# Pumas@Home 2018 Team Description Paper *

Jesus Savage, Reynaldo Martell, Hugo Estrada, Marco Negrete, Julio Cruz, Jesus Cruz, Jose Cruz, Edgar Vazquez, Jaime Marquez, Edgar Silva, Manuel Pano, Luis Alvarez, and Mauricio Matamoros

Bio-Robotics Laboratory, School of Engineering
National Autonomous University of Mexico
http://biorobotics.fi-p.unam.mx

**Abstract.** This paper describes the service robot Justina of team Pumas that has participated in the @Home category of the RoboCup and RoCKIn, both of them international competitions; as well as our latest applied research. These competitions had influenced our architecture in the development of better systems for our service robots by developing algorithms to natural language understanding, facial detection through multiple images using RGB cameras and the sound source localization (SSL). In our robotics architecture, the VIrtual and Real roBOt sysTem (VIRBOT), the operation of service robots is divided into several subsystems, each of them has a specific functionality that contributes to the final operation of the robot. By combining symbolic AI with digital signal processing techniques a good performance of a service robot is obtained.

## 1 Introduction

Service robots are hardware and software systems that assist humans to perform daily tasks in complex environments, to achieve this: they have to be able to understand spoken or gesture commands from humans; to be able to avoid static and dynamic obstacles while navigating in known and unknown environments; to be able to recognize and to manipulate objects and performing several other tasks that a person might request.

Our team has been participated in the category @Home continuously since the start of this competition at the RoboCup in Bremen in 2006. Our team obtained the third place in Atlanta in 2007, and has reached the finals in 2014 and 2015, last year, in the RoboCup 2017, the team obtained the fourth place and got the award for the best in Speech Recognition and Natural Language Understanding.

The paper is organized as follows: section 2 enumerates the hardware and software components of our robot Justina; section 3 presents overview of the latest research developments in our laboratory; and finally, in section 4, the conclusions and future work are given.

* Acknowledgment: This work was supported by PAPIIT-DGAPA UNAM under Grant IG100915

# 2 Justina's Robotics Architecture

## 2.1 Hardware Configuration

Our service robot Justina, see figure 1, has the following hardware configuration:

ACTUATORS:

- **Mobile base:** Omnidirectional through differential pair configuration and omnidirectional wheels.
- **Manipulators:** 2 x 7-DOF anthropomorphic arms with 10 Dynamixel servomotors each.
- **Head:** 2-DOF (Pan and tilt) built with Dynamixel servomotors.
- **Torso:** 1-DOF (Elevation) through a worm screw and a configuration of gears.
- **Speakers:** Two speakers to generate synthetic speech.

SENSORS:

- **RGB-D Camera:** Microsoft's Kinect sensor
- **RGB Camera:** Logitech Pro C920 Full HD.
- **Microphone:** Rode NTG2 directional microphone.
- **Array of Microphones:** An array of four microphones to detect sound sources.
- **Laser:** Hokuyo rangefinder URG-04LX-UG0.

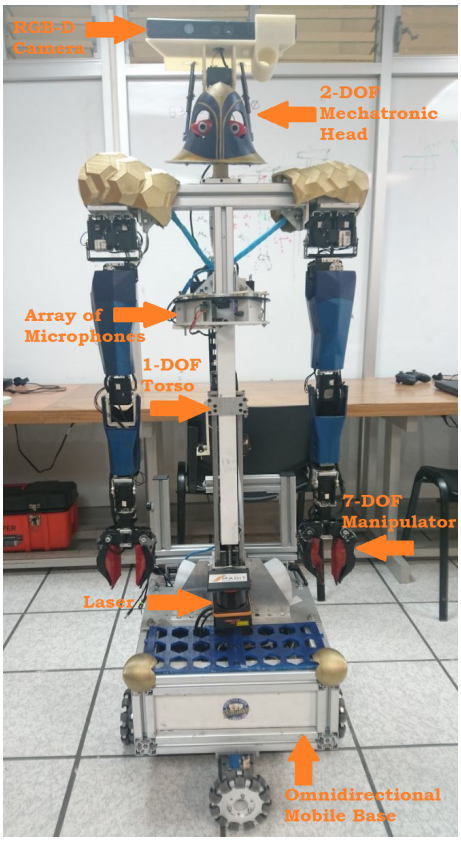


Fig. 1: Robot Justina

## 2.2 Software Configuration

Our software configuration is based on the VIRBOT architecture [1], which provides a platform for the design and development of software for general purpose service robots, see figure 2. The VIRBOT architecture is implemented in our robots through several modules that perform well defined tasks [2], with a high level of interaction between them. The principal framework used for interaction is ROS, where a module is represented by one or several ROS's nodes. Also, for modules using the Microsoft operating system, we use our own middleware called Blackboard to link them with ROS nodes running on Linux. In the following sections are explained each of the layers of the VIRBOT system.

