The Role of Robotics Competitions for the Development of Service Robots *

Jesus Savage, Marco Negrete, Mauricio Matamoros, Jesus Cruz, Ramon Lagunas,
Jaime Marquez, Juan Marentes and Francisco Villaseñor
BioRobotics Laboratory, School of Engineering and School of Art and Design,
Universidad Nacional Autónoma de México (UNAM)

Abstract

This paper presents the influence that competitions of robots, like the RoboCup and Rockin in the category @Home, have had in the development of better robotic architectures for service robots. Our service robot, named Justina, has been participating in these competitions successfully in the last years and our robotic architecture has been evolving according to their requirements. In our robotic architecture, the VIrtual and Real roBOt sysTem (VIRBOT), the operation of this service robot is divided into several subsystems, each of them has a specific functionality that contributes to the final operation of the robot.

The VIRBOT has a combination of basic artificial intelligence (AI) techniques, specifically the ones used in natural language understanding, with devices and technology developed in the last years. By combining symbolic AI with digital signal processing techniques a good performance of a service robot is obtained. In this paper is presented results obtained in two specific test in the @Home competitions, navigation and object recognition. These results were obtained using inexpensive RGB-D cameras, which provide information of the environment through a cloud of points, that are used to create a representation of objects in it, as well as, for the creation of road maps.

1 Introduction

There are several robotic architectures for service robots, and one way to test their performance and to be able to do a fair comparison is through their participation in robotics competitions. One of the most important competition of robots in the world is the RoboCup, in the 2006 version the @Home league was introduced, in which service robots compete in a house environment. The goal of this league is to promote the development of real-world applications and human-machine interaction with autonomous robots, or as they put it: "the aim is to foster the development of useful robotic applications that can assist humans in everyday life" [1]. Our team Pumas participated in this competition with the robot TPR8 from the first edition and it has been participating since then with different versions of our initial robot, Figure 1 shows our last robot, named Justina which participated in the 2015 RoboCup.

Figure 1: Service robot Justina participating in the @Home League in the 2015 RoboCup.

Competitions ground the scientific research and engineers work. Designing a robot requires much more than merging and coordinating top trending technologies in machine learning, computer vision, control, navigation, real time mapping, artificial intelligence, and human-computer interaction. A Service Robot interacts with an ever-changing environment in which the "optimal" conditions of a research laboratory are almost ever met, and here is where competitions play an essential role. While laboratory experiments may be unintentionally tweaked by the creators of the robot, arenas in tournament set the perfect playground for real testing in an uncontrolled and safe environment since they are not designed by experts and allow people of the audience to interact with

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

the robots.

Having people interacting with robots provides also valuable information. Of those interaction, 80 % are conducted in the same manner, and often matching the developer's conception. In contrast, the 20 % remaining is of unexpected nature and robots are never able to deal with it since they were not programmed too. Such interactions are very important because we, as humans, know how to solve them, but somehow have render unable to transmit that knowledge to the machine.

However, competitions may be very discouraging. Even though the interaction provides invaluable feedback for the developer team, the performance of robots are far under the public expectations. The audience is predisposed to the performance of science fiction movies and the tweaked demonstrations shown in TV, feeling really disappointed when the robot fails to accomplish a tasks as trivial as grasping an apple, a task that might be trivial for an industrial robot in a controlled environment, but that turns really complex in real world.

The former is mostly due to the traditional A.I. approach. The experts are so used to work with robots that unconsciously bypass or forget the way in which people interacts with the environment. This means that the robotcist must not think on how the robot can solve a problem, but in all possible ways a human being may solve it. Therefore, the problem of A.I., at least when dealing with robots, is not to find the optimal solution for a problem, but to explore new solutions. In this sense, instead of being constrains to a set of algorithms for solving particular problems, robots must explore their own solutions and learn from their own mistakes.

The RoboCup@Home competition tests different abilities that a service robot should have, like: navigation; detection, recognition and manipulation of objects; recognition and following persons; natural language understanding with spoken commands, etc. Almost every year the league's rule-book is updated making the competition more challenging.

In the firsts @Home competitions the capabilities that our robot had, were based mainly on a traditional architecture of robotics, that is, it had a symbolic representation of the environment, it used polygons to represent the obstacles in it, it had also a topological map, both built by hand, doing direct measurement in the environment.

