Donaxi @HOME

Hector Vargas, Rodrigo Ortega, Luis Felipe Velazquez, Alvaro Bravo, Sebastian Gamboa, Rene Cruz, Harold Vasquez, Aldo Fonseca, Dulce Teutli, Irais Miranda, Gustavo Vargas, Eduardo Arce.
Popular Autonomous University of Puebla (UPAEP)
Puebla, Mexico, 72700
http://upaep.mx/roboticlab
Email: robotdonaxi@gmail.com

***Abstract* — In this paper we described the Donaxi Service Robot. Donaxi belong to the Control and Robotics Laboratory of the Popular Autonomous University of Puebla (UPAEP) in Mexico. The hardware of this robot was built from scratch by students of this university. The software is based in ROS with some own development. Donaxi can navigate in a house environment, is capable of interact with its owner through voice and gestures, can recognize objects, has an arm to manipulate objects, can recognize people and connect with a smart house. Donaxi's software is modular and its central control is based on behavior trees. This robot has participated in some national and international competitions like Mexican Robotics Tournament and Robocup, since 2004 and 2009 respectively. Novel results are presented here in the area of face recognition; we propose the Hybrid Algorithm for Face Recognition using an Evolutionary Software Engineering. The principal aim was the construction of a face recognition system in order to be implemented in the service robot Donaxi, delimited by the Who is who test which is part of the RoboCup's tests set,using an evolutionary development strategy of triple iterations.**

I. INTRODUCTION

In Puebla, Mexico, there is one robotic's laboratory that belongs to the Popular Autonomous University of Puebla State, where a young group of students has the passion of creating and programming robots for different purposes. One of this robots is Donaxi, a service robot who has participated in RoboCup 2009, 2010, 2012 and 2013, and has gradually improved some skills, changing its structure to improve the interaction with their environment in the RoboCup tournament. Donaxi integrates different software modules of the robot in ROS, improve algorithms for natural language processing, vision, handling and navigation. Also can distribute sensor processing and analysis in embedded systems. Finally, Donaxi can communicate with the enviroment (users and home itself) through network protocols. Four systems will be used to support embedded vision systems, handling, performance and learning. Electromechanical design was improved; it was considered the inclusion of two arms to carry out operations in object manipulation. We designed the exterior structure of the robot according to their main activity: assistance for the elderly. Software elements are tested in parallel in the Gazebo simulation system, such as the hardware design is created and tested in SolidWorks as shown in Figures 1 and 2.

Figure 1: SolidWorks Model

II. THE ROBOT DONAXI

A. Team

The Donaxi team is conformed by under and post grade students. Is a multidisciplinary team formed by members with specialities in diverse engineerings like mechatronic, bionic, computation and software. All the members are leaded by Dr. Hector Vargas in order to achieve the goals that allows Donaxi to obtain the qualities to participate in various competitions.

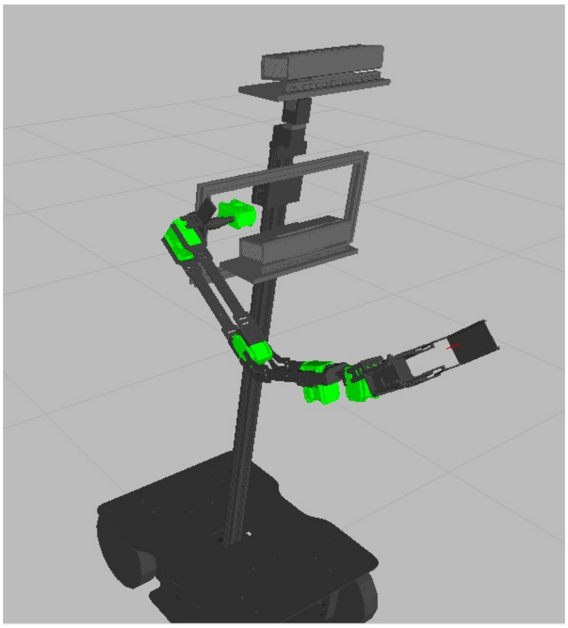

Figure 2: Gazebo Model

The building of Donaxi has gone through many stages and it has been created different versions of the same. Each version has left important experiences and learning about the best hardware and software configurations that facilitate the performance of a service robot in different conditions. Donaxi's philosophy is about inheritance knowledge, passing this knowledge on from generation to generation through all the teams that have worked in the project.

