

# Urbano, an Interactive Mobile Tour-Guide Robot

Diego Rodriguez-Losada, Fernando Matia, Ramon Galan,  
Miguel Hernando, Juan Manuel Montero and Juan Manuel Lucas  
*Universidad Politecnica de Madrid*  
*Spain*

## 1. Introduction

Autonomous service robot applications can be divided in two main groups: outdoor and field robots, and indoor robots. Autonomous lawnmowers, de-mining and search and rescue robots, mars rovers, automated cargo, unmanned aerial and underwater vehicles, are some applications of field robotics. The term indoor robotics usually applies to autonomous mobile robots that move in a typical populated indoor environment. Robotic vacuum cleaners, entertainment and companion robots or security and surveillance applications are also some examples of successful indoor robot applications.

Probably, one of the first real world applications of indoor service robots has been that of mobile robots serving as tour guides in museums or exhibitions. Such one is an extremely interesting application for researchers because allows them to advance in knowledge fields as autonomous navigation in dynamic environments, human robot interaction, indoor environment modelling with simultaneous localization and map building, etc., while also serving as a showcase for attracting the general public as well as possible investors.

We have developed our own interactive mobile robot called *Urbano*, especially designed to be a tour guide in exhibitions. This chapter describes the *Urbano* robot system, its hardware, software and the experiences we have obtained through its development and use until its actual mature stage. This chapter doesn't pretend to be an exhaustive technical description of algorithms, mathematical or implementation details, but just an overview of the system. The interested reader will be referred to more specific bibliography for these details.

The rest of the chapter is structured as follows: This section presents the related work, other existing systems, as well as our motivation to develop our own robot. Section 2 presents an overview of *Urbano*, the description of its hardware and also the software components in which the robot control is structured. These components are afterwards described in subsequent sections: Section 3 describes the feature based mapping and navigation subsystem, while the interaction capabilities including our own proprietary voice recognition and synthesis engine will be described in section 4. Section 5 briefly describes the web based remote visit that *Urbano* is also able to perform. The integration of all these components is managed through a programmable kernel that allows a high level management of all modules, described in section 6. The chapter ends with the presentation of some successful real deployments of *Urbano* in section 7, and our conclusions in section 8.

### 1.1 State of the art

As previously stated, *Urbano* is not the first mobile tour-guide robot. There have been many others that have served as a reference for the mobile robots research community.

Probably, the first one was Rhino (Burgard et al., 1999) robot from Bonn University, followed by Minerva (Thrun et al., 1999) from Pittsburgh, CMU and Bonn Universities. These robots meant important advances in mobile robot mapping, localization and reactive control, using commercial robot platforms as a hardware base for developments. Another approach was Sage (Nourbakhsh et al., 1999) robot that focused on a more commercial vision that was later accomplished by Mobot Inc. Sage robot uses artificial coloured landmarks in the environment to achieve a robust navigation. These robots were later followed by other derivative works as the robotic assistants for the elderly called Flo and Pearl (Montemerlo et al., 2002).

These robots were afterwards followed by many others from different universities and research centers around the globe, as Albert from Freiburg University (Germany), Lefkos from Forth institute (Greece), Tito from Valladolid University-Cartif (Spain), Carl from Aveiro University (Portugal), just to cite some examples. All of them were interactive mobile robots to serve as tour guides.

Recently, several other robotic tour guides have been developed and commercialized by some companies as the BlueBotics RoboX. Eleven RoboX were involved during 5 months in the Robotics exhibition at the International Expo 02. Industry giants have also built their own robotic guides, as Toyota TPR-Robina operating in their company headquarters. Likewise, Fujitsu developed the Enon robot that served as a guide in the Kyotaro Nishimura museum.

### 1.2 Previous work

Our initial trials with interactive robots were carried out with Blacky (Fig. 1), a robot for tour-guiding, tele-visit and entertainment. We developed navigation algorithms focussed on indoor, populated, complex and low structured environments.



Fig. 1. Blacky robot in a fair

Blacky was a MRV4 mobile platform from Denning Branch, Inc., with a ring of sonars, a three wheeled synchro-drive system, a radio link for wireless Ethernet connection, a

horizontal rotating laser called *LaserNav* able to identify up to 32 different bar coded passive landmarks, and loudspeakers for voice synthesis.

The robot implemented reactive behaviours such as follow corridor, go to point, escape from minimum, border by the right or the left or intelligent escape, as well as tasks such as walk along this corridor or take this corridor in this direction.

As the robot moved autonomously in a populated public environment making oral presentations and guided tours, interaction with people was an important point. Pre-defined sentences were used for *greetings*, *welcome*, *self presentations* or *asking for free way*. A very simple web server was also developed to allow remote users to remotely operate the robot.

Blacky worked in long-term experiments and was tested in exhibition-like contests, where the exhibition organizers point of view was also taken into account. Lessons learnt from that experience conditioned the posterior research of our group. Further details about Blacky robot can be found in (Rodriguez-Losada et al., 2002).

### 1.3 Motivation

Our main research line is the development of autonomous navigation algorithms for mobile robots, especially focusing on the Simultaneous Localization and Map Building Problem (Rodriguez-Losada et al. 2006a; 2006b; 2007; Pedraza et al., 2007). This research line was partly motivated because in the setup of Blacky a time consuming manual installation and measurement of landmarks had to be done. Nevertheless, it is also true that the main goal of *Urbano* is more than having a platform to evaluate our navigation algorithms. We also wanted to have a platform that could serve to present our advances to the general public, to help us to get funding for our projects, and very importantly, to attract people and students to get involved in research programmes in our group. We think that we have succeeded in these goals: it has helped to present our research and publish our results, we have obtained increasing marks and funding from Spanish Government research programmes and the number of people in our group has also increased. It could be said that *Urbano* is the spirit of our group.

## 2. System overview

This section presents the description of *Urbano* hardware, both the commercial base but also our own developments: a mechatronic face and a robotic arm for gestures. Also, the general structure of software modules is presented. Later sections will present details about these software modules.

### 2.1 Urbano hardware

*Urbano* (Fig. 2) robot is a B21r platform from iRobot, equipped with a four wheeled synchro-drive locomotion system, a SICK LMS200 laser scanner mounted horizontally in the top used for navigation and SLAM, and a mechatronic face and a robotic arm used to express emotions as happiness, sadness, surprise or anger.

The robot is also equipped with two sonar rings and one infrared ring, which allows detecting obstacles at different heights that can be used for obstacle avoidance and safety. The platform has also two onboard PCs and one touch screen. These PCs are mainly

dedicated to access the hardware, low-level control of the base, interfacing the laser rangefinder, controlling the arm and face, and performing voice synthesis and recognition.



Fig. 2. *Urbano*, our interactive mobile robot

The hardware architecture is completed with two off-board PCs as shown in Figure 3. Communication with them is implemented via wireless Ethernet. One of the external computers is dedicated to the system kernel, which handles coordination of all modules. This system kernel can, however, run also in one onboard PC, leaving the external one just as a simple monitoring and supervision tool that could be even switched off.

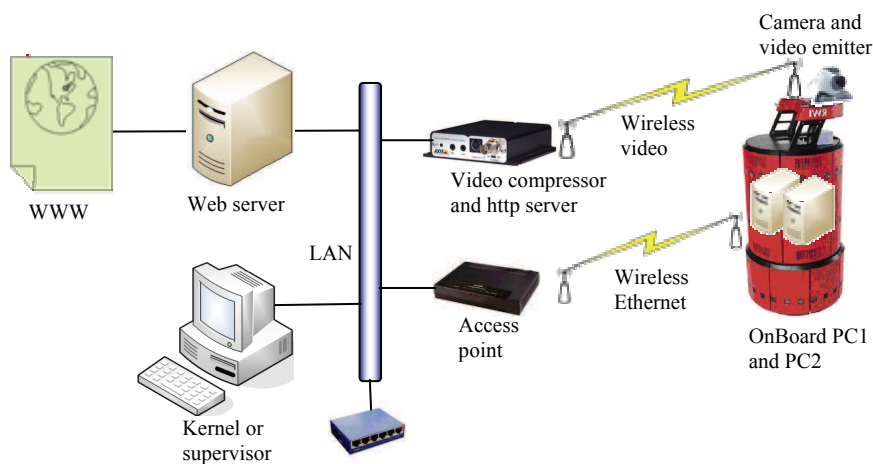


Fig. 3. System hardware architecture

The second PC is fully dedicated to the web server, which communicates with the kernel via Ethernet and TCP-IP protocol. The web server acts as interface between the robot and the

world, allowing remote users connection to operate the robot and visualize dynamic information coming from its sensors. We can also keep the web server at our laboratory, while the rest of the equipment is physically present at the exhibition site. The video stream is served through a dedicated http video compressor and server, which output is just redirected by the web server.

