Ursus Team

M.A. Gutiérrez[1], L.J. Manso[1], L.V. Calderita[1], P. Bustos[1], F. Cid[1], M. Paoletti[1],
A. Sánchez[1], J.P. Bandera[2], J. Martínez[3], M. Martínez[4], J. García[4]
Telephone: +34 927257259. Email: pbustos@unex.es
Team web site: http://ursus-team.org.

Abstract—**The Ursus team is a multidisciplinar group of students, academics and robotics enthusiasts from different universities that work together in designing, building and programming intelligent service and social robots. Our latest platform is Ursus, a two-arm mobile manipulator designed and built by our team that can be acquired commercially. We have also developed a component-oriented, distributed robotics framework called RoboComp that is used as the underlying software to build, integrate, maintain and deploy all algorithms and functional parts running in the robots. Finally, our team is currently working on a cognitive multimodal HRI-oriented architecture called RoboCog that integrates deliberative planning with reactive and opportunistic behaviors using a rich representation of the internal and external worlds of the robot.**

I. URSUS HARDWARE

Ursus hardware has been completely designed and developed from scratch at the Robotics and Artificial Vision Laboratory (ROBOLAB) from the University of Extremadura, Spain. It was initially designed for rehabilitation and therapy assisting purposes but it has advanced so far that is currently capable of performing a wide range of day to day home tasks. It has several different sensors, along with 7-DOF arms, a tablet display for human robot interaction, a differential base, a 3-DOF neck, a grip and a tray for holding things while moving around. See figure 2 for more details.

A. The base

The current base of Ursus is an evolution of an older open-hardware robot platform called RobEx [10]. It is a differential platform (Figure 1) with the driving wheels placed in the front and a caster wheel in the back.

The wheels have a radius which is big enough to absorb the irregularities of the floor of indoor environments while providing good motor-to-wheel power transfer and speed. The platform's case contains all the control electronics and a battery which allows the robot to autonomously operate for periods of up to two hours. The motors are equipped with optic incremental encoders to provide the odometry with precise information.

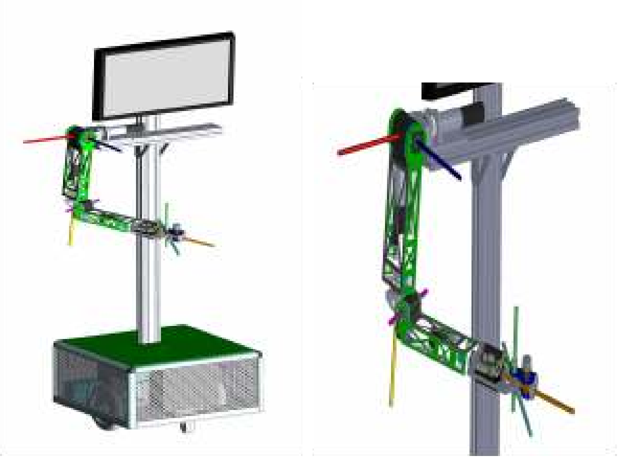
Fig. 1. Final CAD drawing of robot with motor axis in color.

Although RobEx is the current differential base used in Ursus, a new omnidirectional base built with mecanum wheels, which will allow Ursus to move more efficiently, is currently in development. It is expected to be ready for RoCKIn 2014.

B. The arms

The robot is equipped with two 7-DOF arms with two-fingered grippers capable of manipulating object of up to a kilogram. Figure 1 shows a cad model of the arm design. The fingers and the *palm* of the grippers were endorsed with sensors used to provide tactile feedback. The disposition of the motors is anthropomorphic and the wrist has been slightly turned towards the central axis of the body to facilitate grasping procedures. The motors are driven by Faulhaber modular controllers connected via CAN bus and offering a USB interface to the computer. The arm is also controlled by a bus of EPOS that can be accessed from the computer through a simple C API. A more detailed explanation of the arms desing can be found in [9].

