Building a Fully Autonomous Tour Guide Robot: Where Academic Research Meets Industry

N. Tomatis[†‡], R. Philippsen[‡], B. Jensen[‡], K. O. Arras[‡], G. Terrien[†], R. Piguet[†], R. Siegwart[‡]

[†]BlueBotics SA
PSE-C
CH-1015 Lausanne
n.tomatis@ieee.org

[‡]Autonomous Systems Lab, EPFL
Swiss Federal Institute of Technology Lausanne
CH-1015 Lausanne
r.siegwart@ieee.org

Abstract

This paper presents the effort that has been undertaken in designing and building both hardware and software for a fully autonomous navigating vehicle aimed for a tour guide application. The challenge for such a project is to combine industrial high quality production for the mobile platforms with techniques for mobile robot navigation and interaction which are currently best available in academic research. For this the experience and technology of the Autonomous Systems Lab at EPFL has been extended with the industry-driven knowledge of BlueBotics SA, a young enterprise which aims to help taking robots out of the university offices and putting them into real-world applications.

The goal of the project is to maximize the autonomy and interactivity of the mobile platform while ensuring high robustness, reliability and performance. The result, called RoboX, is an interactive moving machine which can operate in human environments and interact by seeing humans, talking to and looking at them, showing icons and asking them to answer its questions.

Keywords: tour guide robot, navigation, interaction, product

1. Introduction

A tour guide robot has to be able to move around autonomously in the environment. It has to acquire the attention of the visitors and interact with them efficiently in order to fulfill its main goal: give the visitors the pre-defined tour. The environment is known and accessible, but a general approach requiring no environmental changes is better suited for a commercial product. For the same reason a fully-autonomous and self-contained robot is preferable. Furthermore such a machine is required to have a long live cycle and a high mean time between failure (MTBF), which minimizes the need of human supervision.

2. Related Work

The tour-guide robot task can be decoupled in two separate issues: navigation and interaction.

Navigation: A limited number of researchers have demonstrated autonomous navigation in exhibitions or museums [5], [12], [15], [9] and [17]. Furthermore, most of these systems have still some limitations in their navigation approaches. For instance *Rhino* [5] and *Minerva* [15] have shown their strengths in museums for one week, 19 kilometers and two weeks, 44 kilometers respectively. However, their navigation has two major drawbacks: it relies on off-board resources, and due to the use of raw range data for localization and mapping it is sensible to environmental dynamics. *Sage* [12], *Chips*, *Sweetlips*, *Joe* and *Adam* [17], use a completely different approach for permanent installations in museums: the environment is changed by adding artificial landmarks to localize the robot. This approach performed well, as shown with a total of more than half a year of operation and 323 kilometers for *Sage* [12] and a total of more than 3 years and 600 kilometers for *Chips*, *Sweetlips*, *Joe* and *Adam* [17]. However their movements, but for *Adam*, are limited to a predefined set of unidirectional safe routes in order to simplify both localization and path-planning. Another robot permanent installation which is operating since March 2000, is presented in [9]. Three self-contained mobile robots navigate in a restricted and very well structured area. Localization uses segment features and a heuristic scheme for matching and pose estimation. Another exhibition where *Pygmalion*, a fully autonomous self-contained robot was accessible on the web during one week [1] has shown its positive characteristics but, due to the unimodal characteristic of the used Extended Kalman Filter, the robot can still lose track if unmodeled events take place.

Interaction: Human-centered and social interactive robotics is a comparatively young field in mobile robotic research. However, several experiences where untrained people and robots meet are available. The analysis of the first public space experience with *Rhino* [5] underlines the importance to improve human-robot interfaces in order to ease the acceptance of robots by the visitors. In [15] *Minerva* attracted visitors and gave tours in a museum. It was equipped with a face and used an emotional state machine with four states to improve interaction. The *Mobot Museum Robot Series* [12] and [17] focused on the interaction. Robustness and reliability was identified as an important part of a public robot. The permanent installation at the *Deutsches Museum für Kommunikation* in Berlin [9], uses three robots which have the task to welcome visitors, offer them exhibition-related information and to entertain them.

The system presented here is designed to offer enhanced interactivity and autonomous navigation with a completely self-contained robot and without requiring changes of the environment. Furthermore it is intended to work permanently with minimal supervision.