## 2.2 Robotic face

The robot was thought to interact with people. In fact, all the environments in which the navigation tests were done were plenty of people. In order to achieve a satisfactory interaction with the public, it was strictly necessary the design and implementation of a robotic face. People find in it an attraction point to look when talking to the robot. At the same time, the face allows the robot, in combination with the voice, to express basic emotions.

The first step was the face design, analyzing its ability to express emotions, and taking into account that the design should be simple enough to be built by ourselves. Other existing robots faces were also analyzed, like Kissmet (developed by MIT) which was too complex for our purposes. Albert's face, a robot from University of Freiburg was finally our referent. In an initial version we developed a face with 5 degrees of freedom, 2 to control the mouth, one for each eyebrow and another one for closing both eyes. The actuators were model servomotors S3003 from Futaba. In this version the eyes were completely static, the mouth could not be opened and both eyelids had to be moved together. Nevertheless, as shown in Figure 4, it was perfectly able to show basic emotions.

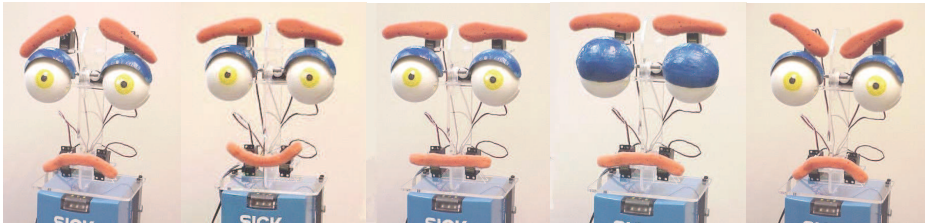


Fig. 4. *Urbano* face initial version with 5 degrees of freedom, showing happiness, sadness, being neutral, angry or asleep.

A simple board, the Mini SSC II, allows controlling the servos in an easy manner through a serial port, just by sending chains of three characters. This controller has low consumption and low dimensions.

This face is the one we have currently mounted in *Urbano*. Nevertheless, we have recently developed a new one (Fig. 5) with increased interaction capabilities. We have incremented the count of servos up to eleven, including four of them to control the mouth that can be opened and closed, simulating speech in a much more realistic way. Four servos are used to move both eyes left and right independently (cross-eyed possible) and independent eyelid closing for winking. Another servo moves both eyes together up and down. The eyebrows are controlled by two more servos. Jamara Mini-blue and Micro-blue servos are used in this version.

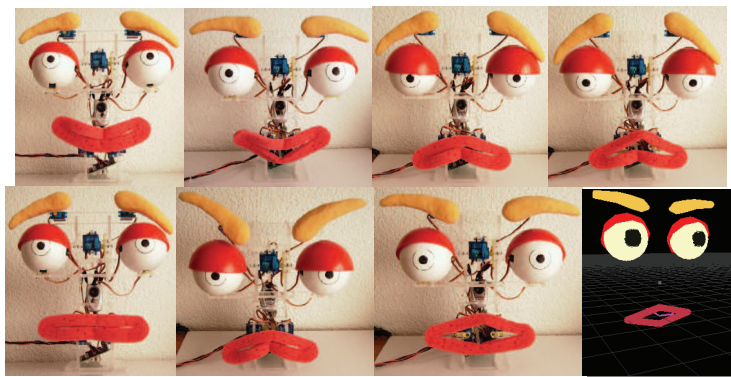


Fig. 5. *Urbano* face with 11 degrees of freedom, showing happiness, sadness, being neutral, angry, and moving the mouth.

### 2.3 Wired robotic arm

Since the beginning of *Urbano* development the need of a gestural interaction system between human and robot was considered. While showing the programmed tour, without gestural communication, the attention is easily missed due to the hieratic interaction between the robot and the visitors. With the inclusion of a robotic arm in the system, a more direct interaction with the environment is achieved. Moreover, it increases the ability to attract attention and helps to give emphasis and to include emotional aspects to the speech. Clearly, the use of sign language makes the interaction more natural, friendly and attractive. All this implies that the robotic arm must have a set of specific qualities and characteristics. Because it must reflect the common gestures in a speech, the structure, proportion and the dimensions of the robotic arm should be similar to the human arm. The arm movements should have similar dynamics, which requires that the movements should be stiff-less, quick and as natural as possible. As a consequence, the absolute accuracy and repeatability are not significant within a range, since the relative motions are more important in order to gesticulate than the absolute positions.

*Urbano* is conceived as a tour guide robot. Therefore it would be moving close to people and in a non structured environment. Safety issues have special relevance both for the humans and the robotic system. Due to its application, the system even has to allow contact with people without risk to them or to the robot hardware. However, as it has been specified before, the robot arm must move with agility and fast movements become more dangerous as the mass of the arm increases. As a consequence, a major requirement of the robot arm is that it has to be as light as possible. Reducing the inertia simplifies the actuator complexity and reduces the safety problems. Moreover, the actuation system should be somewhat reversible, so if the arm is moved manually it has to allow such movement or adopt a compliant behaviour.

The adopted solution in order to accomplish these requirements is to extract the drives from the arm and place them in the base of the robot, which is the equivalent to the shoulder blade. Placing the actuators in such way entails the problem of transmitting the mechanical power to the different joints and in particular to the elbow. Figure 6 shows the robot arm

kinematics. It has four degrees of freedom (dof), three on the shoulder articulation and the forth in the elbow.

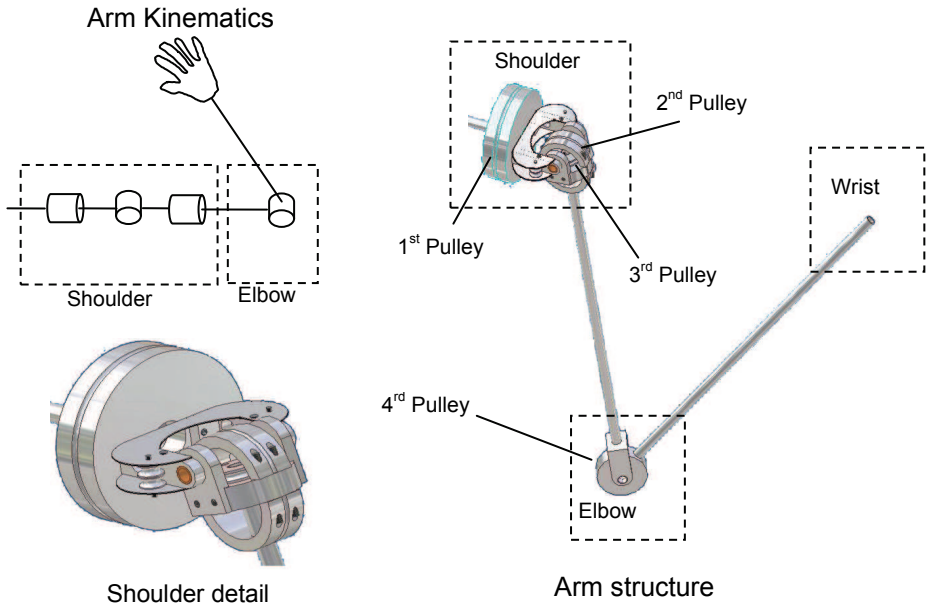


Fig. 6. Kinematics and Joint drive pulleys of the robotic arm.