C. The head

The head of the robot has a tablet and an RGB-D camera fixated to a pan-tilt structure. The tablet runs an Android program used to enable the robot express its *emotional state* through a face interface and provide other kind of visual information. It also holds an RGBD camera which is an *Xtion*

---

[1] Robotics and Artificial Vision Laboratory, University of Extremadura. Avd. de la Universidad sn, Cáceres, Extremadura, Spain. Telephone: +34 927257259. Email: pbustos@unex.es.

[2] Electronic Technology Department, 35 Louis Pasteur Blvd., University of Malaga, Spain. Telephone: +34 952131352, Email: jpbandera@uma.es

[3] Intelligent systems and data mining, Computer Science Research Institute, España Ave, Albacete, Spain. Telephone: +34 967599200 Email: jesus.martinez@uclm.es

[4] Planning & learning research group, Computer Science and Engineering Department Carlos III University of Madrid. Avenida de la Universidad, 30, 28911 Legans, Madrid, Spain. Telephone +34 916245981 Email: momartin@inf.uc3m.es

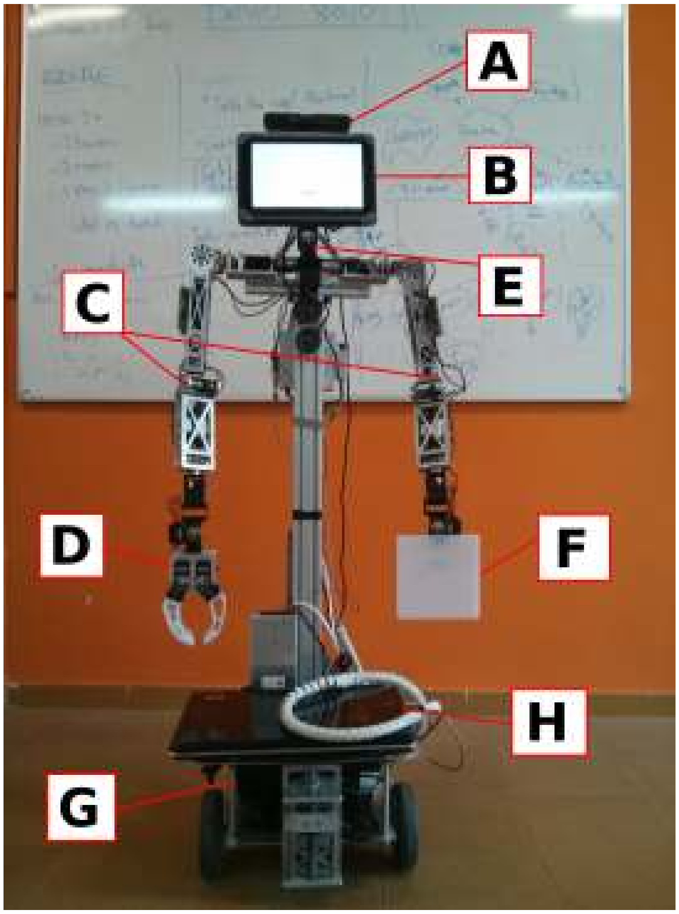

Fig. 2. This picture shows the Ursus robot: A) RGBD sensor B) tablet display for the robot face C) 7-DOF arms D) 2 finger grip for manipulating objects up to a kilogram E) neck with 3-DOF anthropomorphic configuration F) tray for holding objects H) main computer for executing low level components G) Current differential base .

Pro live from Asus. It is used to model objects and humans founds in the surrounding environment.

II. URSUS SOFTWARE: ROBOCOMP

as well as middleware used The software of Ursus is built on top of RoboComp, an open source component-oriented robotics framework which provides several components for solving some of the most essential tasks, tools for robotics software development and libraries for the most common algorithms and data structures.

RoboComp is a model-driven, component-oriented framework built around three key elements: a component model, a communications middleware and a set of tools that facilitates the writing and maintaining of robotics code. It started in 2005 as a means to create and reuse code written by different people and that was meant to be used in many different robots. The central idea is to define a processing and coding entity that can be created and maintained largely decoupled from the rest of the system. These units or components are full fledged processes when running and occupy its own subdirectory in the global code repository, generating a readable history of development. They communicate with other components using a public interface and through an underlying communications middleware.

