, each task is decoded for the purpose of creating a set of more specific primitive tasks.

Using the PTRANS primitive the motion planner finds the best global path between the Robot's place and the Kitchen, then the robot navigates to the kitchen by reaching each node in the path, if there are unknown obstacles not considered by the planner it avoids them using reactive behaviors.

Once the robot is in the Kitchen, with the CD primitive ATTEND, it will active the behavior to find a human, once it find him it approaches the person. The CD SPEAK primitive generates speech asking for a question from the person, and the answer a question behavior is activated.

Each task has a unique structure and a particular way of dividing into primitive tasks, some of the most used tasks are:

1. Find_Person_In_Room: The robot must search for a person in a specific room.
2. Wait_For_User_Instruction: The robot waits for an instruction from a person such as: answer a question, introduce himself, tell time, etc.
3. Go_Object_Location: The robot must navigate to the place where an object is located.
4. Get_Object: The robot activates object recognition to find a specified object that later it will take with its actuators.
5. Put_Object_In_Location: The robot releases the object that it carries in its actuators in the specified place.
6. Handover_Object: The robot delivers an object in the hands of a person.

V. Architecture Implementation

The conceptual base of our implementation is VIRBOT, an architecture which is briefly introduced in the first subsection before proceeding to explain the implementation in a real robot in the following section.

A. The VIrtual and Real roBOt sysTem (VIRBOT)

In general, a service robot should be able to perform the following tasks:

- Autonomous navigation and unknown and dynamic obstacle avoidance.
- Place and object recognition without artificial marks.
- Person detection, recognition and tracking.
- Speech recognition and NLU .
- Autonomous mapping

To deal with these challenges, we propose a robot architecture that combines traditional, reactive and probabilistic techniques. In our robot architecture, known as VIrtual and Real roBOt sysTem (VIRBOT), the operation of a service robot is divided in four general layers: Input, Planning, Knowledge Management and Execution, where each of them has several subsystems, as shown in Figure 1. This system has similar features as those presented in the INTERRAP agent architecture [15]. The VIRBOT has a combination of basic artificial intelligence (AI) techniques, specifically the ones used in NLU , with devices and technology developed in the recent years. By combining symbolic AI with digital signal processing techniques, a good performance in a service robot has been obtained. NLU is used in a service robot to interpret spoken language and then execute a task, where one of the main problems using NLU is the meaning representation. Once the application is defined, we have a framework that establishes the robot semantics, defined as a series of instructions that allow a robot to perform relevant operations.

In this section, we will describe most relevant VIRBOT modules categorized by layer.

B. Inputs Layer

This layer encloses the robot's internal and external sensors, real or simulated, in a series of modules, as follows.

Human-Robot Interface: This super-module is responsible of recognizing and processing voice and gesture commands. Speech is processed here in the NLU module as explained in Section III.

Symbolic Representation and Interpretation: Here, digital signal processing techniques are applied to the data provided by the internal and external sensors to obtain a symbolic representation of the environment.

Perception (Hypothesis Generation): This module generates a set of beliefs about the possible states of the environment. Beliefs are based on the symbolic representation of the sensorial information coming from internal and external sensors, as well as the processed user input from the Human-Robot Interface module. Such beliefs are validated later on to either trigger actions or update the robot's world model.

C. Planning Layer

This layer is responsible of generating plans at a high level of abstraction and performing global reasoning.

Beliefs generated by the perception module are validated in this module with information of the Knowledge Management layer. Once validated or recognized, a belief is concealed as knowledge and either stored or used to trigger the Action Planner (explained further in Section IV-A), which will generate a plan of action or sequence of physical operations to achieve the desired goals. However, if something unexpected happens while executing a plan, the Goal Activator will be notified, interrupting the Action Planner and triggering the generation of a new plan.