3. Product Design

The highly dynamic environment and the requested visitor experience expected for such a product impose various constraints on the robot design and control. This leads to a specification of the mobile platform that can be summarized as follows:

- Highly reliable and fully autonomous navigation in unmodified human-environments crowded with hundreds of humans.
- Bidirectional multi-modal interaction based on speech (English, German, French and Italian), facial expressions and face tracking, icons (LED matrix), input buttons and robot motion.
- Safety for humans and objects at all time.
- Minimal human intervention and simple supervision.

The esthetic of the robot has been designed in collaboration with artists, industrial designers and scenographers. The result of the design of both hardware and software is RoboX: a mobile robot platform ready for the real world.

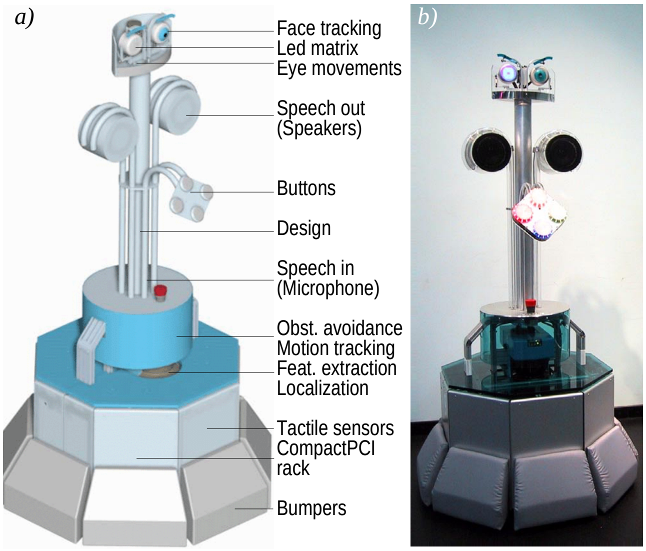

Figure 1: a) Functionality of the tour guide robot RoboX. b) An image of RoboX 9.

The RoboX design is therefore mainly defined by its functionality which is mobility, interactivity and security (figure 1 and 3). Given the above mentioned specifications the mechanical, electronic and software design are now presented.

3.1 Mechanical Design

The lower part (base) of the robot consists mainly in the batteries, the CompactPCI rack with two control computers, the laser range sensors (two SICKs LMS-200), the bumpers and the differential drive actuators with harmonic drives. The base (figure 2) has an octagonal shape with two actuated wheels on a central axis and two castor wheels. In order to guarantee good ground contact of the drive wheels, one of the castor wheels is mounted on a spring suspension. This gives RoboX an excellent manoeuvrability and stability against tipping over in spite of its height of 1.65 m.

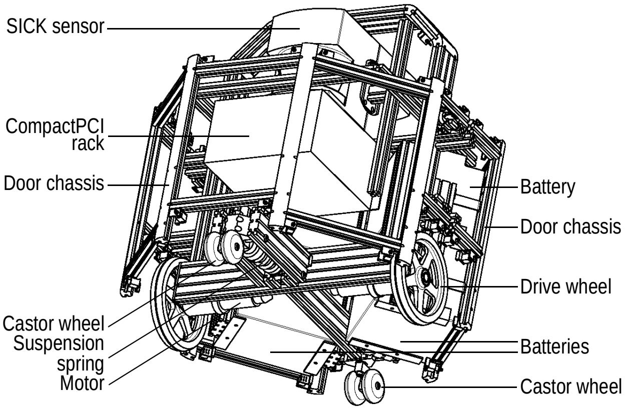

Figure 2: Mechanical design of RoboX base.

The upper part of the robot incorporates the interaction modules of the robot. The face includes two eyes with two independently actuated pan-tilt units and two mechanically coupled eyebrows. The left eye is equipped with a color camera for face tracking. The right eye integrates a LED matrix for the display of symbols and icons. The eyebrows further underline facial expressions with a rotational movement. Behind the face, a gray scale camera pointing to the ceiling is mounted for localization purposes.

The central input device for establishing a bidirectional communication with the humans are four buttons that allow the visitors to select the language, reply to questions the robot asked, and to perform other types of actions. The robot can further be equipped with a directional microphone matrix for speech recognition even though this seems too challenging in the very noisy environment of an exhibition.

3.2 Electronic Design

The control system (figure 3) has been designed very carefully by keeping in mind that the safety of the humans and the robot has to be guaranteed at all time. It is composed of a CompactPCI rack containing an Intel Pentium III card and a Motorola PowerPC 750 card. The latter is connected by the PCI backplane to an analogue/digital I/O card, a Bt848-based frame grabber, an encoder IP module and a high bandwidth RS-422 IP module. Furthermore a Microchip PIC processor is used as redundant security system for the PowerPC card (figure 3).