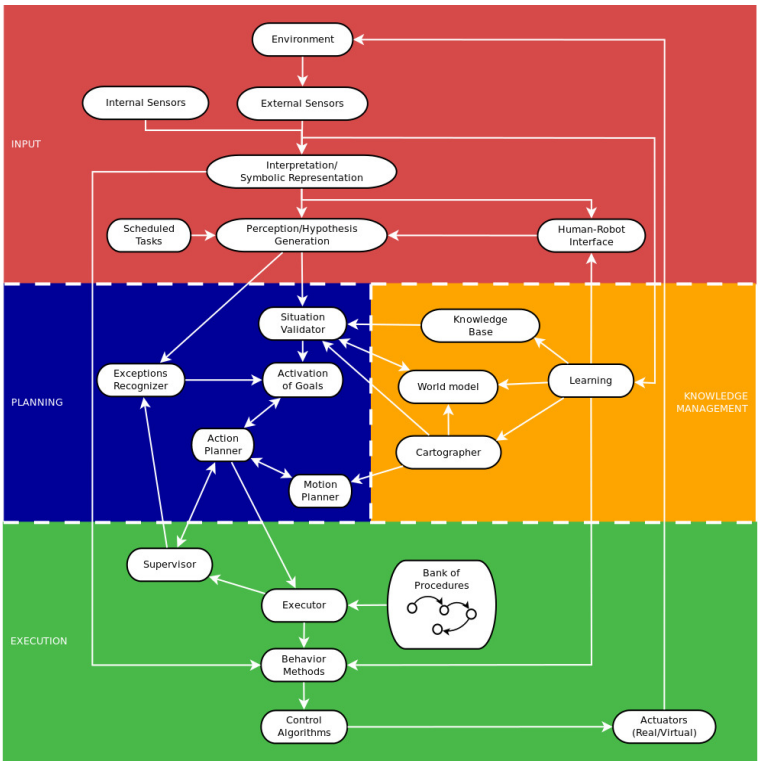


Fig. 2: Block diagram of the ViRBot architecture.

### 2.3 Inputs Layer

This layer process the data from the robot's internal and external sensors, they provide information of the internal state of the robot, along with the external world where the robot interacts. In some of Justina's designs it has lasers, sonars, infrared, microphones and stereo and RGB-D cameras. Digital signal processing techniques are applied to the data provided by the internal and external sensors to obtain a symbolic representation of the data, furthermore, to recognize and to process voice and visual data. Pattern recognition techniques are used to create models of the objects and the people that interact with the robot. With the symbolic representation this module generates a series of beliefs, that represent the state of the environment where the robot interacts.

### 2.4 Planning Layer

The beliefs generated by the perception module are validated by this layer, it uses the Knowledge Management layer to validate them, thus a situation recognition is created. Given a situation recognized, a set of goals are activated in order to

solve it. Action planning finds a sequence of physical operations to achieve the activated goals.

### 2.5 Knowledge Management Layer

This layer has different types of maps for the representation of the environment, they are created using SLAM techniques. Also in this layer there is a localization system, that uses the Kalman filter, to estimate the robot's position and orientation. A rule based system, CLIPS, developed by NASA, is used to represent the robot's knowledge, in which each rule contains the encoded knowledge of an expert.

### 2.6 Execution Layer

This layer executes the actions and movements plans and it checks that they are executed accordingly. A set of hardwired procedures, represented by state machines, are used to partially solve specific problems, person recognition, object manipulation, etc. The action planner uses these bank of procedures and it joins some of them to generate a plan.

## 3 Current research

In this section is presented the current research developed in our laboratory to improve the performance of our service robots.

### 3.1 Natural language understanding

Natural language understanding is used in order to the service robot interprets the language and then perform an especific task. One of the main problems using natural language understanding is the representation of meaning. We have a framework for defining the semantics. The robot's semantics are therefore instructions that allow it to carry out relevant operations.