Motion planning was done using standard search algorithms, as Dikjstra's algorithm. For the recognition of the objects it was allowed to put marks in them to facilitate the recognition. The complete system was controled by a rule based system

Each year the complexity of the competition increased forcing the teams to do an extra effort to have a service robot working in something close to a real environment outside research laboratories. With these new rules the robots needed to do the following:

- Mobile robot navigation avoiding unknown and dynamic obstacles.
- Recognition of objects and places without artificial marks.
- Remembering, recognizing and be able to follow persons.
- Speech recognition and natural language understanding.
- Creation of maps automatically

Thus, to cope with this changes we needed to change our architecture to include reactive modules and also to have a probabilistic robotics perspective. Our architecture evolved into a hybrid one that combines traditional, reactive and probabilistic robotics' methods.

Another very important issue was, that we had a very monolithic software system, and the complexity of the competition had forced us to have a software system that was structured using various modules that needed to communicate between them, thus we developed our own middle-ware, and later we started using ROS.

We changed also the robot's physical structure, we put it two arms of seven degrees of freedom to be able to pick objects in different heights; a mobile torso and a mechatronic head with pan and tilt movements and in its top with a RGB-3D sensor, the Kinect, to be able to find persons and objects in the environment.

Another competition that has had also an impact in service robots' architectures is the Rockin@Home [2], one of the features of it, it is that the organizers collect data that can be used as benchmarking data useful for comparing the robots performance. This feature forced us to gather data in a scientific way that later can be presented in robotics' conferences and journals.

The paper is organized as follows, first a description of our evolved robotic's architecture, the VIRBOT system, is presented, then how this service robot architecture is implemented, then our current research with tests and results that were obtained, and finally the conclusions.

2 VIRBOT: A System for the Operation of Service Robots

Our service robot Justina, where the VIRBOT is tested, has the following hardware configuration:

ACTUATORS:

- Differential mobile base with Arduino Mega for control and Pololu RoboClaw board as motor driver.
- Two arms with anthropomorphic design: 7 DOF implemented with 10 Dynamixel servomotors and CM-700 microcontroller board for control and path planning.
- Mechatronic Head with 2 DOF (Pan and tilt) built with Dynamixel servomotors.
- Speakers.

SENSORS:

- One Kinect sensor and one stereo RGB camera.
- Directional microphone.
- Laser rangefinder Hokuyo URG-04LX-UG0

The design of Justina's external structure and appearance has been provided by a group of artists from the school of Art and Design in our university.

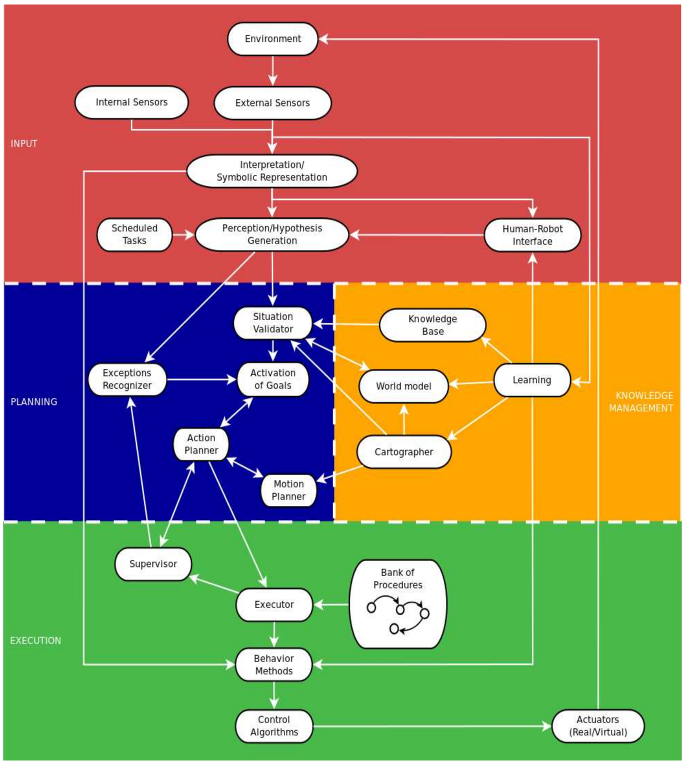

Figure 2: The VIRBOT System consists of several subsystems that control the operation of a mobile robot.

In the VIRBOT system, the operation of our service robot, Justina, is divided in four general layers: Inputs, Planning, Knowledge Management and Execution, having each of them several subsystems, see Figure 2. Each subsystem has a specific function that contributes to the final operation of the robot. This system has similar features presented in the INTERRAP agent architecture [3]

2.1 Inputs Layer

In this layer are the robot's internal and external sensors, as well as, the simulator that simulates these sensors when a VIRBOT Robot is used.