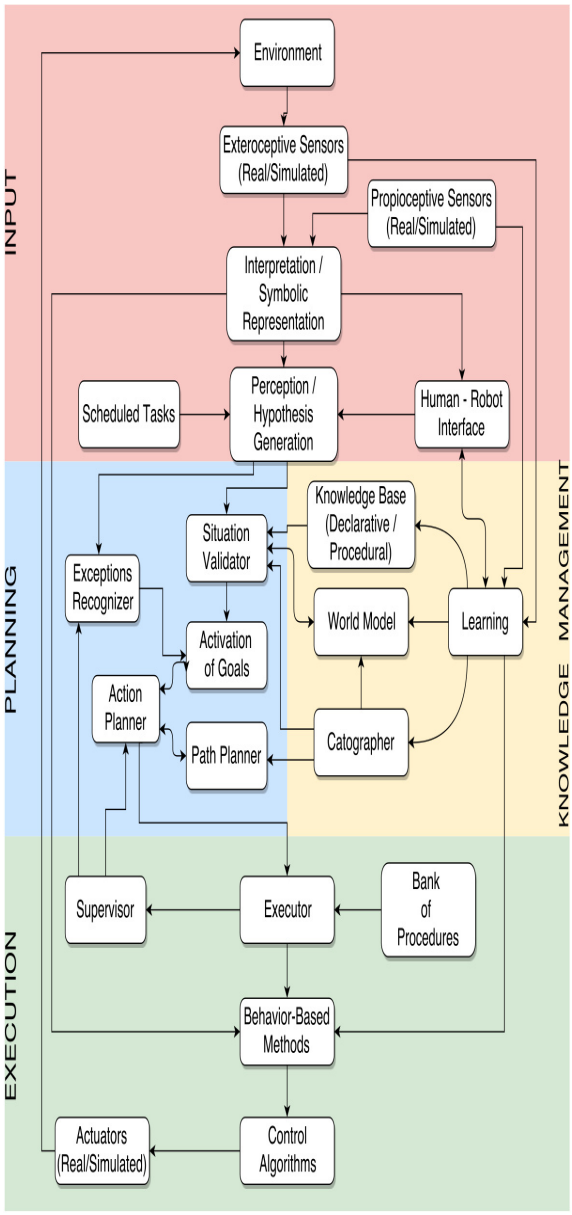

Fig. 1: The VIRBOT consists of several modules that control the operation of a mobile robot with four main layers, namely Inputs, Planning, Knowledge Management and Execution layers.

D. Knowledge Management Layer

This layer involves all modules that store, conceal, and provide access to the robot's knowledge. Such knowledge, which may not be symbolic, ranges from raw maps, to semantic knowledge of the language.

For high-level reasoning, a rule-based system is used. The facts and rules are written in CLIPS, a language developed by the NASA, and represent the robot's knowledge while encoding knowledge of an expert, as explained in detail in Section III-D.

E. Execution Layer

This layer is responsible of executing generated plans and taking local decisions.

At its core, the Bank of Procedures encapsulates a set of hardwired functions the Action Planner combines to assembly more complex plans. These functions implement state machines to partially solve very specific problems, including robot motion and object manipulation. Such functions rely on low level Behavior Methods, a set of reactive algorithms to solve local, not foreseen situations like obstacle avoidance.

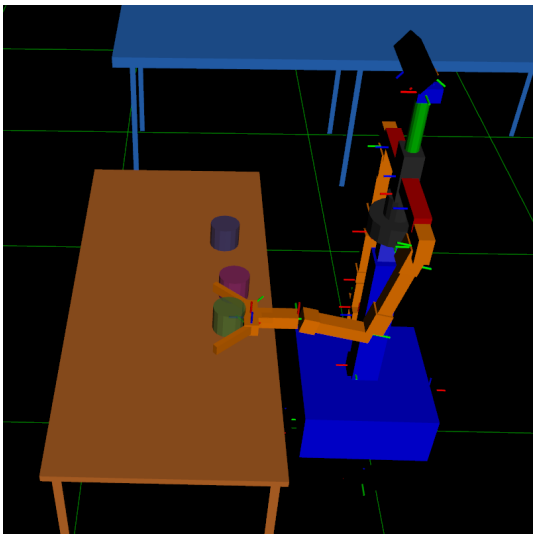

Fig. 2: Symbolic sensorial information allows the VIRBOT system to operate in both, real and virtual environments.

F. Implementation in Robot Justina

The implementation of VIRBOT is made through several highly-specialized interdependent modules running in a distributed heterogeneous environment. Such modules are coordinated by a *Simple Task Planner* (VIRBOT's Bank of Stored Procedures) and the VIRBOt's *Action Planner*. Based on the robot's internal state and input from the NLU , the action planner selects the most adequate procedure to follow, delegating its execution to the Simple Task planner.

Although we decided to migrate all our code to ROS, our best results in speech recognition were achieved using the Microsoft speech recognition engine. Recognized speech is sent to the NLU module (coded in CLIPS, see Section III). Hence, two middlewares are used, ROS under Linux, and Blabkboard (BB hereinafter) under Windows. BB is a centralized shared-variable storage and message-passing hub following the publisher-subscriber pattern that runs in Windows and Linux, therefore it is used to bridge between both operating systems. Currently there are APIs to build BB modules in C/C++, C#, python, and CLIPS (via PyCLIPS).