Conceptual Dependency (CD) is a theory, developed by Schank [3], for representing the meaning contained in sentences. This technique finds the structure and the meaning of a sentence in just one step. It is useful to represent sentences using this technique when there is not a strict grammar associated with the sentences, and also when the objective is to make inferences from them. The CD representation of a sentence is built using conceptual primitives, these represent thoughts and the relationships between thoughts. Using conceptual dependency facilitates the use of inference rules, because many inferences are already contained in the representation itself. There are several primitives to represent actions, for example two of the more commonly used are the following:

ATRANS: Transfer of an abstract relationship (e.g., give.)

PTRANS: Transfer of the physical location of an object (e.g., go.)

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 one. Each primitive is represented by a set of rules and data structures. Basically each primitive contains the following components:

An Actor: He is the one that perform the ACT.

An ACT: Performed by the actor, done to an object.

An Object: The action is performed on it.

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))

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. The CD technique were implemented in an expert system.

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. This expert system provides a cohesive tool for handling a wide variety of knowledge with support for three different programming paradigms: rule-based, object-oriented, and procedural. 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. 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 are represented by a state-space graph.

In the previous example, when the user says **"Robot, please give this book to Mary"**:

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

All the information required for the actions planner to perform its operation is contained in the CD and knowledge data base. Our system has been successfully tested in robotics competitions [4], as the RoboCup and RockIn [5], in the category @Home. In RoboCup@Home 2017 our robot was awarded as the best in Speech and Natural Language Understanding.

### 3.2 Facial detection through multiple images using RGB cameras

The facial detection is one of the most primordial tasks that a service robot must be able to perform, proof of them is that in the RoboCup@Home there is a test in which the robot must correctly state the number of people who make up a crowd, these people can be stand, sit and even lie down, the crowd can be composed of very tall people or children and the crowd size can range from 5 to 10 people, so that correctly stating the crowd turns out to be a difficult task to be solved with a single image, that is why we decided to figure out the problem by generating a registered image composed by several images at different angles, in figure 3 are shown some samples to perform the image registration. Stitching is the process of merging multiple images with fields of view that overlap to produce a registered image. The image stitching method can be divided into three main phases: image registration, calibration and mixing.

Fig. 3: Frames Captured by the Robot at Different Angles.

The first thing we do in our implementation is to convert the color images to grayscale images, then we use SURF [6] as a feature detector. The next step is to perform pairing of the detectors using FLANN [7], we also execute a computation to obtain the maximum and minimum distances of the points of interest. We later found the homography matrix using RANSAC [8]. Finally we combined the images using the homography matrix. All this process is done iteratively, first we start with two images, to the union of those two images we can join another and so on. Once we have the registered image, we work with it to compute the facial detection. The results are shown in figure 4.

Fig. 4: Facial Detection on a Registered Image.

### 3.3 Sound Source Localization

The Sound Source Localization (SSL) is performed by estimating the direction of arrival (DOA) of the voice signals of people who are in unknown positions around the robot. Considering the DOA estimate is required over an angular range of $[0, 2\pi)$, a triangular arrangement of microphones was used, as shown in figure 5.

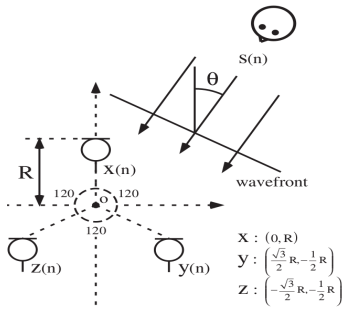


Fig. 5: Equilateral triangular arrangement [9].

For the DOA estimation it is crucial to obtain the phase information present in the signals of each of the microphones [10]. In order to obtain accurate information about the signal offset in each of the microphones, an MCU was used for the acquisition and processing of the signals, the MCU used is the TMS320F28377S, which has two 12-bit analog-to-digital (ADC) converters, each of these ADCs has seven channels with independent conversion starts (SOC), with these ADCs can be obtained up to 1.1 MSPS [11]. The DOA calculation is performed using the cross-power spectral density (CPSD) between pairs of microphones, the CPSD between signals $x_1$ and $x_2$ is obtained as:

$$G_{12}(\omega) = X_1^*(\omega)X_2(\omega) = P(\omega)e^{-j\omega\tau} \tag{1}$$

being $X_1(\omega)$ and $X_2(\omega)$ the Fourier transforms (FT) of $x_1(n)$ y $x_2(n)$, respectively.

