Pumas@Home 2020 Team Description Paper *

Jesus Savage, Julio Cruz, Reynaldo Martell, Hugo Estrada, Marco Negrete, Diego Cordero, Manuel Pano, Julio Martinez, and Luis Gonzalez

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 (OPL) 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 and Human-Robot interaction using multiple service robots. 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 fourth place and got the award for the best in Speech Recognition and Natural Language Understanding in Nagoya in 2017, in the last years, in the RoboCup 2018 and 2019, the team obtained the second place.

The paper is organized as follows: the section 2 enumerates the software description of our robot Justina; the section 3 presents an overview of the latest research developments in our laboratory; the section 4 is about our contributions for RoboCup @Home; in section 5, the conclusions and future work are given; finally, in Appendix A and B, you can find the hardware description of our robot Justina and the Information about Team Pumas.

* Acknowledgment: This work was supported by PAPIIT-DGAPA UNAM under Grant IG100818

2 Justina's Robotics Architecture

2.1 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 1. 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 subsections are explained each of the layers of the VIRBOT system.

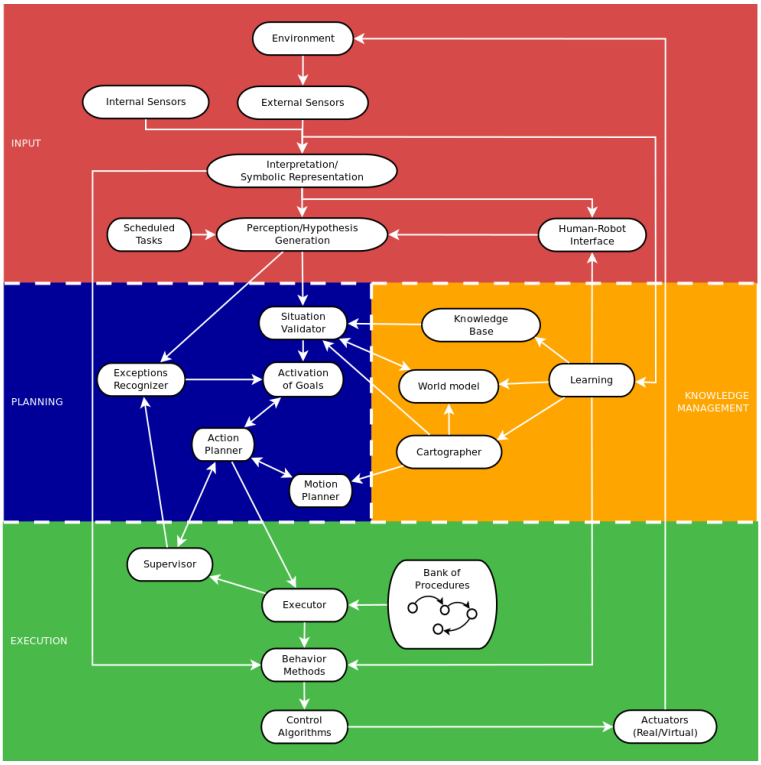

Fig. 1: Block diagram of the ViRBot architecture.

2.2 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. Using the symbolic representation this module generates a set of beliefs, that represent the state of the environment where the robot interacts.

2.3 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.

2.4 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.

2.5 Execution Layer

This layer executes the actions and movements plans and it makes sure that they are executed appropriately. A set of hardwired procedures, represented by state machines, are used to partially solve specific problems, such as, person recognition, object manipulation, etc. The action planner uses these bank of procedures and link up some of them that one may 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 described in [4] and successfully tested in robotics competitions [5], as the RoboCup and RockIn [6], in the category @Home. In RoboCup@Home 2017 our robot was awarded as the best in Speech and Natural Language Understanding.

3.2 Human-Robot interaction using multiple service robots

In our laboratory we have the vision that in the future we will not only have a service robot in our homes, but a variety of them performing specific tasks. The first problem to solve is find the way to communicate with all these robots regardless of where the user and the robot are. The solution we propose is to use smart home devices to send the users requests to the service robots, see figure 2. Specifically we are using Alexa, the voice service that Amazon offers.

Alexa integration in Justina's environment. Alexa is an Amazon service for automatic language recognition and natural language understanding, allowing you to interact with different devices provided by this same company. However, it is also possible to extend its functionalities to third parties, so it has been possible to integrate it into the JUSTINA robot. Alexa is made up of two main elements:

- Amazon Echo - Is an array of microphones and speakers that Alexa uses as end device to be listening the user voice.
- Alexa Voice Service (AVS) - Is a service for voice recognition [7].