Navigation is considered as safety-critical and is therefore running on the hard real-time operating system XO/2 [4] installed on the PowerPC. This processor has direct access to the camera looking at the ceiling, the two SICK sensors, the tactile plates and the main drive motors. It communicates with the interaction PC through Ethernet via an on-board switch.

Interaction is not considered safety-critical. It is running under Windows 2000 on an industrial PC. This allows using commercial off-the-shelf (COTS) software for speech synthesis and recognition, and makes scenario development easier. The PC has direct access to the eye camera, the eye and eyebrow controller, the input buttons, the microphone and the two loudspeakers.

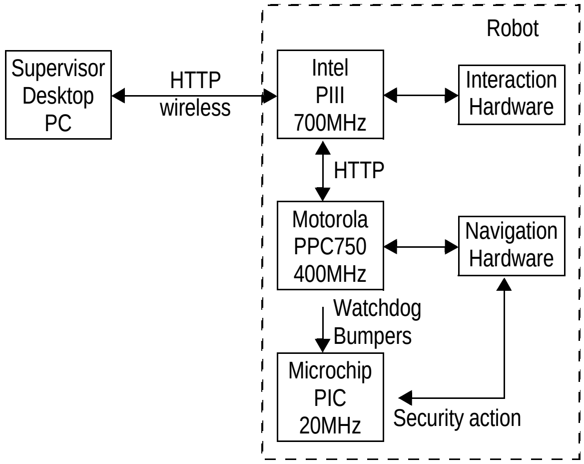

Figure 3: Simple scheme of the electrical design

The whole robot is connected by a radio Ethernet to a supervision computer which allows to track the status of the robot at any time on a graphical interface. However, RoboX operates fully autonomously, the wireless connection is thus not security relevant.

3.3 Software Design

As explained in the section above, the robot is composed of both an Intel Pentium and a Motorola PowerPC systems. The software has been firstly designed without taking into account this fact. However, as soon as the implementation started, the objects have been assigned to one of the two distributed systems. For hardware related objects (mainly sensor drivers) the choice was obvious. For the others, their relevance to safety has been evaluated: due to the hard real-time characteristics of XO/2, all the time-critical objects in relation with the security have been implemented on the PowerPC. Objects requiring COTS components have been implemented on the Windows machine because of their wider availability (f.e. MBrola for speech out, small FireWire camera in the eye for the face tracking, ...).

The resulting object distribution is represented in figure 4. All security critical resources are under control of the XO/2 operating system. Software watchdogs are implemented for the speed controller, the obstacle avoidance and the bumpers. Failure of one of these tasks is detected by the security controller which then either restarts the failed task or stops the robot and sends an e-mail to the maintenance. The security controller sends its watchdog signal and bumper acknowledgements to the redundant security processor (PIC in figure 3): if no signal is received within two cycles (200 ms), meaning that either the security controller or the operating system has crashed, the redundant security software running on the PIC safely stops the whole system. Furthermore the security software on the PIC ensures that the pre-defined maximal speed is never exceeded.

The central object of the interaction subsystem is the scenario controller which accesses all the other objects. A *scenario* is a sequence of tasks from all modalities (speech, face expression, motion, LED matrix, etc.). A sophisticated tour-guide scenario consists of several small scenarios which are played by the scenario controller. The software for scenario creation is a stand-alone application with a user-friendly graphical interface. Even the untrained user can then build its own scenario as free as possible.

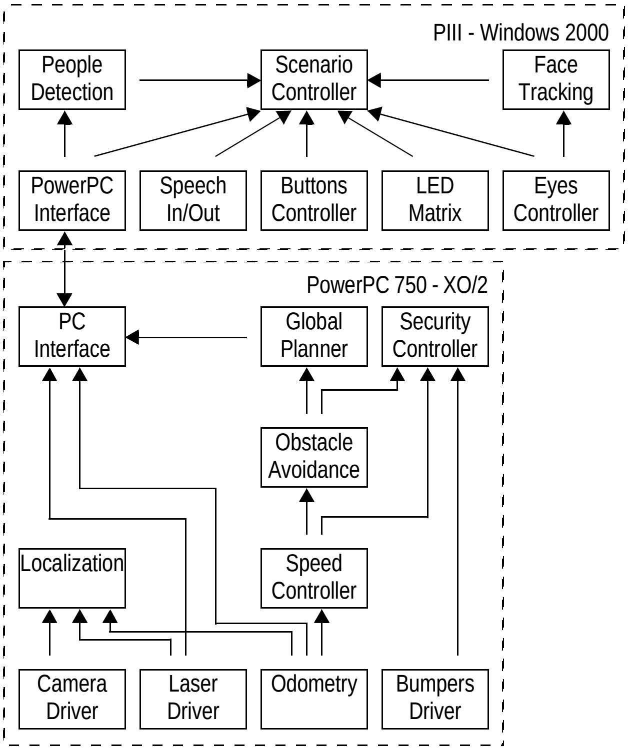

Figure 4: Software architecture on the distributed embedded system.