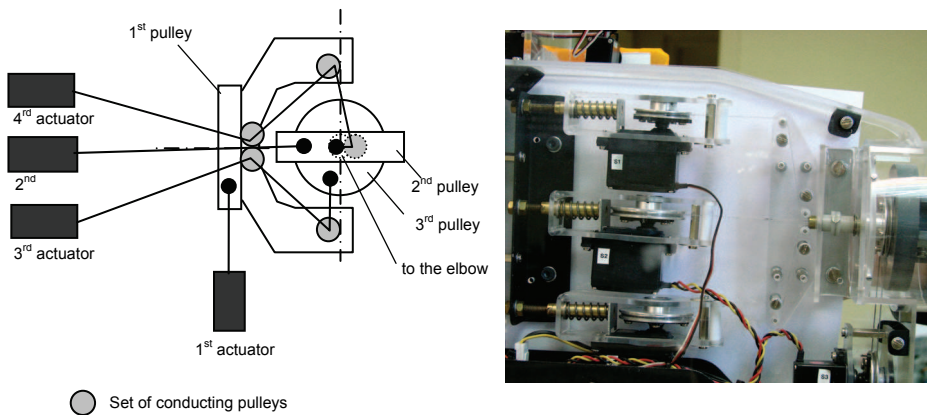


Fig. 7. Left) Schematic representation of the drive cables that are going through the shoulder. Right) Picture of the robot shoulder blade.

A cable based transmission system has been chosen as solution rather than gears. Using a gear based transmission system would be more expensive and complex from the mechanical and control point of view. In such systems a fine and complex control loop have to be used

in order to cancel the coupling among the different joints, due to the effect that a joint movement has on the subsequent articulations.

Each joint has a pulley where two wires are attached. Those cables turn the joint in opposite directions, and therefore, the length of wire that is wound must be the same that is released. Wires that run through the articulations have to be not affected by the joint movements. These can be accomplished conducting the cables through the axis of the previous joints. An added difficulty is the emergence of a friction that increases exponentially with the number of turns that the cable performs. Therefore, all the turns are made through a set of small polyester pulleys because of its low friction coefficient with nylon. Figure 7 (left) shows an scheme of the different drive cables that are going through the shoulder. The two drive cables for the elbow articulation have to go through the three previous joint axes. Therefore it is conducted by three different sets of pulleys as is represented in the figure.

Finally, the winding and releasing of the wires is done by four servo based drive units. The Figure 7 (right) shows these units placed on the shoulder blade of the robot. In order to keep the cables tensed each unit tighten them by an adjustable spring attached to the servo and the winding pulley. Given that the absolute accuracy is not a major requirement; the position feedback is done in the drive itself, not in the joint. Therefore there is no electronic components on the arm neither signal or power wires.

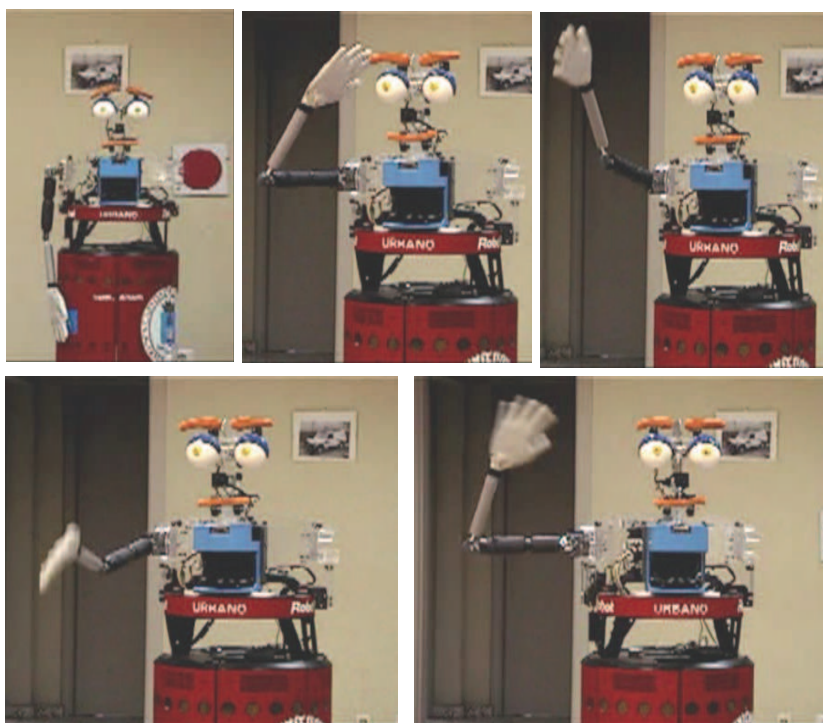


Fig. 8. Several video frames captured during a speech of *Urbano*.

The actuator units are controlled through a microcontroller based control board, that is linked through a serial RS232 connection to the onboard robot computer. This computer has



a server process that is responsible of the execution of the different commands received by a TCP/IP socket.

Figure 8 shows several video frames captured during a speech of *Urbano*. Predefined sequences of joint movements are able to express messages like this, goodbye, at your commands, everybody, etc.

**2.4 Software components overview**

The software is structured in several executable modules (Fig. 9) to allow a decoupled development by several teams of programmers, and they are connected via TCP/IP. Most of these executables are conceived as servers or service providers, as the face control, the arm control, the navigation system, voice synthesis and recognition, and the web server. The client-server paradigm is used, being the only client a central module that we call the *Urbano* kernel. This kernel is the responsible of managing the whole system, issuing requests to the services based on the input data defined by the exhibition database and the established robot behaviour, that is defined in a high-level programming language that will be described later.

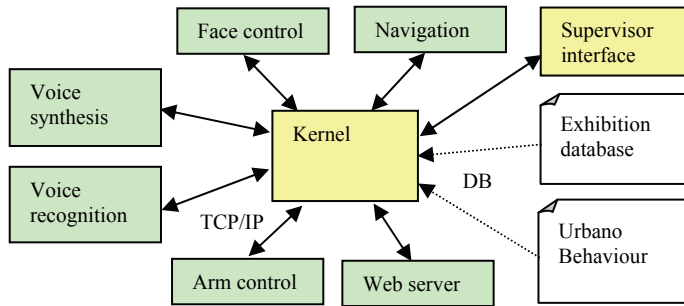


Fig 9. *Urbano* software control modules overview.

The supervisor interface acts as a client of the kernel, that reflects all necessary information to the user. Although this is the common use, the supervisor is also able to directly connect the server modules to check low level functionality.

**3. Feature based mapping and navigation**

We realized from our experiences with Blacky that automatic map building was required for an easy deployment of *Urbano* in new environments. The analysis of the exhibitions and the setup procedure indicated that a feature based approach could probably achieve better results and more robustness could be obtained both in the mapping procedure and the later localization in the built map. We noted that the environments were plenty of representative geometric entities, mainly straight walls, but they were also crowded because the setup procedure had to be performed while the exhibitions were open to the public. The most extended approach for feature based SLAM is the EKF algorithm, but this filter is difficult to apply when the features of the environment cannot be completely observed, e.g. when a wall is observed partially because of occlusions. Most of our recent research has been

focused in the feature based SLAM problem under an EKF approach. We developed our own version of the SPMAP (Castellanos et al., 1999) algorithm, which is probably the best existing solution to handle the problem of partial observations. Our algorithm (Rodriguez-Losada et al., 2006a) efficiently handles the edges information, which is extremely important when navigating in corridors.

We soon realized that the SLAM-EKF filter was quite optimistic due to the intrinsic inconsistency (Rodriguez-Losada et al., 2007) that arises due to EKF linearizations. We proposed the use of perfectly known shape constraints (parallelism, orthogonality, collinearity) between segments of a map to reduce the angular uncertainty of the robot that is the main source of linearizations (Rodriguez-Losada et al., 2006a; Rodriguez-Losada et al., 2007). With this solution, medium size maps with loops can be built in real time, which is more than enough for all the environments where *Urbano* has been deployed. Nevertheless, we also developed an algorithm based on the use of local maps (Rodriguez-Losada et al., 2007b), that allows multirobot mapping of large environments in real time.