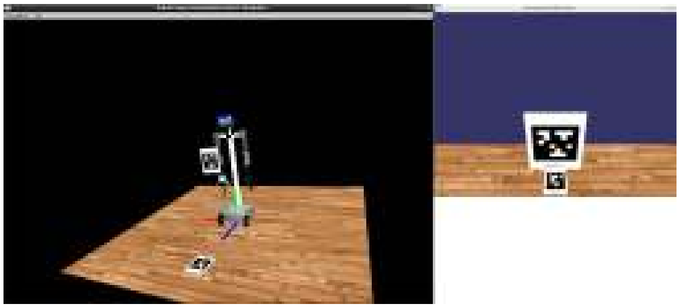

Fig. 3. Screenshot of the RoboComp InnerModel Simulator with a model of the robot Ursus.

Building on this generic idea, RoboComp is now the result of many years of further elaboration and adaptation to our everyday research and engineering activity. Nevertheless, many more improvements are being evaluated now due to the increasing complexity of current robots and their control architectures. The repository holds now more than one hundred components, along with classes and tools specifically designed to improve and ease the robotics software designer experience. It covers functionalities of different robotics and artificial vision topics mainly through integration of third party libraries. RoboComp is the result of the collaboration of many people from different Universities from all over the world along with different programs such as Google Summer of Code or the European Space Agency Summer of Code in Space. More detail of the initial design of RoboComp is given in [8], [5] and in RoboComp's web page: http://robocomp.org.

GSoC

A. *RoboComp InnerModel Simulator*

One of the main efforts taken recently in RoboComp has been the design and construction of a robotics simulator. This effort has been partially mitigated by the reuse of the InnerModelDSL [5] elements and of the OpenSceneGraph visualization technology that was already employed for monitoring and debugging purposes. Combining these components along with a careful design has taken us to the RoboComp Innermodel Simulator (RCIS), a 3D simulator. The most important feature of this simulator is that it is also a native RoboComp component. Being so, it can implement all the interfaces of the existing components in the hardware abstraction layer, i.e. cameras, lasers, kinect, motors, bumpers, ultrasound, tactile and any others that may come in the future. The rest of the components in a certain deployment graph can communicate to these interfaces as if they were the original components, facilitating enormously the development cycle of complex algorithms. Figure 3 shows a screenshoot of RCIS.

Having complete control over the simulator kernel allows us to adapt it to our needs. For example, it is very useful to be able to activate or deactivate the physics engine or to modify

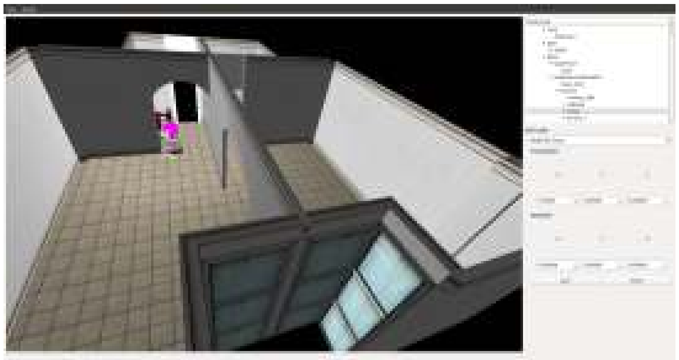

Fig. 4. Screenshot of the RoboComp Innermodel Editor with a model of a robot and the innermodelDSL tree ready for edition.

the level of noise in the simulated sensors and actuators, Also, we plan to introduce semi-autonomous humans in the system to simulate and develop HR interaction behaviors. We should be able to, for example, connect the OpenNI tracking software to a synthetic RGBD sensor located in the robot inside the simulator and track the evolution of the synthetic human figure. As a side effect of these developments we have built InnerModelViewer 4, a graphic editor of InnerModelDSL files. This tool has proven of great utility in the modeling of new robots and scenes using meshes and partial models created in other modeling programs such as Blender.