B. Hardware

We can see in Figure 3, all the elements forming part of Donaxi's hardware, starting with DC motors connected to the omnidirectional wheels, for this purpose we have four wheels. For navigation and planning it has a laser, encoders as position sensors of the wheels, gyroscopes and Kinects 2. The torso is moved by a 24 volts DC motor (Figure 4). For the manipulation part, the robot now has a functional arm formed by Dynamixel Servo Motors. For the first time, it's planned to introduce the second arm before the competition. Both arms are composed each one(as mentioned before) by 6 Dynamixel Servo Motors, 5 of them used for the articulations and 1 for the parallel gripper, producing arms of 5 degrees of freedom with one end effector. The neck has two Dynamixel Servo Motors making the pan-tilt movement. The neck gives support to a Kinect 2, that holds a tablet to interface with the user, and a router to connect with the smart house.

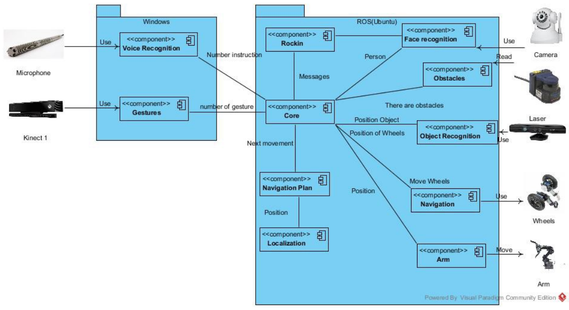

Figure 3: Global architecture software diagram

Figure 4: The robot can move the torso until the arms reach an object on the floor.

C. Software: Control Architecture

The software developed for Donaxi is made in modular way and tries to maintain software engineering quality standards in extensibility and maintainability. To achieve this, each team member develops software corresponding to its assignment, documents and keeps it and posted on GitHub. That permits the team leader monitoring all the development. The location of all the developed software is in the webpage https://github.com/donaxi. As a contribution to the robotics society this GitHub has public access permission.

Donaxi use an architecture based on a hybrid scheme made up from a number of elements, called modules, which are grouped into three layers [1] (see figure 5).

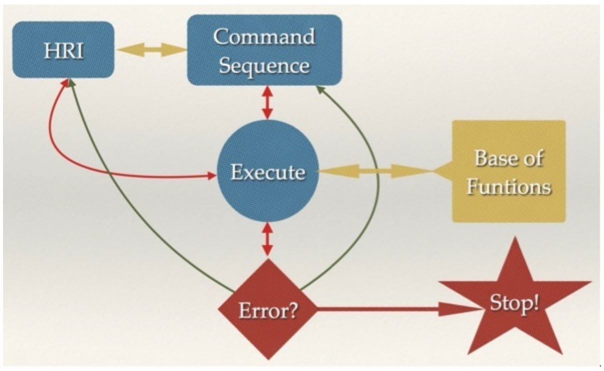

Figure 5: General View of Control Architecture.

1. Deliberative layer maintains and reasons about a symbolic and hierarchical representation of the environment. Such an internal world model is used to produce plans (sequence of human/robot actions to solve a goal) as well as to facilitate human-robot interaction (HRI).

2. Execution and control layer generates the sequence and monitors the execution of plans taking into account the information collected from the functional layer and the sensors of the robot. According to such information, it may tune the behavior of certain modules, i.e: if some dangerous situation is detected by the laser or the kinect, the execution layer will reduce the speed of the motors in order to maintain the security of the people around and the security of the robot itself.