The setup procedure is usually performed with a laptop connected to the robot base, used to manually drive the robot around the environment, while the SLAM-EKF algorithm runs, building the map in real time that is showed to the operator. Nevertheless, the system can also serve for remote exploration and autonomous return, in a fashion similar to (Newman et al., 2002) as we showed in (Rodriguez-Losada et al., 2007). Once the map is built, it is downloaded to the robot, so it can automatically start a simple pose tracking algorithm. This continuous localization or pose tracking is just a simplified version of the SLAM-EKF algorithm, with the map of the environment considered as perfectly known and static. Thus the estimation is only done over the robot position and orientation, resulting in a fast and robust algorithm.

Path planning in a feature based map is not recommended, as not every obstacle is represented in the map. Grid maps could be used, but the problem of obstacles at different heights still remains. Consider the existence of tables, stairs, fences, etc, which are basically undetectable by the robot perception system. The only way to achieve a safe navigation is to constraint the robot to certain areas supervised by the installer of the system. We used a graph based approach. While exploring, *Urbano* automatically builds a graph of the environment deploying nodes in the virtual map, that are connected by branches only when revisiting them is showed to be possible. Path planning is computed in this graph with an A-star heuristic, giving as a result a sequence of ordered nodes or waypoints to the final goal. The reactive controller moves the robot to the next waypoint with a simple regulator, but also avoiding obstacles with a deviation from the direction provided by the regulator. Safety is obtained by permitting only a limited distance to the actual branch. Usually, the graph computed by the robot is not enough to allow guided tours, so a graphical user interface allows the installer to add, delete, edit, move nodes and branches, as well as assigning tags to places that can be used to identify particular exhibits *Urbano* can show.

To allow the supervision of the map building procedure and the navigation performance, a GUI application has been developed. The SLAM and navigation kernel has been implemented in portable C++ for efficiency, and the interface has been developed (Figure 10) in a multidocument-view MFC application, using OpenGL for 3D rendering. This application has been proved to be of critical importance for an easy deployment of *Urbano*.

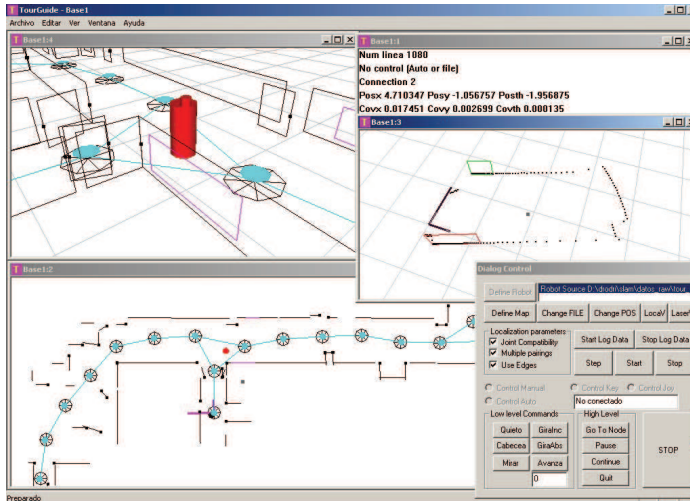


Fig 10. Map building and navigation GUI tool.

This navigation software has been also used in a different robot: the robotic smartwalker Guido of Haptica Ltd. (Dublin, Ireland), an assistive walker to support and guide the frail blinded elderly. The feature based mapping and navigation approach proved to be an improvement in Guido control, as shown in (Lacey & Rodriguez-Losada, 2008).

## 4. Interactive subsystem

### 4.1 Interaction capabilities

As described above, *Urbano* possesses several features that could be used for interacting with the people. If we conceive *Urbano* as a system, the interaction capabilities could be classified in inputs and outputs:

#### Outputs:

- The robotic arm is only able to perform gestures, but not force feedback is allowed. Thus the arm is not able to sense the environment or feel any contact. Although this would be a very interesting feature, it would also be quite complex and expensive.
- The face is able to show basic emotions, to move the mouth while speaking and to direct the eyes to any point.
- The robot base itself is an element that can interact with the people. It can move faster or slower, to look at the closest person, it can perform basic movements as steps, nodding, quick rotations, that can be used for complementing interaction.
- The voice synthesis is the most powerful and versatile output, being able to transmit any kind of information but also to change voice parameters (volume, speed, tone) and speak with different emotional pronunciation, but on the other hand it also requires a more complex handling.

#### Inputs:

- Voice recognition is the main input for interaction, despite its complexity. Both the difficulty of understanding the speaker in a noisy environment like an exhibition, and the management of textual information, makes impossible a general dialog manager. Nevertheless, it is still quit a powerful tool when the dialog is managed by the robot.

- The robot navigation system provides useful information, about close obstacles and people blocking its path, that can be easily used for initiating interaction.
- The face webcam is used for automatic face tracking (Figure 11) with the robot eyes, with a simple threshold of the image in the hue space, plus a geometric analysis of the binarized image.
- Some other information can be used as modifiers of the interaction with the user, as the battery level that can be associated to fatigue, or the time employed to perform a task that can produce stress to *Urbano*



Fig. 11. Face detection for tracking with the robotic eyes.

## 4.2 Proprietary voice synthesis and recognition

In order to provide *Urbano* with an appropriate human-robot interface, a speech synthesizer and a speech recognizer must be designed and developed. The proposed interface would allow a natural but reliable speech dialogue between visitors and the robotic guide. Although speech technology has progressively become a mature engineering area with several commercially available products, the development of robust applications in real-life ever-changing environments is still a topic of comprehensive research.

### 4.2.1 Speech recognition and understanding

Commercial speech recognition products are mainly oriented to classical speaker-dependent dictation products developed by Dragon Systems and IBM, telephone-based systems (the market is currently dominated by Nuance) or restricted-domain applications (Philips has developed several products mainly for hospitals). These systems come with limitations, as they cannot be used in open-access museums or trade-fairs without a significant reduction in their performance, because human spontaneity and limited linguistic coverage minimize the potential benefits of commercial products (Fernandez et al, 2006).

In addition to this, available systems do not provide automatic systems for speech understanding, but just speech recognition. The mobile robot needs procedures to extract concepts and values from the text that outputs the recogniser, being able to cope with recognition errors and ambiguities.

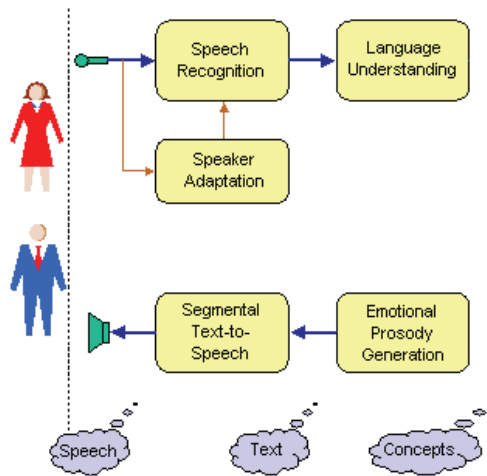


Fig. 12. Speech processing architecture in *Urbano*

Finally, commercial products generally do not provide a confidence measure on the result of the recognition process, a measure that allows a robust behavior on noisy working conditions. For example, when there are many children surrounding the robot.

Considering all these limitations, we have developed speech recognition software customized for use with a robot, as have we developed adapted modules for an air-traffic control domain or for controlling a HIFI system (Cordoba et al, 2006).

The standard speech recognition technique (Hidden Markov Models) is based on stochastic modeling of each phoneme in its context and trainable language model (bigram) that uses the probability of two words to be consecutively uttered in the specific application domain. We have trained our system with a 4000-speaker speech database to achieve speaker-independent models. In a 500-word command-and-control task, recogniser's word accuracy is greater than 95% (word accuracy takes into account speech recognition errors due to word substitutions, insertions and deletions).

Although any microphone can be used successfully, close-talk head-microphones are the best choice, due to immunity to ambient noise (which can be high in children-oriented museums, for example).

As for improving the performance of the recognition for certain special speakers, a speaker-adaptation module has been included. This module significantly improves the general models trained with 4000 speakers). Error reduction can be as high as 20%, especially for female speakers.

As speech recognition is only the first component of the speech processing, we have also developed an automatic speech understanding module. In order to adapt the system to a new exhibition or trade fair, we must provide the system with a set of samples that should be recognized by the robot, and the set of concepts and values involved in each sentence. The system automatically learns a set of understanding by induction. The rules that are learnt can convert the recognized speech into the suitable sequence of concepts and values, without the need of a human expert. Nevertheless, if the set of examples is reduced, new rules can be manually added.