Robot Internal Conditions: This module provides information about the robot internal state through temperature, encoders, battery charge, inclinometer, accelerometer and magnetometer sensors.

Sensors: This module provides information from the external world where the robot interacts. In some of Justina's designs it has laser, sonar, and infrared sensors; a microphone; stereo and RGB-D cameras.

Human/Robot Interface: The purpose of this module is to recognize and to process voice and gesture commands, this module is explained with more detail in section 4.1

Robot's Tasks: A set of programmed tasks that the robot needs to accomplish during an established time, as the deliver of objects from one place to another, the cleaning of places, etc.

Interpretation and Symbolic Representation: Digital signal processing techniques are applied to the data provided by the internal and external sensors to obtain a symbolic representation of the environment. More details are given in section 4.4.

Perception and Hypothesis Generation: With the symbolic representation this module generates a series of beliefs, that represent the state of the environment where the robot interacts.

2.2 Planning Layer

Situation Validation: The beliefs generated by the perception module are validated by this module, it uses the Knowledge Management layer to validate them, thus a situation recognition is created.

Activation of Goals: Given a situation recognition, a set of goals are activated to solve it.

Action Planner: The objective of action planning is to find a sequence of physical operations to achieve the desired goals generated by the previous module. More details are given in section 4.2.

Motion Planner: The module receives, from the action planner, a set of locations that the robot needs to visit and it finds paths to reach them using a topological map and the A* Algorithm.

Exception Recognizer: If during the execution of a plan an exception occurs, that means, if something not considered in it happens, this module enter in action and tries to solve the problem by setting new goals and re-planning the overall plan.

2.3 Knowledge Management Layer

Cartographer: This module creates and contains different types of maps for the representation of the environment.

- *Map building:* SLAM techniques are used to create a representation of the obstacles and environment's free space.
- Raw maps. These are obtained by detecting the position of the obstacles using the robot's laser sensors and the Kinect, and they are represented using point clouds..
- Symbolic maps. These represent all known obstacles as polygons defined by their vertexes.
- Road maps. These contain roads maps used to navigate in the environment's free space. This section is explained with more detail in section 4.3
- *Self Localization:* The Kalman filter is used to estimate the robot's position and orientation.

Knowledge Representation: 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.

Learning: The following learning algorithms are used to improve the robot capabilities:

1. *Objects and persons recognition:* Pattern recognition techniques are used to create models of the objects and the persons that interact with the robot.

2. *Behaviors:* Genetic Algorithms are used to learn new behaviors.

2.4 Execution Layer

Executor: This module executes the actions and movements plans.

Supervisor: This module checks that the actions and movements plans are executed accordingly.

Behavior Methods: A set of reactive algorithms is used to solve problems not foreseen by the movement planner, like the avoidance of unknown obstacles.

Bank of Procedures: A set of hardwired procedures, represented by state machines, are used to partially solve specific problems, including movement, object manipulation, etc. The action planner uses these bank of procedures and it joins some of them to generate a plan.

Control Algorithms: Control algorithms, like PID, are used to control the operation of the virtual or real actuators.

Virtual an Real Actuators: The virtual or real actuators receive the commands and execute them by interacting with the virtual or real environment and with the user.

3 Architecture Implementation

The implementation of VIRBOT is made through several modules (executables) that perform well defined tasks and have a high interaction between each other. The information exchange between modules is made through a central module called Blackboard (BB), which supports shared variables, with publisher/subscriber pattern, and message passing. Blackboard was developed with C# and the .NET framework, and there are currently APIs to build BB modules for languages C#, C/C++ and python. It runs on Microsoft Windows and linux-based systems. Also, a BB module has been developed for an Android system and a CLIPS interpreter has been used with the help of PyCLIPS, expanding the number of platforms and languages available to use with it.

BB's commands and shared variables have a certain equivalence to ROS's services and topics respectively. Integration between modules working with ROS has been accomplished through the implementation of a Blackboard Bridge ROS module, which communicates with both systems. This bridge had help us to use the best applications written in the Microsoft and the Linux environments. But, due to the fact that ROS has been gaining momentum and there is too many application that can be used with it we decided to migrate almost all our code to this platform in the following years.

4 Current research

In this section it is presented the latest research topics in which we have been concentrated in the last years, mainly how to create good action plans, how to navigate with unknown unpredictable obstacles and how to recognize objects with almost no texture.

4.1 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.