3. Functional layer is formed by a number of groups of skills called functional groups, which physically perform actions, like navigation, manipulation, etc. Each functional group may entail different ways to accomplish a particular type of action. For example, the robot can traverse between two spatial points either by a reactive algorithm, by tracking a computed path, or by the user manual guidance.

To support the all-level human integration we claim in this paper, we have used the common module structure shown in Figure 5 for all the modules in the architecture. The common module structure integrates human and robot abilities through the so-called skill units. Each common module structure may contain a variety of (human or robotic) skill units that materialize a particular type of ability, like producing a plan, checking for risky situations, moving between two locations, manipulating objects and others.

D. Face Recognition using an Evolutionary Software Engineering

Once the techniques and methods for the face detection and recognition have been established, it is possible to present a complete algorithm, which would be integrated as shown below.

1) Software Engineering Methodology: Once the conceptual shape for the face recognition algorithm is defined, the next phase consists in the construction and implementation of the algorithm within software which allow integration in a service robot.

Nowadays, the software development focused on research faces problems in time estimation, costs and work, some of these problems are due to current technological limitations (insufficient software, hardware restrictions, non suitable algorithms, etc.) and others are due to a lack of methodology during the planification time, in order to solve this, a development methodology was adopted, which consists in an engineering process and a life cycle. By doing this, the algorithm conceptual design was seen and done as a software project, to achieve this statement, all previous, present and posterior documentation was generated for the face recognition software.

2)PSP: Due to the presented characteristics of the project, it was decided to employ the PSP process. This process allow us [3] to know us as developers and because there is not much information about process implementation in research environment, it results suitable in order to identify behaviors, routines, and common mistakes that are committed during the software development and with all this information be able to improve development quality of the involved people.

Because it is a research project, all requirements were perfectly defined and limited since the beginning and only the six basic process phases were considered (planning, design, coding, compiling, tests, and post mortem). For documentation ends, proper documents for each phase were generated, through them history can be created and can be used for calculations and future estimations.

The metrics are basic in any software process [15], for this project the following were utilized: Time, Size and Defects.

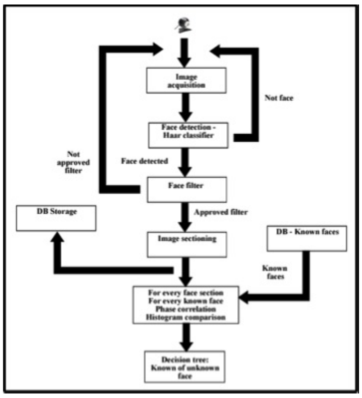

Figure 6: Face Recognition Algorithm.

3)Evolutionary Life Cycle: In a parallel way to the engineering process choice is located the model election of a life cycle for the development. A correct choice of life cycle will allow to have a good control and tracking during software elaboration, by the other hand, a wrong choice of life cycle will provoke delays, cost increases and required work time.

The evolutionary model, which can be also named incremental and iterative, is based in the realization of little prototypes which are analysed in order to know how far is the development from the final customer requirements, and from this analysis a new prototype is created which includes the customer requirements. This process is realized in a cyclical way, increasing the aims and requirements to cover in determinate portions by the developer (generally little portions) until achieving the complete requirements.

Considering the project type to be realized [5] (research and robotics) the application of an evolutionary life cycle is convenient due to it generates an advantages series: Evolutionary software developing allows evaluating if the proposed method in the algorithm is suitable, or if it is needed to propose a new solution; the no-ends ways detection (due to research) is done at the early moments of development, requirements are divided so the system construction allows to observe significant advances; finally this project was integrated in a service robot and the requirements could be unstable but evolutionary model is planned for that.

For the development of the face recognition system, the process was divided by three iterations:

First iteration: During this first iteration phase it is included face detection which involves image acquisition, implementation of the Haar classifier algorithm, the confirmation filter, the image severing and disk storage.