### 4.2.2 Emotional speech synthesis

While in recent years many speech synthesizers have managed to achieve a high degree of intelligibility, one important problem remains, which is the inability of simulating the variability in human speech conveyed by factors such as the emotional state of the speaker.

The approach of this work has been based on formant synthesis, including four primary emotions, namely happiness, sadness, anger and surprise as well as a neutral state. Although this approach produces less natural speech when compared to other approaches such as concatenate synthesis, it provides a high degree of flexibility and control over acoustic parameters.

To improve the results of previous approaches, we have optimized the prosodic models that mimic the rhythm and intonation of the reference professional speaker we have recorded. Each emotion is prosodically modeled as a deviation from the neutral way of speaking. To simulate sadness we have included an artificial tremor that, although not used by the professional actor, has significantly increased the identifiability of this emotional synthetic speech.

Our actor has simulated cold anger (instead of hot anger), which is a very-controlled but menacing emotion. To simulate this kind of anger, he created a special noise during the articulation of most of the sounds, without modifying the quite neutral prosody. This non-prosodic anger is very difficult to be completely simulated on formant synthesis. This time, the significant improvement was obtained by combining an artificial articulation noise with an intensity pattern that progressively simulates hot anger.

Finally, to improve happiness, the most difficult emotion, we have increased the amount of high frequency in synthetic speech, to provide it with richer sound that is easily associated with a happy state.

The simulation of emotions in synthetic speech was tested by a group of 24 non-trained listeners. The confusion matrix obtained for the 25 sentences that composed the test is showed in next table.

Simulated Emotion	Identified emotion (%)					
	Happiness	Cold Anger	Surprise	Sadness	Neutral	Other
Happiness	<b>53.9</b>	9.6	20.9	0.0	7.8	7.8
Cold Anger	7.0	<b>70.4</b>	14.8	2.6	3.5	1.7
Surprise	17.4	2.6	<b>79.1</b>	0.0	0.0	0.9
Sadness	0.0	1.7	0.0	<b>87.0</b>	10.4	0.9
Neutral	1.7	3.5	2.6	7.8	<b>83.5</b>	0.9

Table 1. Confusion matrix from emotion identification experiments on speech synthesis

We can observe that all the emotions present a recognition level above 50%, and for all the emotions with the exception of happiness, this level exceeds 70%. The mean identification rate in the new perceptual test is 75%, in this semantically-neutral short-sentence emotion identification experiment. When compared to previous formant-based results on the Spanish work package in VAESS project, (Montero et al, 2002) the improvement ranges from 65% for anger, 42% for neutral, 15% for happiness, to just 5% for sadness.

These results are even better (>65%) for the last 10 sentences that composed the test, in spite of the fact that listeners did not receive information about the identification success or failure they were getting

### 4.3 Emotional manager

Many investigations in the emotional model area have been done and many others are currently under way. It is quite a new field and it involves many different sciences, for that reason it is not common to find fix structures for studying or developing artificial emotional models. One of the most significant studies is (Picard, 1997). From a pure scientific point of view, emotional models are studied in psychology, neuroscience, biology, etc. Humaine Network of Excellence (<http://emotion-research.net>) aims to create an investigation community to study emotions in the frame of human-robot interaction.

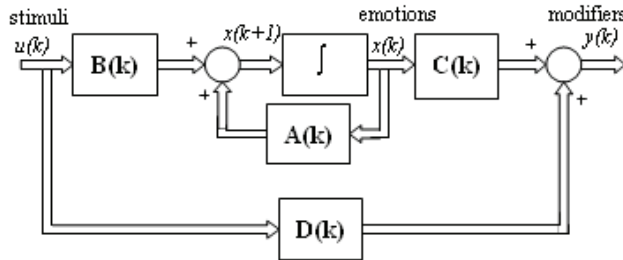


Fig. 13. Emotional state model

In order to reach a nearer approximation to human emotional system, the *Urbano* model makes use of dynamic variables to represent internal emotional state. The model follows the classic diagram showed in Fig. 13, being the system stimuli  $u(k)$  considered as inputs variables, emotions  $x(k)$  as state variables and task modifiers  $y(k)$  as output variables. In the following paragraphs the concepts used to build the emotional model are introduced more accurately.

Trying to define an emotional state in a human being, an emotion and its magnitude are used. For example, the statement “I am very happy” includes qualitative information, the emotion “happy”, and quantitative information that is expressed with terms that give an idea about the intensity of the emotion “very”.

Based on that, the *emotional state* at the time  $k$  is defined as the set of considered emotions with their intensity levels. Intensity levels of each emotion change continually, giving dynamics to emotional state. Emotional state tends naturally to a nominal emotional state where a balance of emotion intensities exists. An emotion is an internal variable.

A *system stimulus* is any event that has an influence in the system producing an emotional state change. There are many events that may stimulate the system, the only limitation is the system ability to sense, i.e. sensors, cameras, etc. Robotic stimuli may be internal or external. An example of internal stimuli is the life or energy the robot has, usually considered as the battery state.

*Urbano* has scheduled tasks; such schedule can be modified because of instantaneous emotional state. All these changes are considered as *system task modifiers*. An example of scheduled task is the tour in a museum, which a guide robot has to direct. Modifiers for this task could be the tour tempo, information to give, jokes used to build a better connection with public, etc.

Following the classic state variable model, four matrices have to be defined: A-matrix *emotional dynamic matrix* represents the model dynamic, the influence of each emotion over

itself and over the other emotions. B-matrix is the *sensitivity matrix*. C-matrix has the information of how emotional state influences modifiers. Let us call this matrix the *emotional behavior matrix*. D-matrix is the *direct action matrix*.

Due to the difficulty of finding an analytic calculation for the matrices coefficients, a set of fuzzy rules is used to obtain each coefficient. The matrices coefficients are function of time  $k$ , giving dynamics to the system. Because of that coefficients are calculated for each time  $k$ . To define fuzzy rules is a simple task; the information contained in the rules can be obtained from experts in emotions. The use of fuzzy knowledge bases opens the opportunity to a future automatic adjustment, e.g. genetic algorithms.

## 5. Web based remote visit

One of the project goals was the development of a Web server to allow users to visit remotely an exhibition, navigating through the robot movement and watching through its sensors. The user can be a normal citizen that enjoys connecting from his home, or a business man that connects from his office. This allows saving the displacement costs derived from travelling physically to the exhibition site, especially when the visitor lives or works in another city or country. Three kinds of users are allowed:

- The standard visitor, which can navigate through the web page accessing general information and watching the behaviour of the robot, or ask for an account.
- Privileged visitor, which can operate the robot and interact with the remote site, as well as with other connected users.
- Administrator, which can manage users, creating new accounts and assigning access privileges.

A privileged user can:

- Set a destination goal for the robot (high level command). The navigation system works in autonomous mode.
- Chat with other users.
- Command the robot sending low level commands (move forward or backward, turn). A security system avoids the robot to crash.
- To receive dynamic information of the surroundings of the robot.
- Visualize the robot environment through its camera.
- To receive the audio signal present at the remote site.
- To write down sentences to be synthesized by the robot.
- To select emotions to be expressed by the robot face.

The web server was developed using Jakarta Apache Tomcat 4.0, which includes Java support, over a Linux operating system (Debian 3.0 release1). The programming tools used were those included in Java 2 Platform, Enterprise Edition, J2EE (Java Server Pages -JSP-, JavaBeans, JavaXML), server and applets applications, and every program was written in standard Java 2. The Web pages format is standard HTML 4.0. Figure 14 shows the typical frames displayed during normal operation (map, camera, chat and control windows).

All data is stored in a MySQL data base. Information exchanged with the database is carried out using SQL (Structured Query Language) through queries to a data base server (MySQL Server) resident in the same PC. The development application was the programming environment supplied by Sun Microsystems, SunOne Studio 4.1 Community Edition.