We have been testing the proposed system in our service robot Justina (see figure 3). In addition, a simulator has been also implemented in ROS and the 3D visualization tool Rviz. A simulation example is shown in Figure 2.

VI. Experiments and Results

The VIRBOT's semantic reasoning system was tested using the RoboCup@Home General Purpose Service Robot (GPSR) test [**?**]. In this test there are three categories that have been been classified according to their difficulty.

For category I the service robots need to solve easy tasks with a low difficulty degree, involving indoor navigation, grasping known objects, answering questions (from the predefined set of questions), etc. Some examples are:

- Bring me the apple juice from the counter.
- Put the crackers on the kitchen table.
- Tell me how many beverages are in the shelf.

Fig. 3: Service robot Justina. The external structure and appearance has been designed by a group of artists from the Faculty of Fine Arts, National University of Mexico.

- Speech recognition and natural language understanding.
- Tell me the name of the person at the door.

For category II the service robots need to solve tasks with a moderate difficulty degree. According to the rules in this category involves following a human, indoor navigation in crowded environments, manipulation and recognition of alike objects, find a calling person (waving or shouting), etc. Some examples are:

- Tell me how many beverages in the shelf are red.
- Put the banana on the kitchen table.
- Count the waiving people in the livingroom.
- Follow Ana at the entrance.
- Tell me the name of the woman in the kitchen.

For category III the robots need to comprehend challenging tasks involving dealing with incomplete information, environmental reasoning, feature detection, natural language processing, outdoors navigation, pouring, opening doors, etc. Some examples are:

- Pour some cereals in the bowl.
- Go to the bathroom (Bathroom's door is closed).
- Bring me the milk from the microwave (The milk is inside the microwave)

There is also an extend GPSR in which the robots need to do three simple actions, which the robot has to show it has recognized. The robot may repeat the understood command and ask for confirmation.

Command examples

- Go to the kitchen counter, take the coke, and bring it to me.
- Bring the chips to Mary at the sofa, tell the time and follow her.
- Find a person in the living room, guide them to the kitchen and follow them.

General Purpose Service Robot commands are generated randomly using the official [EE]GPSR Command Generator and grammars publicly available at https://github.com/kyordhel/GPSRCmdGen. The VIRBOT's semantic reasoning system was tested using this generator with the following commands' type:

- **Follow person commands:** Eg. Navigate to the living table, meet Charlie, and follow him.
- **Guide person commands:** Eg. Navigate to the cabinet, meet Edward, and accompany him to the dining table.
- **How many people commands:** Eg. Tell me how many people in the bedroom are male.
- **Person instructions commands:** Eg. Find a man in the bedroom and answer a question.
- **How many objects commands:** Eg. Tell me how many drinks there are on the cupboard.
- **Feature object commands:** Eg. Tell me what is the smallest object on the cupboard.
- **Bring me object commands:** Eg. Navigate to the fridge, find the apple, and bring it to me.
- **Place object commands:** Eg. Take an apple to the fridge.

Table 4 shows the results obtained for the semantic reasoning system for 50 sentences generated by the official [EE]GPSR Command Generator and given to our robot Justina.

Justina's Skill Ranks						
	Speech Recognition	Interpretation	Navigation	Person Recognition	Object Recognition	Object Manipulation
Follow Person Commands	0.6	0.8	0.5	0.5	...	...
Guide Person Commands	0.5	0.76	0.77	0.53	...	...
Gender Person Commands	1	0.66	0.6	0.6	...	...
Pose Person Commands	1	0.66	0.6	0.6	...	...
Name Person Commands	1	0.66	0.6	0.6	...	...
How many people Commands	0.4	0.8	0.66	0.66	...	...
Person instruction Commands	0.6	0.75	1	0.66	...	...
How many Objects Commands	1	0.85	1	...	1	...
Feature Object Commands	1	0.72	1	...	1	...
Bring me Object Commands	0.4	0.79	1	...	0.5	0.5
Place Object Commands	0.6	0.9	1	...	0.5	0.5
Handover Object Commands	0.6	0.8	1	0.5	0.5	0.5

Fig. 4: Results.

In this test were evaluated our service robot skills in terms of Speech Recognition, Interpretation, Navigation, Person Recognition, Object Recognition and Object Manipulation. The performance values are between 0 and 1, with 1 means a perfect perfomance.