Finally, a last feature of RCIS that is hard to find in other simulators is that it provides a flexible interface to control objects on the fly. This interface is implemented as a RoboComp interface -an IDSL- so it can be accessed from any other component. These components can create and transform elements in the scene graph. As a consequence, RCIS may be used as an internal modeling and simulation system. This use of a full-fledged simulator as a cognitive module has been proposed before in theories of consciousness [11] and we plan to include it as the central element of a new cognitive architecture being developed on top of RoboComp, called RoboCog. The simulation capabilities of RCIS can be used internally to predict the outcome of the robot actions on its represented environment. The robot itself would occupy the central place in the simulator and the objects and agents around it would be modeled and updated by the robot's perceptive system. Sensor models in the original RCIS now can be used to generate the sensorial data that the model of the robot would perceive when interacting with its modeled environment. Furthermore, the robot's self-model could be temporarily cloned to execute and evaluate plans computed by an opportunistic task-planner. The unfolding of alternative courses of action in the internal simulator and the interleaved execution of the partially validated plan creating a real course of action, is part of our current ongoing research on RoboCog.

B. RoboComp Components

the architecture. One of the main reusable resources RoboComp provides to developers is the existence of already functional and tested components. Among these components there are a wide range of essential functionalities that eases the deployment of robot software.

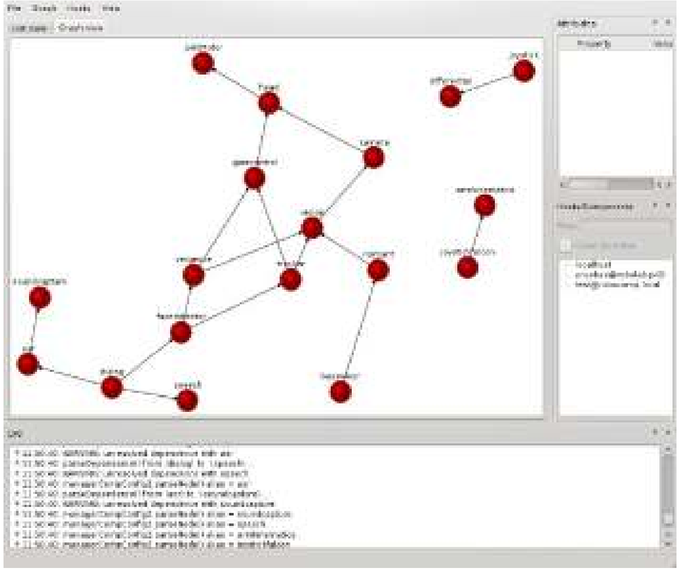

Fig. 5. Screenshot of the RoboComp Manager tool managing a network of components.

A group of hardware abstraction layer (HAL) component is provided. These component directly access the hardware sensors and actuators and provide other components or tools with an easy interface to deal with hardware. Currently HAL interfaces like RGBD, Camera, DifferentialRobot, GPS, IMU JointMotor, Joysitic Laser or Micro ar provided.

There are also components for high level algorithms such as path planning, object perception, artificial intelligence or kinematics problems. Huge and complex networks of components can be easily deplyed in order to achieve high level tasks with the robot. For managing this networks RoboComp also provides a managing tool (Figure 5)More info on all the components available can be found on the RoboComp's wiki page: http://wiki.robocomp.org/

III. MAIN RESEARCH AREAS

A. Navigation

The group is working on a new Navigation framework that will solve some integration problems that can be seen in most current approaches. Our idea is to build a layered system composed of three concurrent processes: a trajectory following module [6], an elastic bands module [12] and a RRT path planner [7], accessing a common shared data structure that represents the current path. Each process works at a different time scale and access different sensors, such as the odometry system, laser and RGBDs and the internal maps of the environment.

Map localization and building is done using the GMapping library.