4. Application Software

In this section, the research relevant techniques implemented on RoboX are briefly presented. Even if some interaction approaches use standard available components (motor control for eyes, text to speech, ...), some other like the face and motion tracking still remain research topics. Navigation is a more extreme example where the whole field remain essentially a research topic.

4.1 Navigation

Map

The map of the environment is a graph-like structure with nodes representing $[x, y, \theta]^T$ positions the robot has to reach in order to perform a certain task. This graph is therefore used for path-planning. Furthermore it contains the information about all the features in the environment. This permits to calculate which feature is visible from the current position of the robot.

Path Planning

On RoboX, three path planning algorithms are used. They work on different levels of abstraction and take sensor readings into account in varying degrees. The topmost layer is the graph-based global planner. It is based on the above mentioned graph structure where nodes are locations of interest (e.g. a showcase, a docking station) and edges denote traversability between locations. The planner employs a depth-first search and generates a length-optimal path. Since the path is global and no sensor readings are taken into account, dynamic path modification cannot be treated on this level.

The second layer of path planning uses the NF1 navigation function in a local grid around the robot [11]. It can thus take into account the current sensor readings and is not limited to nodes of the a-priori map. However, the paths generated by NF1 have a very poor geometry, consisting of linear segments that lie on angles which are multiples of 45°. Another disadvantage is their tendency to graze obstacles.

Smoothing the path and adapting it to dynamic surroundings is done in the third layer of path planning. It is based on the elastic band [13]. The initial plan, generated by the NF1, evolves toward a smoother curve (a list of via points) as long as the elastic band does not "snap". In case dynamic obstacles move in such a way that the minimum clearance along the path cannot be maintained, or if the path lengthens beyond a reasonable amount, the NF1 is called upon again to re-initialize the path.

Obstacle Avoidance

The actual real-time obstacle avoidance task is based on the dynamic window method [8]. Using the dynamic window allows to:

- Take into account the actuator limits of the robot (speed which could result in later collisions are not allowed, motion commands never exceed the robot's speed or acceleration limits).
- Take into account the "exact" robot shape as represented by a convex polygon (extension to any polygon can be done by decomposition).

In comparison to the original dynamic window publications [8], two adaptations have been made:

- Instead of using the distance travelled before hitting an obstacle, the time until collision is used. This solves a singularity when the robot is turning on the spot (any collisions would seem instantaneous because the distance travelled seems zero). It also means the robot will choose more clearance when travelling at higher speeds.
- The objective functions for speed, heading, and clearance are calculated on the actuator phase space (v_l, v_r) instead of the usual (v, ω). Actuator limits are thus more directly taken into account.

The dynamic window task is executed with a frequency of 10Hz. It is part of the time- and security-critical processes on RoboX. Special attention has therefore been paid to optimize its execution time, which should be short and predictable. Both issues are addressed by the use of look-up-tables [14]: Their fixed size give an upper bound to the number of operations, and the intensive calculations (intersecting circles with line segments) can be done once at the initialization step. Adopting look-up-tables means large memory usage, especially when storing floating point values. This problem has been addressed by compressing the tables, using a Lloyd-Max quantizer. The compression is handled transparently, a fact that was facilitated by the object oriented design philosophy underlying our navigation software.

Localization

While autonomous guided vehicles (AGVs) usually employ for their navigation expensive and inflexible environment modifications such as floor tracks or retro-reflective beacons, nowadays localization approaches are ready for *unmodified* environments.

The new localization system [2] employed here, takes advantage of experience from earlier work [1] gathered over a five year period and more than hundred kilometers travelled distance. The method is a global feature-based multi-hypothesis localization using the Kalman filter as estimation framework. It overcomes limitations of the single-hypothesis Kalman filter [7], since the data association problem is explicitly addressed. The robot cannot get lost anymore, as it was possible before, while preserving typical advantages of feature-based approaches, such as very high localization accuracy and an efficient implementation.