Of the 50 sentences Justina completed 45 complete plans (with certain errors like: several attempts when trying to find objects, not recognizing the voice command or finding the wrong person), and 5 plans completely failed, there were serious faults like: Justina lost his position, created a plan that did not correspond to what was requested, hit an obstacle.

Our system has been successfully tested in robotics competitions [16], as the RoboCup and RoCKIn [17], in the category @Home. In the following link is presented the qualification video for 2017 RoboCup @Home competition of our team Pumas, in which we present how our robot is able to interact with users and the environment updating its knowledge data base in real time, using vision and speech understanding: `https://youtu.be/gPiU8NAiz7k` In RoboCup 2017 competition, at Nagoya Japan, in the @Home league our robot was awarded as the best in Speech and Natural Language Understanding.

VII. Conclusions

The most significant contribution in this research is that we have combined successfully speech recognition system with AI techniques together to enhance semantic reasoning. At this time we have the following characteristics for the system: speaker independent, small vocabulary, loose grammar and context dependent.

References

[1] R. C. Schank, "Conceptual information processing author: Roger c. schank, publisher: Elsevier science publishing co inc., us pages: 384," 1975.

[2] J. Savage-Carmona, "A hybrid system with symbolic ai and statistical methods for speech recognition." Ph.D. dissertation, University of Washington, 1995.

[3] J. Savage, A. LLarena, G. Carrera, S. Cuellar, D. Esparza, Y. Minami, and U. Peñuelas, "Virbot: a system for the operation of mobile robots," in *RoboCup 2007: Robot Soccer World Cup XI*. Springer, 2008, pp. 512–519.

[4] J. Savage, A. Weitzenfeld, F. Ayala, and S. Cuellar, "The use of scripts based on conceptual dependency primitives for the operation of service mobile robots," in *RoboCup 2008: Robot Soccer World Cup XII*. Springer, 2009, pp. 284–295.

[5] Y. Bisk, D. Yuret, and D. Marcu, "Natural language communication with robots," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 751–761.

[6] L. Iocchi, D. Holz, J. Ruiz-del Solar, K. Sugiura, and T. Van Der Zant, "Robocup@ home: Analysis and results of evolving competitions for domestic and service robots," *Artificial Intelligence*, vol. 229, pp. 258–281, 2015.

[7] L. Chiticariu, Y. Li, and F. R. Reiss, "Rule-based information extraction is dead! long live rule-based information extraction systems!" in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 827–832.

[8] W. De Mulder, S. Bethard, and M.-F. Moens, "A survey on the application of recurrent neural networks to statistical language modeling," *Computer Speech & Language*, vol. 30, no. 1, pp. 61–98, 2015.

[9] J. Thomason, S. Zhang, R. J. Mooney, and P. Stone, "Learning to interpret natural language commands through human-robot dialog." in *IJCAI*, 2015, pp. 1923–1929.

[10] R. Paul, J. Arkin, N. Roy, and T. M. Howard, "Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators." in *Robotics: Science and Systems*, 2016.

[11] E. Bastianelli, G. Castellucci, D. Croce, R. Basili, and D. Nardi, "Effective and robust natural language understanding for human-robot interaction." in *ECAI*, 2014, pp. 57–62.

[12] R. C. Schank, "A conceptual dependency representation for a computer-oriented semantics," Ph.D. dissertation, University of Texas at Austin, 1969.

[13] "Microsoft speech sdk," 2006. [Online]. Available: http://www.microsoft.com/speech/

[14] S. L. Lytinen, "Conceptual dependency and its descendants," *Computers & Mathematics with Applications*, vol. 23, no. 2-5, pp. 51–73, 1992.

[15] J. P. Müller, "The agent architecture interrap," *The Design of Intelligent Agents: A Layered Approach*, pp. 45–123, 1996.

[16] J. Savage, A. LLarena, G. Carrera, S. Cuellar, D. Esparza, Y. Minami, and U. Peñuelas, "Virbot: a system for the operation of mobile robots," in *Robot Soccer World Cup*. Springer, 2007, pp. 512–519.

[17] J. Savage, M. Negrete, J. Cruz, J. Marquez, R. Martell, J. Cruz, E. Vazquez, M. Pano, J. Cruz, E. Silva, H. Estrada, H. Arce, M. Matamoros, A. Garzón, and O. Fuentes. (2017) Pumas@home 2017 team description paper. [Online]. Available: https://biorobotics.fi-p.unam.mx/downloads/finish/3-papers/699-pumas-home-tdp-2017