Second iteration: Includes face recognition phase, involving reading disk, the implementation of phase correlation, the implementation of the histograms comparison and tree configuration for decision-making.

Third iteration: Finally, it was incorporated a third and final iteration to adjust the face recognition system to the activity required by the service robot. Additionally, this iteration was used to add enhancements and optimizations to the previous iterations.

4)Face Recognition Results: The face recognition system was designed in order to solve the vision of the service robot Donaxi presented by the Who is who test corresponding to the RoboCup 2012 competition. In order to test its efficiency many tests were realized simulating this specific activity. The tests consisted in system training with a number of known people and then show the system various people (known and unknown people) to see if the system was able to identify if a person is known or unknown. The people was located in an hall room in different places and postures (stand up or sit) and the only condition (indicated in RoboCup Rulebook) was that the people must be viewing to the camera in every moment. The robot navigates in the room and search for people.

During those tests, the system was exposed to a total of 205 actions situation (decisions) where the robot had to take the decision if the detected face in the image was known or not, the following results were obtained:

During the tests development, the performance of the system was observed, this performance in real time was positive, achieving results with high percentage of certainty. *5)Software Engineering Results:* Parallel to the results of efficiency of the algorithm were obtained the results shown by the methodology and used software process.

There are several ways to analyse and interpret results from software engineering in a project; this paper will center the analysis on the three key metrics of a project (times, sizes and defects).

Time: Time estimations are vital in the software development, a good estimate can produce quality software in a timely manner, while a poor estimate generally increases the time and required costs and reduces the quality of the final product. The software engineering allows us to analyse our times and take steps to improve our estimations. The Figure 6 shows the obtained results from the recollection of time estimations data for each development phase. It can observe the positive evolution of the estimations, which were decreasing its error edge between planned time and real required time.

Size: For software size measurement was employed the Lines of Code unit (LOCs). The estimated sizes are crucial in developing quality software because it is what gives us the information to calculate the time and complexity required for a system development. Table 2 shows the estimated sizes versus actual sizes obtained in the three iterations of development.

Time in Phase — First Iteration

Phase	Plan	Actual	To-Date	To-Date%
PLAN	200	202	202	12.7%
DLD	400	253	253	15.9%
DLDR	50	27	27	1.7%
CODE	200	374	374	23.5%
CR	40	26	26	1.6%
COMPILE	10	24	24	1.5%
UT	200	457	457	28.7%
PM	100	231	231	14.5%
Total	1200	1594	1594	

Deviation = 32%

Time in Phase — Second Iteration

Phase	Plan	Actual	To-Date	To-Date%
PLAN	240	229	229	8.9%
DLD	400	418	418	16.2%
DLDR	120	62	62	2.4%
CODE	480	720	720	27.9%
CR	90	106	106	4.1%
COMPILE	30	40	40	1.5%
UT	420	843	843	32.6%
PM	230	165	165	6.4%
Total	2010	2583	2583	

Deviation = 28.50%

Time in P — Third Iteration

Phase	Plan	Actual	To-Date	To-Date%
PLAN	342	351	1651	30.3%
DLD	104	107	503	9.2%
DLDR	114	200	631	11.6%
CODE	177	171	842	15.4%
CR	23	105	194	3.6%
COMPILE	288	250	1344	24.6%
UT	35	64	196	3.6%
PM	17	31	95	1.7%
Total	1100.0	1279	5456	

Deviation = 16.27%

Figure 7: Face Recognition Results

The evolution of size estimates shows a positive performance by reducing the error margin advancing the development iterations.

Defects The defects registration allows to visualize the most committed mistakes develop-ing software. The Fig. 8 shows the graphs of required time for defects correction and the quantity of defects by type. The function defects and environment defects need a higher fixing time in comparison with the interface errors, syntax errors or allocation errors. This is the normal behavior for a software development project.