In the case of the triangular arrangement, there will be three pairs of microphones, so there are three CPSDs. Using the three previous CPSDs, the integrated cross-spectrum can be obtained as:

$$G_{\phi,\theta}^{(\omega)}(\theta,\phi) = G_{x2y}^{(\omega)}(\phi)G_{xy}^{(\omega)}(\theta) + G_{yz}^{(\omega)}(\theta) + G_{z2y}^{(\omega)}(\phi)G_{zx}^{(\omega)}(\theta) \tag{2}$$

The above equation has its maximum when $\phi = \theta$, In this way, the angle of arrival is located at the angle $\phi$ that maximizes the equation (2).

## 4 Conclusions and future work

It is clear, that during the 10 years in which our team Pumas has been participated in the RoboCup and 2 years in the Rockin [5] in the category @Home, the performance and research developed, in the service robot area, in our laboratory has been improved considerably. Our service robot architecture, the VIRBOT, has been evolving according to the requirement that these robotics competitions asked each year. In these years, the full system has been improved, both in hardware and software, having reliable performance and showing promising results. Particularly, this year, we have a new omnidirectional mobile base for navigation and a new torso. In terms of software, we have change the way of conceiving the tests of the competition: from static state machines to inferred action planning generated by a rule based system. As for future work, the task of following humans will be improved by combining vision algorithms using the Kinect sensor and our current system tha use a laser. Also, it will be explored topics as the memory and enviromental reasoning of the robot, in order to explain things that hapenned in the past. Moreover we are working in a mask with lights that serves as an interface to express different emotions.

## References


1. *ViRbot: A System for the Operation of Mobile Robots*, Savage, Jesus and et al, RoboCup 2007: Robot Soccer World Cup XI, pp 512-519, Springer Berlin Heidelberg, 2007.
2. *The Design of Intelligent Agents: A Layered Approach*, Muller, Jorg P,Springer-Verlag New York, Inc.1997.
3. *Conceptual dependency and its descendants*, Steven L. Lytinen, Computers & Mathematics with Applications, 1992.
4. *The Role of Robotics Competitions for the Development of Service Robots*, Jesus Savage, Marco Negrete, Mauricio Matamoros, Jesus Cruz, IJCAI'16, Workshop on Autonomous Mobile Service Robots, New York, USA, 2016.
5. *RoboCup@Home* http://www.robocupathome.org
   *Rockin* http://rockinrobotchallenge.eu/home.php
6. *SURF: Speeded Up Robust Features*, Herbert B., Andreas E., Tinne T., Luc Van G., Computer Vision and Image Understanding (CVIU), 110(3): 346-359, 2008.
7. *Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration*, Marius Muja and David G. Lowe, International Conference on Computer Vision Theory and Applications (VISAPP'09), 2009.
8. *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*, Fischler, Martin A. and Bolles, Robert C., Commun. ACM, 24(6): 381-395, 1981.
9. *DOA estimation of speech signal using equilateral-triangular microphone array*, Yusuke Hioka and Nozomu Hamada, 8th European Conference on Speech Communication and Technology (EUROSPEECH 2003), 2003.
10. *A DOA estimation method for an arbitrary triangular microphone arrangement*, Amin Karbasi and Akihiki Sugiyama, 14th European Signal processing Conference (EUSIPCO 2006), 2006.
11. *TMS320F2837xS Data Sheet* http://www.ti.com/lit/ds/sprs881b/sprs881b.pdf

## 5 Team Information

**Name of Team:**
Pumas

**Contact Information:**
Jesus Savage
Bio-Robotics Laboratory
School of Engineering
National Autonomous University of Mexico
robotssavage@gmail.com

**Web Site:**
http://biorobotics.fi-p.unam.mx

**Team Members:**
Jesus Savage, Reynaldo Martell, Hugo Estrada, Marco Negrete, Julio Cruz, Jesus Cruz, Jose Cruz, Edgar Vazquez, Jaime Marquez, Edgar Silva, Manuel Pano, Luis Alvarez, Mauricio Matamoros

**Description of Hardware:**
Justina's Robotics Architecture (cf. section 2)

**Description of Software:**
Most of our software and configurations are open-source and can found at: https://github.com/RobotJustina/JUSTINA

| | |
|---|---|
| Operating System | Ubuntu 16.04 LTS; Windows 7 VM |
| Middleware | ROS Kinetic; Blackboard |
| SLAM | ROS Gmapping |
| Navigation | Navigation using Kinect + Ocupancy grid + A* |
| Object Recognition | Histogram Disparity |
| Face Detection | Haar Cascades |
| People Detection | OpenPoses |
| Face Recognition | EigenFaces |
| Speech Synthesis | Loquendo |
| Speech Recognition | Microsoft Speech Recognition |
| Inference Engine | CLIPS |