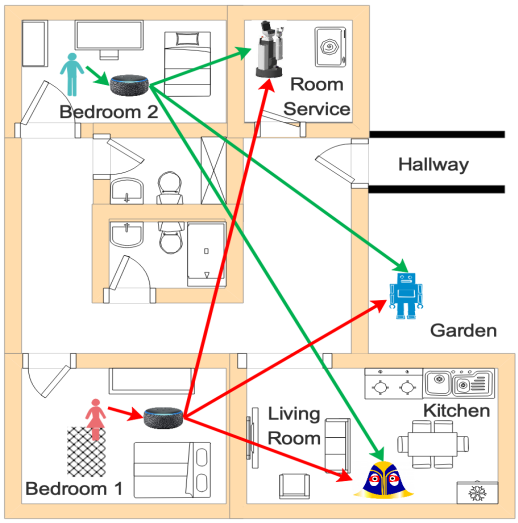

Fig. 2: Human interaction with multiple robots trough intelligent devices.

AVS request / response handling. The communication between Alexa and Justina is ilustratedted in Figure 3 and consists of a local area network, where an echo device is waiting for a user's voice command (Figure 3 - 1). Then the voice command is sent to the AVS for recognition (Figure 3 - 2), for a better voice recognition we use AVS's *Skills* system [8], where the *Intents* or voice commands that AVS must recognize are defined. If AVS recognize an *Intent*, a request is created and sent back to the local area network (Figure 3 - 3). To attend the request generated by AVS a web service node was created with the detail of the request (Each request is represented in JSON format) (Figure 3 - 4). Finally a CD, such as PTRANS or ATRANS, is generated and sent to Justina for execute the user's voice command (Figure 3 - 5).

To use other robots through this system, it is possible to create a web service node for each of them.

Applicability of the approach in the real world. As previously stated, our vision is to have different robots inside the home performing independent or joint tasks, where robots can communicate with each other and with the user, also robots can interact with different smart devices to perform another tasks such as turn on lights, play music, turn on appliances, open or close doors, etc. The main purpose is to divide the work and offer to the user more options to complete the daily tasks at home. We showed some of this work in the fnal of the RoboCup@Home Open Platform League 2019, where Justina and Takeshi worked together, in a party atmosphere, Takeshi was in charge of receiving guests at the entrance while Justina served drinks for the guests.

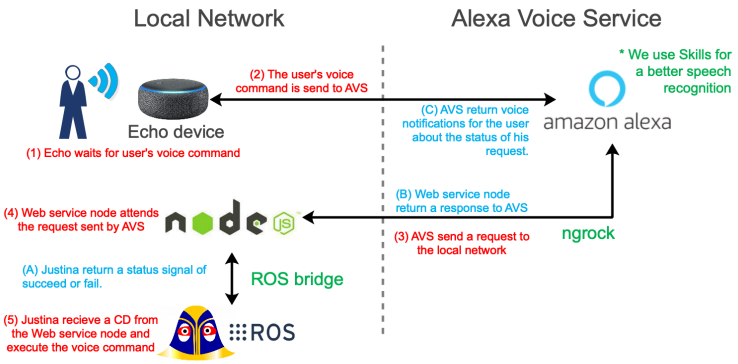

Fig. 3: Block digram of Justina and alexa comunication.

3.3 Object recognition with semantic segmentation

We are currently working to integrate, as a new feature, instance segmentation (figure 4) in the Justina Robot vision, for object detection precisely, the main goal is to get a better object detection system.

The method we want to use is a state of the art convolutional neural network called Mask R-CNN, which is based on Faster R-CNN. The architecture of Mask R-CNN is divided in four main steps, the first one consist in extract features of a map feature by applying different kernels to extract important information of the images, the first layers extract simple features such as borders and lines, the deepest layers extracts finer information from the images, the other three steps works in parallel, prediction, bounding-box regression and mask generation, all of them has the same input, a RoI (region of interest), the feature map extracted from the first step is divided into small map feature where can be an object of interest this is done with an intermediate step called RoI align.

The classification layer has different regions of interest (ROI) as input, each ROI is flatten before to push it into the fully connected layer, the output layer is a softmax, the object class is defined as the maximum output in the softmax layer. The softmax layer has as outputs as the number of classes plus one because of the background.

The bounding-box regression is only for get a more accurate bounding box alignment this is because of each ROI could have small translations for the original object.

The mask generation step consists in generating a mask for each class in the data set and then the correct mask is chosen by the classification made by the classification step.

The implementation of this method focuses on the objects that we want to find, for that, we use, transfer knowledge and then re-training parameters of the neural network with the information of objects we want to find in the environment of the robot.