E. Manipulation

One service robot can not be considered completed without a manipulation system, in the case of Donaxi (as we mentioned before) we've considered the implementation of two autonomous manipulators, these manipulators are formed by 5 Dynamixel Servo Motors for the articulations, and 1 for the end effector system. That way we can have 5 DOF, enough for reaching objects relatively close to the robot. For the software part, the Orocos Toolchain seems very effective. Whit this tool the robot can calculate the poses needed to reach the target that the vision system provides. Along with the tf structure provided by ROS, the manipulation part worked in a very accurate way. The results that the simulations throw were very similar to the results that the actual arm gets. Beside this, the arms are attached to the mobile torso, so the workspace increases and now the robot can reaches targets in the floor.

E. Navigation

For the navigation part we could have choice either the differential or omnidirectional movement, for a lot of reasons the omnidirectional navigation was selected. The most important algorithms used in this part are GMapping and SLAM. For the odometry encoders, gyroscopes and magnetometers as well as a laser are used. An algorithm used for finding each wheel speed was also developed. We use a Pololu's master servo controller and a serial communication for the control of the whole base.

III. CONCLUSION

Donaxi is a service robot that has been evolving for many years. There are still many improvements on its currently abilities as well as there are other features to add. In any case, this robot is able to meet some basic tasks that allows it to participate in national and international competitions, as has been the case until now.

The effort made by all the members of the development team has constructed the robot and developed new skills in the field of service robotics. Likewise, this project serves as a framework for all investigations being carried out at the undergraduate and graduate levels. The participation of Donaxi in RoCKIn 2015, undoubtedly will give a major boost in efforts to have a robot that can assist people in their homes.

References

[1] Harold Vasquez, Hector Vargas and Luis Sucar. *Using gestures to interact with a Service Robot using Kinect 2*. Research in Computer Science, 96, 85-93, 2015.

[2] Aldo Martinez, Hector Vargas, and Jose Perez. *An Evolutionary Software Engineering Approach Applied to a Hybrid Algorithm for Face Recognition.* October 2012 Brazilian Robotics Symposium and Latin American Robotics Symposium, published by IEEE Computer Society.

[3] Hector Vargas, Esperanza Medina, Daniel Martinez, Edson Olmedo, Gerson Beristain. *Hybrid Algorithm to Human-Face Detection and Tracking Unrestricted"*. Springer Verlag in Advances in Soft Computing Series.

[4] Hector Vargas, Esperanza Medina, Daniel Martinez, Edson Olmedo, Gerson Beristain. *Human-Robot Interface using Face Detection and Recognition, for the Service Robot "Donaxi"*. 1st International Workshop on Intelligent Domestic Robots, 26th-27th of June 2012 Guanajuato, Mexico.

[5] Hector Vargas and Aldo Martinez. *"Hybrid Algorithm for Face Recognition using an Evolutionary Software Engineering"*. Applied Mechanics and Materials Vols. 423-426 (2013) pp 2543-2546, (2013) Trans Tech Publications, Switzerland ISSN: 978-3-03785-888-2.

[6] Hector Vargas, Edson Olmedo, Daniel Martnez, Vctor Poisot, Agustin Perroni, Andres Rodriguez, Omar Granillo, Carlos Merino, Carlos Peto, Wesli Antonio, Carlos Carbonel, Aldo Orozco. *"Project Donaxi@HOME Service Robot"*. Applied Mechanics and Materials Vols. 423-426 (2013) pp 2543-2546, (2013) Trans Tech Publications, Switzerland ISSN: 978- 3-03785-888-2.

[7] C. Wuthishuwong, Chaiyaporn Silawatchananai, Manukid Parnichkun *Navigation of an Intelligent Vehicle by using Stand-alone GPS, Compass and Laser Range Finder*. Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics Bangkok, Thailand, February 21 - 26, 2009

[8] T.Laue,T.Rofer. *A behavior architect for autonomous mobile robots based on potential fields.* Lecture note in artificial intelligent, Springer, 2005.

[9] R.C.Arkin. *Behavior-based robotics*. MIT Press, Cambridge, Massachusetts, 1998.