The web server was deeply tested at INDUMATICA 2004 fair celebrated at UPM. The server worked for 3 days, a total of 16 hours. 63 users registered, being 18 professors, 31 students



and 14 people form outside of the university. The web Server was also successfully proven at the Science Museum Principe Felipe of Valencia, and allows carrying out remote tours to our laboratory at UPM.

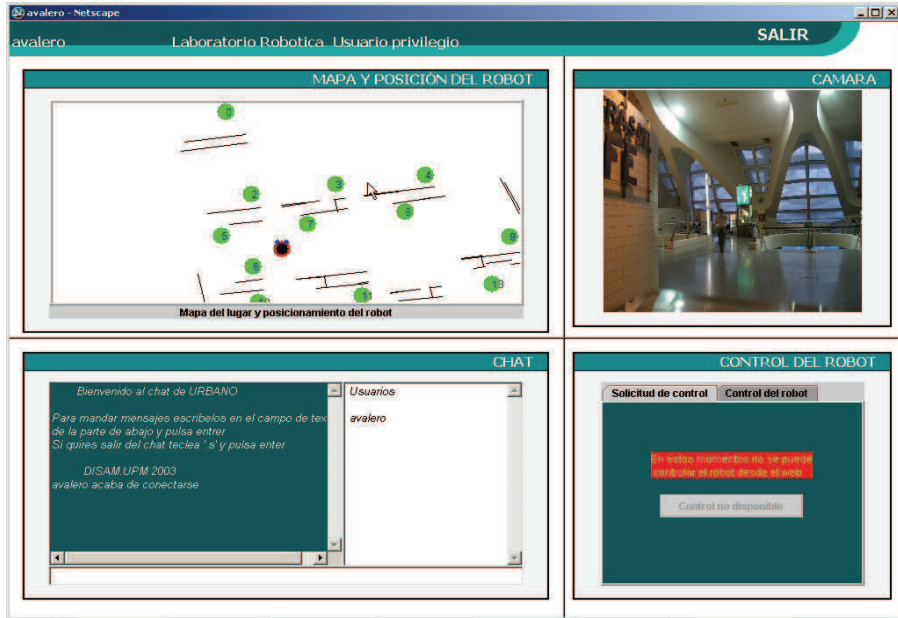


Fig. 14. *Urbano* Web based remote visit

## 6. Integration of components: Multitask Kernel

From the point of view of *Urbano's* software components, it is an agent based architecture. A specific CORBA based mechanism is used as integration glue. Every agent is a server and there is only one client, the *Kernel* module. Each computer has a *Monitor* program that interacts with the Operating system to start, suspend or kill the applications assigned to this machine.

Watchdog supervision mechanism are used to detect blocks in every client and if it is necessary to restart it. Some agents need to save a safe state in order to recovery the whole functionality (robot's recent position).

There are different kinds of information involved in *Urbano*:

- Configuration. All necessary configuration data (IP address, file names, etc.)
- Working data. Each agent can use specific information usually data files (sequence of movements for the "Hello" action in Arm agent)
- General information. About social, humoristic, sportive information that *Urbano* uses to interact with the public
- Corpus. About the specific domain which *Urbano* works (Museum or fair contents).

A relational database was implemented to support general and corpus information, and specific files for working and configuration data. There is not redundant or shared information. The agents and their function are described in Table 2.

Agent	Function	Computer	Activity
Kernel	Task scheduler, knowledge	OnBoardPC2 (win)	Client, Server
Speech	Voice synthesis	OnBoardPC2 (win)	Server
Listen	Voice recognition	OnBoardPC2 (win)	Server
Face	Face expression control	OnBoardPC1 (linux)	Server
Arm	Arm movements control	OnBoardPC1 (linux)	Server
Navigation	Base movements control	OnBoardPC1 (linux)	Server
Emotional	Emotional model control	OnBoardPC2 (win)	Server
Supervisor	Monitoring of kernel and modules	External PC1 (win)	Client of kernel
Web server	Serve web pages	External PC2 (linux)	Server Http server

Table 2. Agents and functions

Some other programs have been developed for different needs. *Mapper* was designed to elaborate and managing maps and graphs for path planning. *UDE* the *Urbano* development environment is a complex program designed to help the end user in the maintenance and task development, also is a supervisor program of the whole architecture. Figure 15 shows the main window of this program.

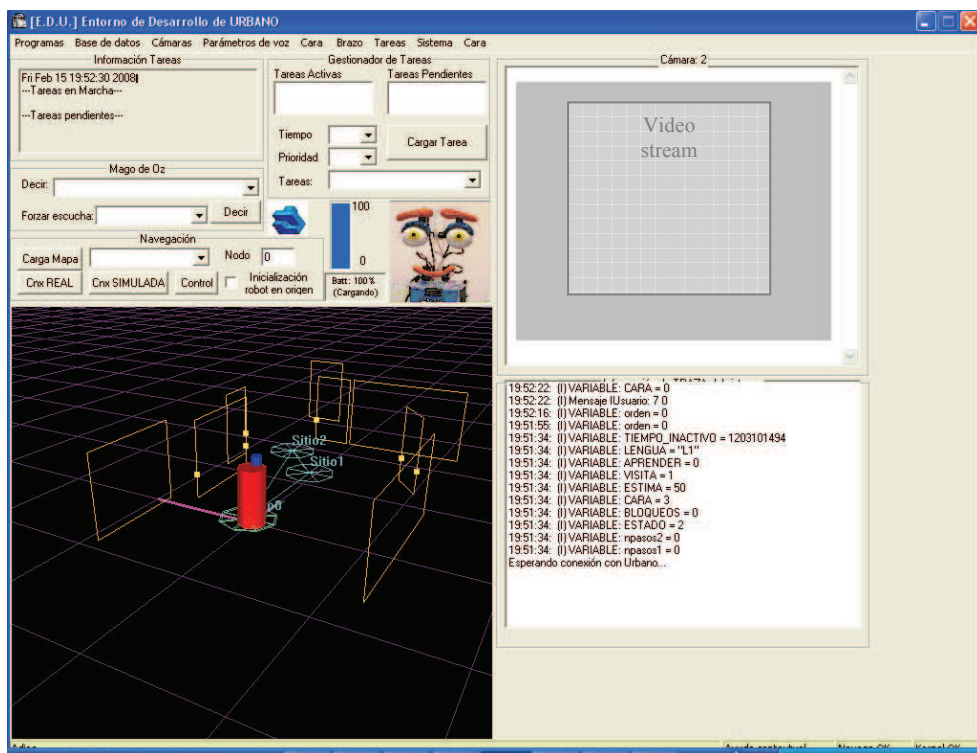


Fig. 15. Urbano Development Environment

The Kernel agent is a scheduler that executes *Urbano* tasks. Each task has a starting time and a priority. High priority tasks interrupt lower priority tasks. Tasks are coded by the user in a high-level programming language designed for this purpose. The tasks are compiled with yacc-lex technologies to avoid errors and to simplify the execution by the Kernel.

### 6.1 High-level programming

High level programming language designed is C-like. Variables can be numerical or string and the first assign defines the type. Expressions and execution control sentences are available in the same syntax that C language.