B. Visual perception and cognition

The deliberative layer of the Ursus' architecture makes use of a new planning system —a planner and a new language— which allows the robot to reason about how to perceive the elements in the environment. The robot represents the world in a graph-like structure that contains symbols, stored as nodes, and relationships between them, represented by edges. Additionally, each symbol can contain a map of metric properties. The world representation does not only contain physical elements but also other information that can be necessary for the robot, such as the estimations of the emotional status of the humans or the objects they might be looking. The planning system of Ursus can also take humans into account and, to some extent, a theory of mind for humans.

C. Speech perception

The development of systems based on speech is one of the most important fields of research of the framework RoboComp. These systems acquire, process and generate the human voice, within a controlled interaction with non-trained users to exchange information. The current processes of recognition and generation of speech presented a novel solution for a non-invasive communication using the natural language of humans. The speech recognition process of this proposal composed by two separate steps: transcription generation and comprehension.

The first step processes the audio source and obtains the most reliable text transcription. This is completely carried out in the WinKinectComp using the Microsoft Kinect Speech SDK. Transcription generation is performed by using two internal key elements: the acoustic model and the language model. The acoustic model represents the probability of obtaining an input utterance x given a sequence of words w (transcription). It is directly provided by the Microsoft Speech SDK (for several languages such as Spanish or English). The language model scores the transcription w using the joint probability of the sequence of words. The probability for each word w_i depends on the list of previous words $w_i - 1 ... w_i - n$.

The language model is generated by following the n-gram model, where n defines the number of words considered in the joint probability. We propose the generation of 3-grams models from some of the available corpus fitting the task requirement. The generated model could be then compiled using the Microsoft Speech SDK tools, obtaining a Kinect compatible grammar. Thanks to this development, each received audio (suitable for being modelled with the generated grammar) would result in a text transcription used as input for the comprehension step.

The comprehension step should assign a semantic label to any of the input transcriptions. Therefore, it can be modeled as a classification problem and managed using a Bag of Words (BoW) procedure in conjunction with any type of classifier. The BoW training sequences should be generated taking into account the comprehension capabilities provided to the robot.

Finally, the speech generation can be performed using some of the standard text to speech solutions, such as Verbio or festival.

D. Human robot interaction

RoboLab has made an important effort to develop methods and mechanisms in the field of the affective human robot interaction. In particular, our team has worked in a multimodal emotion recognition system, where different information channels (*i.e.*, modes) are integrated for estimating the human emotion during the interaction [2]. Audio signal is analyzed to detect emotional component in the human speech. Besides, the visual information acquired by RGB or RGB-D cameras is used for recognizing human facial expressions [4], [13], and also for analyzing the affective behavior of the human body gestures [3].

Based on the idea of learning methods where the imitation presents the main solution to the transfer of knowledge from the user to the robot, RoboLab is also working in the use of affective affordances, as a new definition that extends the classic concept of perceptual affordances [1]. We work with the hypothesis that the robot is able to predict and learn how to influence in the human emotional state by generating emotional facial expressions or interacting with the environment.

IV. INNOVATIVE TECHNOLOGIES

A. RoboComp

RoboComp is unique in its Domain Specific Language [14] components development, as well as some other tools and resources evolved from the direct roboticist needs. Its resources speed up the robotics software development and deployment. It also contains a wide range of original components containing innovative algorithms for different purposes. As said

B. Hardware

Hardware is all designed and developed from scratch which constitutes an innovation in terms of design. Special designs and development have been done taking into account the specific roboticists needs. This hardware can be reused in future robot designs helping robot hardware designers.

V. REUSABILITY OF THE SYSTEM

The RoboComp framework is reusable for almost any robotic platform. The component oriented paradigm that the framework follows makes possible to reuse almost any component for a wide range of tasks and technologies. Also several tools such as the simulator, the RCMonitor (to monitor components) and a DSLEditor (to develop components using our own Domain Specific Languages [14]).