Human/Robot Interface

The Human/Robot Interface subsystem in the VIRBOT architecture has tree modules: Natural Language Understanding, Speech Generation and Robot's Facial Expressions. The natural language understanding module finds a symbolic representation of spoken commands given to the robot. It consists of a speech recognition system coupled with Conceptual Dependency (CD) techniques [4].

For the speech recognition system it is used the Microsoft Speech SDK engine [5], one of the advantages of this speech recognition system is that it accepts continuous speech without training, also is freely available and with the C++ source code included, it means that 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 and with that feature the number of recognition errors it is reduced. Almost every speech recognition system currently developed has the problem of insertion words, words incorrectly added by the speech recognition system. These words may cause a robot to fail to perform the asked command, then it is necessary to find a mechanism that, even if these errors exists, that a robot should be able to perform the required commands. 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. One way to represent a spoken command is by describing the relationships of objects mentioned in the input sentence. During this process the main event described in the sentence and participants are found. In this work the participants are any actors and recipients of the actions. The roles the participants play in the event are determined, as are the conditions under which the event took place. The key verb in the sentence can be used to associate the structure to be filled by the event participants, objects, actions, and the relationship between them.

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.

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.

4.2 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.

Actions 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.

For example when the user says **"Robot, go to the kitchen"**, the following CD is generated:

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

It is important to notice is that the user could say more words in the sentence, like **"Please Robot, go to the kitchen now, as fast 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.

All the information required for the actions planner to perform its operation is contained in the CD. The planner just needs to find the best global path between the Robot's place and the Kitchen.

4.3 Navigation Planner

The Navigation system is composed of several subsystems performing different tasks in different levels of abstraction: a set of behaviors controlling the mobile base and another set controlling the mechatronic head, a localization subsystem, which contains the world representation and a Kalman Filter for estimating the robot's position, and a perception module, responsible for processing the raw sensor data.

Justina's perception module includes objects and faces detection and recognition, natural landmark extraction, speech recognition, detection of obstacles and their position, skeleton detection and proprioception, that includes odometry and position estimations of the head and arms. The navigation system uses the subprocesses of odometry, landmark extraction and obstacle detection. Landmarks are extracted from laser readings and point clouds generated by the Kinect sensor, using an algorithm based on the work of [6].

World representation includes a geometric representation of the obstacles in the environment and a set of nodes used for path planning. Such path planning is made using the A* algorithm, considering information contained in the world representation and the perception module. Localization is made using a Kalman Filter.

The mobile base and the mechatronic head are controlled by a set of behaviors. Head is moved by two behaviors: the first one tries to point the head towards the nearest landmarks and the second one, towards the nearest obstacle. Mobile base is controlled by three behaviors. First behavior tries to move the robot towards a given goal point. The second one, avoids obstacles using potential fields. Laser readings are used for this purpose. The third behavior is checking if there are obstacles with which the robot could crash and stops the robot in case of collision risk.

The navigation system also builds roadmaps when it is moving to improve the obstacle avoidance. Roadmaps are constructed following the next steps:

- Point cloud acquisition from the kinect sensor.
- Transformation to the robot frame coordinates.
- Separation of free and occupied space (considering the z-coordinate of the points).
- Clusterization of free and occupied space by vector quantization techniques.

Centroids of free space are taken as nodes to calculate paths and clusters of occupied space are taken as obstacles. see figure 3. A more detailed description of the roadmap construction method is given in [7].

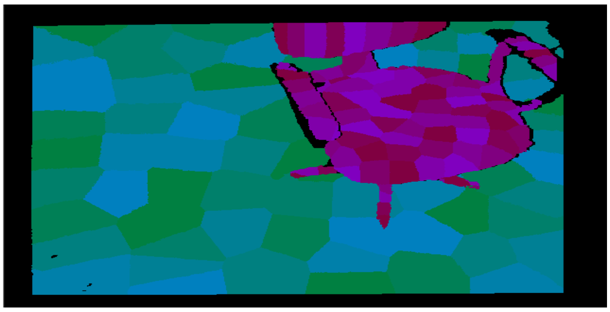

Figure 3: Free space clusters are colored in green and occupied space clusters, in purple. The black regions are those points with no depth information.

4.4 Recognition of objects for low or null texture

Contrary to the objects found in the Robocup@Home competition the objects in the Rockin have little or no texture at all, which led us to the development of the following algorithm.