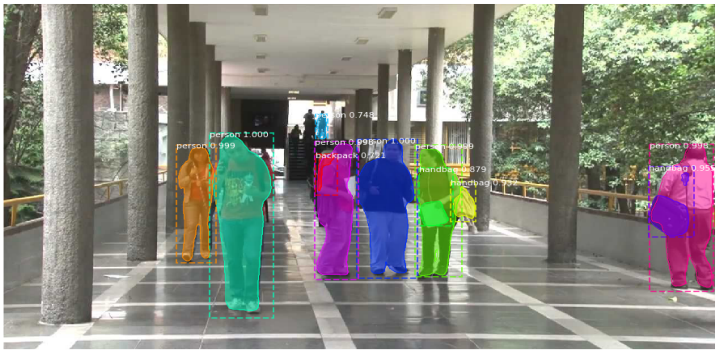

Fig. 4: Semantic segmentation of people scene.

Our team is working in testing different backbones for the feature extraction in order to get a good relation between time of inference, accuracy and hardware requirements.

4 Contributions for RoboCup@Home

During our participation in the RoboCup@Home we have promoted, mainly in our country, the interest in participating and collaborating in this competition through the following contributions:

1. We host the RoboCup@Home in Mexico City, Mexico, 2012.

2. Extracurricular course for the programming and development of service robots, Engeneering Faculty, UNAM, 2019.

3. Conferences at the Robotics Winter School to promote the study of robotics related topics, Xalapa Veracruz, Mexico, 2018.

4. Outreach programs (domestic workshops, international academic exchanges, et.)

5 Conclusions and future work

We have described the main features of Justina's architecture as well as the recent research for the Robocup@Home Open Platform League 2020. Based on the performance of Justina in the competences of recent years and the state of development achieved, we improved the hardware and software systems providing our robot with better reliability and new skills to work in different environments. Particularly, this year, we add an Embedded Develotment kit [11] for perform

GPU's solutions. In terms of software, we have change the way of conceiving the tests of the competition: passing from static state machines to inferred action planning generated by a rule based system. We also began to develop an architecture in which different robots can communicate with each other and collaborate to help users. Team Pumas will continue working to improve Justina's capabilities through the experience that RoboCup@Home competencies have offered us since 2006.

References

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 Use of Expert Systems for Semantic Reasoning in Service Robots*, Jesus Savage, Julio Cruz, Reynaldo Martell, Hugo Leon, Marco Negrete, and Jesus Cruz, 2nd Workshop on Semantic Policy and Action Representations for Autonomous Robots (SPAR), IROS 2017
5. *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.
6. *RoboCup@Home* http://www.robocupathome.org
   Rockin http://rockinrobotchallenge.eu/home.php
7. *AVS* https://developer.amazon.com/en-US/alexa/alexa-voice-service
8. *Skills* https://developer.amazon.com/docs/ask-overviews/build-skills-with-the-alexa-skills-kit.html
9. *Robocup@home 2018: Rules and regulations*, M. Matamoros, C. Rascon, J. Hart, D. Holz, and L. van Beek, 2018
10. *Intelligent flat-and-textureless object manipulation in Service Robots*, A. Ortega, H. Estrada, E. Vázquez, R. Martell, J. Hernández, J. Cruz, E. Silva, J. Savage, and L. Contreras, IROS 2018 Workshop "Towards Robots that Exhibit Manipulation Intelligence", 2018.
11. *Jetson* https://developer.nvidia.com/embedded/buy/jetson-tx2-devkit

A Appendix

A.1 Hardware Configuration

Our service robot Justina, see figure 5, has the following components:

HARDWARE:

- **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.
- **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.
- **Embedded System:** NVIDIA Jetson TX2 to image processing.

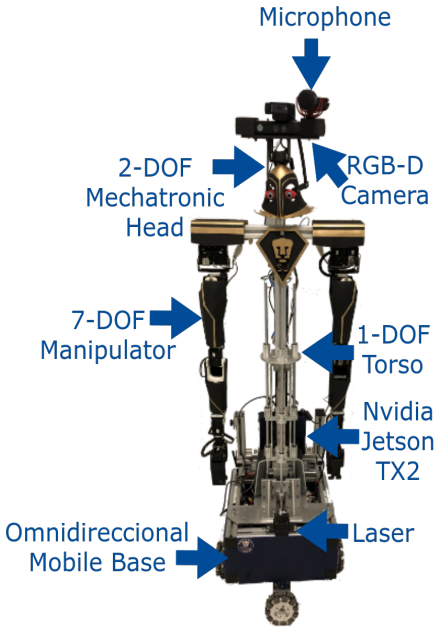

Fig. 5: Robot Justina

B 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, Julio Cruz, Marco Negrete, Diego Cordero, Manuel Pano, Julio Martinez, Luis Gonzalez, Jesus Cruz, Jose Cruz, Jaime Marquez

Description of Hardware:

Justina's Robotics Architecture (cf. Appendix A)

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 + YOLO
Face Detection	Haar Cascades
People Detection	YOLO
Gesture Recognition	OpenPoses
Face Recognition	Facenet
Speech Synthesis	Loquendo
Speech Recognition	Microsoft Speech Recognition
Inference Engine	CLIPS