Currently the framework is intensively by several research groups across the world and in a wide range of different robotic platforms. *i.e.* all the robots here run RoboComp: http://goo.gl/ZTjhPS

Ursus is robotic platform ideal for most robotic applications. This robot has a size similar to a human, and it moves using wheels. It is equipped with two arms able to manipulate and move around simple objects. This arm has 7 degrees of freedom. All the processing can be done on-board, in a computer allocated in the lower part of the robot. We can connect multiple sensors to the robot, being a RGBD camera the current main source of information. The possibilities that the platform offers is almost infinite, a huge range of tasks can be

The unique and cost-effective design of Ursus makes it possible to also reuse the different parts of the robot. The robotics arms can be mounted on any other platform in order to reach objects in different ways or perform different kind of tasks. The head and display app can be moved to some other robots in order to provide them with a better human robot interaction. The base can also be used in other robots along with the same RoboComp components in order to provide them with motion capabilities without much effort.

VI. APLICABILITY AND RELEVANCE TO DOMESTIC ROBOTICS

As a humanoid robot Ursus is able to perform domestics tasks specially those involving moving objects around and opening/closing doors or windows. Many targets can be achieved with this abilities. Tasks such as helping elder people in day to day tasks, cleaning the table, answering the door, opening/closing windows depending on temperature, closing or opening water taps or bringing objects to users can be performed with Ursus.

REFERENCES

[1] F. Cid, A.J. Palomino, and P. Núñez. A new paradigm for learning affective behavior: Emotional affordances in human robot interaction. In *Workshop of Physical Agents*, pages 47–52, 2013.

[2] F. Cid, J.A. Prado, P. Bustos, and P. Núñez. Muecas: A Multi-Sensor Robotics Head for Affective Human Robot Interaction and Imitation. *Sensors*, 14(5):7711–7737, 2014.

[3] C. Doblado, E. Mogena, F. Cid, L.V. Calderita, and P. Núñez. Rgb-d database for affective multimodal human-robot interation. In *Workshop of Physical Agents*, pages 35–40, 2013.

[4] F.Cid, J.A. Prado, P. Bustos, and P. Núñez. A Real Time and Robust Facial Expression Recognition and Imitation Approach for Human-Robot interaction using gabor filtering. In *In IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2188–2193, 2013.

[5] Marco A. Gutirrez, A. Romero-Garcs, P. Bustos, and J. Martnez. Recent advances in robocomp. *Journal of Physical Agents*, 7(1):38–47, 2013.

[6] Gabriel M. Hoffmann, Claire J. Tomlin, Michael Montemerlo, and Sebastian Thrun. Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In *26th American control Conference*, 2007.

[7] Steven M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, 1998.

[8] L.J. Manso, P. Bachiller, P. Bustos, P. Nu nez, R. Cintas, and L. Calderita. Robocomp: a tool-based robotics framework. In *Simulation, Modeling and Programming for Autonomous Robots*, pages 251–262, 2010.

[9] Francisco Martín, José Mateos, Francisco J Lera, Pablo Bustos, and Vicente Matellán. A robotic platform for domestic applications. In *In to appear in Workshop of Physical Agents*, 2014.

[10] J Mateos, A Sánchez, Luis J Manso, Pilar Bachiller, and Pablo Bustos. Robex: an open-hardware robotics platform. In *Workshop of Physical Agents*. Citeseer, 2010.

[11] H. Owen and R. Goodman. *Robots with internal models*, volume 10. 2003.

[12] S. Quinlan and O. Khatib. Elastic bands: connecting path planning and control. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 802–807 vol.2, May 1993.

[13] P. Romero, F. Cid, and P. Núñez. A novel real time facial expression recognition system based on candide-3 reconstruction model. In *Workshop of Physical Agents*, pages 41–46, 2013.

[14] Adrián Romero-Garcés, Luis Manso, Marco A. Gutierrez, Ramón Cintas, and Pablo Bustos. Improving the lifecycle of robotics components using domain-specific languages. *CoRR*, abs/1301.6022, 2013.