There are an important set of functions related with database access, string operations, global variables, system, etc. There are also functions to control the robot. The following table 3 shows some of these functions:

Function	Description
<code>listen</code>	Waits for a specific sentence from de voice recognition module
<code>listendb</code>	Waits for a sentence defined in Database
<code>say</code>	Synthesizes a sentence
<code>saydb</code>	Synthesizes a random sentence of a category from the database.
<code>face</code>	Shows a specific expression in the face
<code>arm</code>	Does a set of arm movements that was defined as a expression.
<code>play</code>	Shows a multimedia movie in the Touch-Window
<code>image</code>	Shows an image in the Touch-Window
<code>buttons</code>	Returns the identification of the selected button in the Touch panel
<code>feeling</code>	Does an evaluation of robot emotions
<code>go</code>	Goes to a specific point
<code>where</code>	Returns where the robot is
<code>turn</code>	Turns some degrees
<code>isblock</code>	Returns true if the robot is blocked

Table 3. Control Functions of *Urbano* programming language

The following text shows an example of task. The robot is walking and hellowing around the available 14 places in the map. Task starts with an order to go to the next place. While is moving call to another task to verify if the robot is blocked in his path by objects or people and say a random hellowing phrase from the database and wait 30 seconds. When the robot is in the next place, put itself in the agenda as a new task with 5 seconds of delay to start and a priority of 20.

```
// Walking !
    destination=where()+1;
    if(destination ==14) destination = 0; endif
    go(destination);
    while (where() != destination)
        jump("blocks");
        saydb("Hello");
        sleep(30);
    endwhile
task("walking",5,20);
end
```

In this new task example, the robot wait for a question (recorded in the database), then the *listen* function returns a keyword and a SQL query is performed to obtain from the 'explains' table all records with this keyword. In every record there are a Text, a voice type, an arm movement and a face expression that are used to give the answer.

```
// Questions
say("What is your question?");
Theme=listendb();
pTable=dbsql(format("SELECT * FROM EXPLAINS WHERE KEYWORD = '%s'
ORDER BY ORDEN", Theme));
if (pTable>=0)
    ndatos=dbgetcount(pTable);
    while (ndatos>0)
        arm(dbgetint(pTable,"ARM"));
        setspeakingvoicetype(dbgetint(pTable,"VOICE"));
        face(dbgetint(pTable,"FACE"));
        say(dbgetstr(pTable,"TEXT"));
        dbnext(pTable);
        ndatos=ndatos-1;
    endwhile
    dbclose(pTable);
    gestobrazo("POSICION_CERO");
else
    say("I don't now anything about this theme!");
endif
end
```

## 6.2 Managing visits

*Urbano* database has an inventory of objects. Each object is included in several categories, for example a Picasso's picture is a picture, modern art, cubism style, big size, etc. Each object is in a place in the map and the order of visit is important in order to avoid comings and goings. About each object there are different kinds of information: general description, specific for expert, specific for child, components, history, details, anecdotes, etc.

*Urbano* as tour guide robot must guide to a people group in a museum or fair in a visit. For *Urbano* a visit is defined as a set of categories to explain in limited time for some kind of visitors defined by some topics: Expert, Normal, Child, etc. Some SQL queries to database select the objects and the information about each object to be explained. If there isn't enough time for the exposition of all selected objects, a prune process is executed to reduce the number of explanations of each object (a priority value). This work is previous to the visit and can produce a test of visit, the robot makes the visit and controls the moving time and the explanation time in each object.

During the real visit timing can vary depending on questions or moving time (visitors blocks *Urbano*) if time lacks a prune process is used. Free time can be used by *Urbano* to tell jokes or recent social news.

## 7. Urbano successful deployments

*Urbano* robot has been successfully deployed in several environments, and has operated as tour guide in many occasions:

- Lab Tour: guided visit to our laboratory

- Indumatica 2004 (ETSII, Madrid, Spain): industrial trade fair
- Indumatica 2005(ETSII, Madrid, Spain): industrial trade fair
- Fitur 2006 (IFEMA, Madrid, Spain): international fair of tourism
- Principe Felipe Museum (CACSA, Valencia, Spain). Science museum.
- Demonstration at UPM.

A demonstration was performed at our university that started with a teleoperated real time exploration and mapping at rush hour with the environment crowded with students. The installer used the GUI tool to teleoperate the robot with the reference of the map graphical render, and the assistance of the robot reactive control for safety and automatic graph building, path planning and execution for convenience and comfort. In this way, the installer teleoperated the robot while exploring, but *Urbano* could go to any previous explored area fully autonomously, releasing the user from direct control most of the time. The duration of the experiment was 22'15" with a travelled distance of 134 meters. With an experiment duration similar to the "Explore and return" experiment (Newman et al., 2002) the explored and mapped (in real time) environment is much bigger (Figure 16).

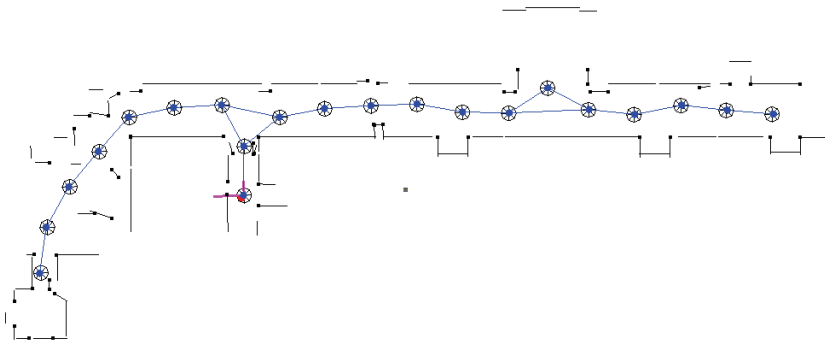


Fig. 16. Map of UPM built in an "Explore and return" experiment

*Urbano* robot was also deployed in the Indumatica trade fair (Figure 17) in two occasions 2004 and 2005. In both occasions it had to be installed while the fair was open to the public, and the map building was accomplished with the exhibition plenty of people. Next figure shows the map provided by the organizers; as it can be seen it is useless for navigation, as it does not resemble the actual environment. The built map accurately represents the features of the environment.



Fig. 17. Indumatica 2004 trade fair. Left) Map provided by organizers. Center) Actual environment. Right) Partial view of the built map.

The map of the environment was built in real time while manually driving *Urbano* in a 102 meters long trajectory in less than five minutes. The complete map and navigation graph is shown in Figure 18, as well as *Urbano* guiding two visitors around the fair.

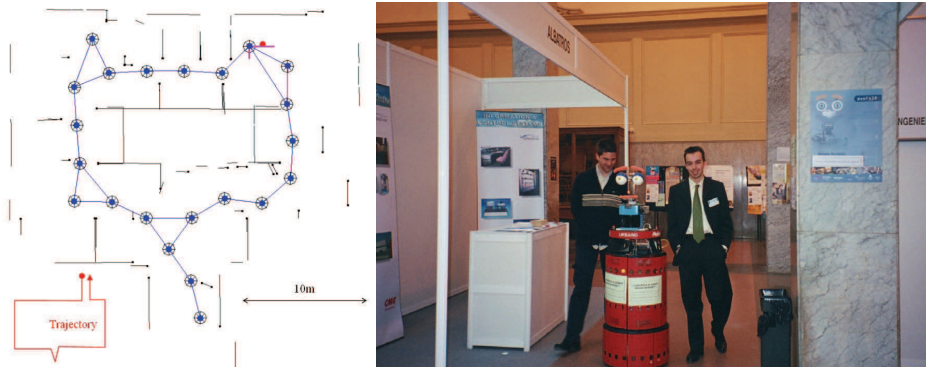


Fig. 18. Indumatica 2004 trade fair. Left) Map built by *Urbano* in real time. Right) *Urbano* guiding two visitors.

The *Urbano* project has been supervised by the “Principe Felipe” museum at the City of Science and Arts of Valencia (CACSA), one of the biggest museums in Spain, as partner and potential end user of *Urbano*. A demonstration of the system deployment was performed (Figure 19), as well as the functionality of *Urbano* as a tour guide. The map of the exhibition was correctly built in real time along a 130 meters long trajectory in approximately 16 minutes.

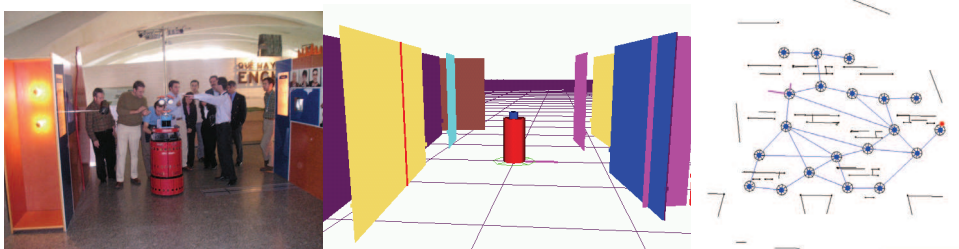


Fig. 19. Map building at Principe Felipe Museum. Left) Manually operating the robot. Center) Real time map building as seen by the installer. Right) Resulting map and navigation graph.