Several robust techniques based on feature extraction and description exist for object recognition. However, if the objects are low textured, only a few number of features can be extracted, making the matching process unreliable. For these

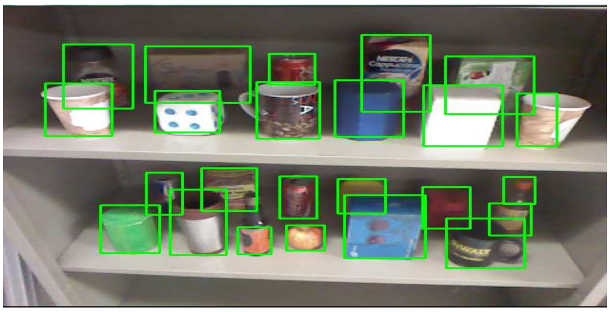

Figure 4: Example of object detection and segmentation on multiple planes

cases, we developed a method that combine three characteristics: color, size and shape, combining the color and depth information for the recognition process, after a 3D detection and segmentation in a plane for each object.

Color Information is extracted from the HSV space of the object's pixels and it is represented by the histogram of the Hue component, but only for pixels with Saturation and Value above certain threshold. For pixels below these thresholds, two more bins are added to the histogram. For campaign histograms we used the histogram Intersection [8].

The size and shape is estimated from the object's point cloud, which are obtained using an oriented bounding box (OBB) of the point cloud as follow: the base of the OBB is obtained from the oriented bounding rectangle of the projection of the points in the plane below them. The heights are obtained from the maximum distance of the points to the plane, see Figure 4. The shape is characterized using the Hu Moments of the convex hull calculated from the points projected over the plane below them.

For the recognition process we compare in three steps: size, shape and color characteristics, removing candidates below a certain threshold for each step. At the end, from the remaining candidates, we select the best one according to a color-based similarity function.

5 Experiments and Results

The clustering algorithm for the free and occupied space was tested using a standard PC computer mounted on the robot and also a PC computer with a GPU. Processing six frames of Kinect data using the GPU took in average 0.078 [s] and it was 18 times faster than CPU without a GPU. This processing time to find a roadmap allows a safe navigation since the maximum robot speed is 1.0 [m/s]. The resulting roadmaps have been tested in the service robot Justina, in the Robocup@Home tests in stage I, "Navigation", where "the robot must visit a set of waypoints while avoiding obstacles on its path, following a person outside the arena and, finally, guide that person back to the arena."

The method for the recognition of objects for low or null texture has been tested experimentally in the Rocking 2015 robotics competition, where the team Pumas obtained a second place in the "object perception test", showing fast and robust results for changes in light , scale, and rotation in a plane parallel to the plane below the object. Figure 4 shows an example of object recognition on a shelf (multiple planes).

6 Conclusions

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 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 ask each year. This improvements are not only on how to combine better AI and DSP techniques, but also how to incorporate new devices to the robot as the Kinect sensor, as well as, the use of new middle-ware paradigms like ROS. One issue that is not commented in the research papers, is how to incorporate new students in the team, not only for the continuation of the system, but also in the improvement of it. If the system is well organized into separated modules then this integration can be made effortlessly and smoothly. Without the participation of our team to these robotics competitions our service robot Justina would not be able to obtain the degree of sophistication that in the present it has.

In conclusion, robotics competitions not only allow to grasp the research and advances of other robotics groups with much more detail than any scientific paper would allow, they also provide invaluable insights of the real state of robotics development, how far are we from taking robots to an unsupervised domestic environment, and even more important, gives to the public a real perspective of what Service Robotics is, while at the same time, remind scientists what is expected from the machines they are building.

References

[1] *RoboCup@Home* 2015: Rule and Regulations, Loy van Beek, Mauricio Matamoros and Sven Wachsmuth, http://www.robocupathome.org/rules/2015_rulebook.pdf, 2015.

[2] http://rockinrobotchallenge.eu/home.php

[3] *The Design of Intelligent Agents: A Layered Approach*, Muller, Jorg P,Springer-Verlag New York, Inc.1997.

[4] *Conceptual Information Processing*, Schank, Roger C., Elsevier Science Inc, New York, NY, USA. 1975.

[5] http://www.microsoft.com/speech/

[6] *Natural corners extraction algorithm in 2d unknown indoor environment with laser sensor*, R. Yan, J. Wu, W. Wang, S. Lim, J. Lee, and C. Han, Control, Automation and Systems 12th, (ICCAS), pages 983-987., 2012.

[7] *Parallel implementation of roadmap construction for mobile robots using rgb-d cameras*, Marco Negrete, Jesús Savage, Jesús Cruz, and Jaime Márquez, OGRW2014, pages 184–187, 2014.

[8] *Color indexing*, Michael J Swain and Dana H Ballard., International journal of computer vision, 7(1):11–32, 1991.