## 8. Conclusions and future work

The *Urbano* service robot system has been presented in this chapter, with an overview of both its hardware and control software. The hardware used for interaction (robotic face and arm), that has been specifically designed and built for *Urbano* following performance and cost criteria, has been showed to successfully accomplish its task. All the control, navigation, interaction (including speech) and management software has been developed from scratch according to our research lines. These developments have served to increase our scientific

publication records, but have also resulted in the attainment of a quite mature service robot system that has been successfully deployed and tested in several occasions in different scenarios. Moreover, due to its success, we have been requested many times to rent *Urbano* for several days in exhibitions by several institutions and private companies. The only reason we couldn't go on with this renting, was the lack of support in the University for this purpose, as our University is public and a non-profit organization. We are currently considering forming a spin-off to continue with *Urbano* in a more commercial line.

We are currently working in 3D data acquisition, modelling, mapping and navigation in order to achieve a much more robust system (able to detect stairs, obstacles at different heights), that wouldn't require any human supervision (navigation graph editing) for a more automated setup. In fact our goal (Robonauta project, see Acknowledgement) is the fully automated deployment of *Urbano* by showing it the environment, guiding it with natural language, just as it would be done with a new human guide in a museum staff. The interaction capabilities of *Urbano* are also being expanded, implementing some people tracking and following behaviours, as well as an improved image processing system.

The software distributed architecture will also be improved by the standarization of modules interfaces using XML technologies and the (Web Services Definition Language) WSDL specification. In this way, the modules will not require to have the interfaces hard-wired, and more flexibility and simplicity will be allowed for a more fast and error-free development. Also, the programming language will be substituted by some standard as the State Chart XML (SCXML), that could result in a more powerful and simpler to manage tool that took full advantage of the new architecture.

## 9. Acknowledgment

The *Urbano* project has been the result of the work of many people, whose contributions we gratefully acknowledge: Agustin Jimenez and Jose M. Pardo for project management and supervision, Alberto Valero for web development, Andres Feito and Marcos Doblado for face design and building, Enrique Lillo for his work in the wired arm, Javier Diez for programming the *Urbano* high-level programming language and kernel, Jaime Gomez and Sergio Alvarez for improvements in the kernel, and all of DISAM and IEL (both at UPM) staff for their support.

This work is funded by the Spanish Ministry of Science and Technology (URBANO: DPI2001-3652C0201, ROBINT: DPI-2004-07907-C02, Robonauta: DPI2007-66846-C02-01) and EU 5<sup>th</sup> R&D Framework Program (WebFAIR: IST-2000-29456), and supervised by CACSA whose kindness we gratefully acknowledge.

## 10. References

- Burgard W., Cremers A.B., Fox D., Hähnel D., Lakemeyer G., Schulz D., Steiner W., Thrun S. (1999) Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*. Vol. 1-2 N. 114. pp. 3-55.
- Thrun S., Bennewitz M., Burgard W., Cremers A.B., Dellaert F., Fox D., Hahnel D., Rosenberg C., Roy N., Schulte J., Schulz D. (1999). MINERVA: A Second-Generation Museum Tour-Guide Robot. *IEEE International Conference on Robotics and Automation*. Vol.3, pp. 1999-2005.

- Nourbakhsh I., Bobenage J., Grange S., Lutz R., Meyer R., and Soto A. (1999). An Affective Mobile Educator with a Full-time Job. *Artificial Intelligence*, Vol. 114, No. 1 - 2, pp. 95-124.
- Montemerlo M., Pineau J., Roy N., Thrun S., and Verma, V., (2002). Experiences with a Mobile Robotic Guide for the Elderly. *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada.
- Rodriguez-Losada D., Matia F., Galan R., Jimenez A. (2002). Blacky, an interactive mobile robot at a trade fair. *IEEE International Conference on Robotics and Automation*. Vol. 4. Washington DC, USA. pp. 3930-3935.
- Rodriguez-Losada D., Matia F., and Galan R. (2006a) Building geometric feature based maps for indoor service robots. *Robotics and Autonomous Systems*, vol. 54, pp. 546-558, 2006.
- Rodriguez-Losada D., Matia F., Jimenez A., Galan R. (2006b). Local map fusion for real-time indoor simultaneous localization and mapping. *Journal of Field Robotics*. Wiley Interscience. Vol 23, Issue 5, p 291-309, May 2006
- Rodriguez-Losada D., Matia F., Pedraza L., Jimenez A., Galan R. (2007). Consistency of SLAM-EKF Algorithms for Indoor Environments. *Journal of Intelligent and Robotic Systems*. Springer. ISSN 0921-0296, Vol. 50, N<sup>o</sup>. 4, 2007, pags. 375-397.
- Pedraza L., Dissanayake G., Valls Miró J., Rodriguez-Losada D., and Matia F. (2007). BS-SLAM: Shaping the world. In *Proc. Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.
- Castellanos J.A., Montiel J.M.M., Neira J., Tardos J.D. (1999). The SPmap: A Probabilistic Framework for Simultaneous Localization and Map Building. *IEEE Transactions on Robotics and Automation*. Vol. 15 N. 5. pp. 948-953.
- Newman P., Leonard J., Tardos J.D., Neira J. (2002) Explore and Return: Experimental Validation of Real-Time Concurrent Mapping and Localization. *IEEE International Conference on Robotics and Automation*. Washington DC, USA. pp. 1802-1809
- Lacey G. and Rodriguez-Losada D. (2008) The evolution of Guido: a smart walker for the blind. Accepted for publication in *IEEE Robotics and Automation Magazine*. To appear in 2008.
- Fernández, F.; Ferreiros, J.; Pardo, J.M. ; Sama, V.; Córdoba, R. de ; Macías-Guarasa, J.; Montero, J.M.; San Segundo, R.; D'Haro, L.F.; Santamaría, M. & González G. (2006). Automatic understanding of ATC speech. *IEEE Aerospace and Electronic Systems Magazine*, Vol. 21, No 9, pp. 12-17, ISSN: 0885-8985
- Córdoba, R. de ; Ferreiros, J.; San Segundo, R.; Macías-Guarasa, J.; Montero, J.M.; Fernández, F.; D'Haro, L.F. & Pardo, J.M. (2006). Air traffic control speech recognition system cross-task & speaker adaptation. *IEEE Aerospace and Electronic Systems Magazine*, Vol. 12, No 9, pp. 12-17, ISSN: 0885-8985
- Montero, J.M.; Gutiérrez-Arriola, J.; Córdoba, R.; Enríquez, E. & Pardo, J.M. (2002). The role of pitch and tempo in Spanish emotional speech: towards concatenative synthesis. In: *Improvements in speech synthesis*, Eric Keller y Gerard Bailey, A. Monahan, J. Terken, M. Huckvale (Ed.) pp. 246-251, John Wiley & Sons, Ltd.
- Picard R. W., (1997). *Affective Computing*, The MIT Press, Massachusetts, USA. ISBN:0-262-16170-2





## **Advances in Service Robotics**

Edited by Ho Seok Ahn

ISBN 978-953-7619-02-2

Hard cover, 342 pages

**Publisher** InTech

**Published online** 01, July, 2008

**Published in print edition** July, 2008

This book consists of 18 chapters about current research results of service robots. Topics covered include various kinds of service robots, development environments, architectures of service robots, Human-Robot Interaction, networks of service robots and basic researches such as SLAM, sensor network, etc. This book has some examples of the research activities on Service Robotics going on around the globe, but many chapters in this book concern advanced research on this area and cover interesting topics. Therefore I hope that all who read this book will find lots of helpful information and be interested in Service Robotics. I am really appreciative of all authors who have invested a great deal of time to write such interesting and high quality chapters.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Diego Rodriguez-Losada, Fernando Matia, Ramon Galan, Miguel Hernando, Juan Manuel Montero and Juan Manuel Lucas (2008). Urbano, an Interactive Mobile Tour-Guide Robot, *Advances in Service Robotics*, Ho Seok Ahn (Ed.), ISBN: 978-953-7619-02-2, InTech, Available from:

[http://www.intechopen.com/books/advances\\_in\\_service\\_robotics/urbano\\_\\_an\\_interactive\\_mobile\\_tour-guide\\_robot](http://www.intechopen.com/books/advances_in_service_robotics/urbano__an_interactive_mobile_tour-guide_robot)

# **INTECH**

open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.