The technique which provides this property is a constrained-based search in an interpretation tree [6], [2]. This tree is spanned by all possible local-to-global associations, given a local map of observed features and a global map of model features. The same search is consistently employed for hypothesis generation and pose tracking.

4.2 Interaction

Awareness has been found as one of the most relevant characteristic for man-machine interaction for the human [17]. Being detected by a machine which can look to the interlocutor (face tracking and eyes) or go to him or follow him (motion tracking and robot movement) is one of the major attention keeper. This section presents these two different techniques allowing the robot to detect human presence.

Face Tracking

The left eye of the robot face contains a color camera. An image processing based on the experience from [10] has been developed. It uses the *Intel Image Processing Library* and detects and follows skin colored regions. Using the visual servoing capabilities of the eyes, the robot can look at the person it is interacting with. The algorithm's main steps are:

Skin color detection: Among the different color spaces, the RGB space has been chosen. Green and blue values are normalized using the red channel. This partially cancels differences in illumination. A fixed range for blue, green and brightness values are accepted as skin. Erosion and dilation are performed on the resulting binary image in order to remove small regions.

Contour extraction and filtering: The binary image is clustered and the contour of each cluster is extracted. By employing heuristic filters skin color regions, which are not faces, are suppressed. These filters are based on a rectangular area, the ratio of its height over width and the percentage of skin color inside the rectangle. Another filter is based on the morphology of the skin color region. For regions representing faces, holes within the region that correspond to eyes or the mouth are expected. This reduces the cases where the robot is looking at hands or other body parts.

Tracking: The system tries to update the positions for regions seen before, based on the position of clusters in the current image. Clusters that remain unassigned to previous tracks initialize new tracks that are kept until they leave the camera's field of view.

The information gathered from face tracking is used in several parts. Together with the motion tracking it helps to verify the presence of humans. Furthermore, it is used to look at the person the robot is addressing.

Motion Tracking

The main goal of the motion tracking is to distinguish between moving and static elements in the environment. The proposed algorithm is composed of:

Static map construction: The detection of moving elements is possible since they change the range readings over time. To do so the environment is assumed to be convex and static at the beginning. Then the range readings are integrated. Further on, this is referred to as the *static map*, consisting in all the currently visible elements, which do not move. In contrast to other approaches, which are grid-based, here only one range information for each angle is stored. This results, with the current angular resolution of 0.5° in a map size of 720 elements.

Classification: In the next step, the new information from the laser range finder is compared with the static map. Assuming a Gaussian error of the sensor readings belonging to the same element, a simple *chi-square* test can be used to decide whether the current sensor reading belongs to one of the elements of the static map. If not, the reading is classified as *dynamic*.

Update and validation: All readings classified as static are used to update the static map. Readings labeled as dynamic are used to validate the map: if the reading labeled as dynamic is closer to the robot than the corresponding value from the static map, the latter persists. In case it is farther away than the map value, it is used to update the static map, but it remains also labeled as dynamic.

Clustering: All dynamic elements are clustered according to their spatial distribution. Each cluster is assigned a unique ID and its center of gravity is computed. Thus motion is detected.

The classification, update and validation, and clustering steps are repeated for every new scan. In case of robot motion the process becomes slightly more complicated, since the static map has to be warped to the current robot position.

Having adopted a bidirectional human-robot interaction, focus is put on a single person, the interlocutor. For this, tracking of a single moving object is implemented by means of a Kalman-filter based tracker, which shows satisfying results even in the presence of several persons (figure 5). However the extension to multi objects tracking is under study.

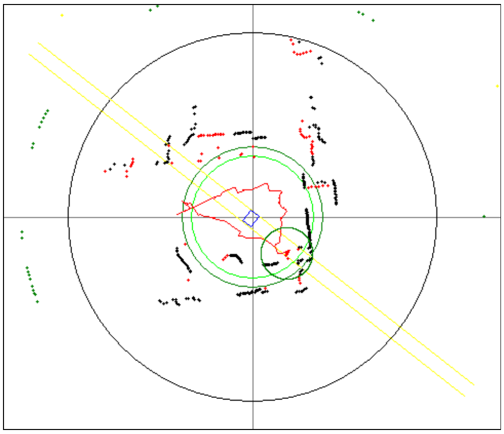

Figure 5: The robot (blue) has followed a person (red close line and green circle) during 35 seconds (107 steps at 3 Hz). Static elements are shown in black, dynamic ones in red. The assumption of convex environments cause the miss-detection of some points as dynamic when they first appear to the laser sensor.

5. Experiments: from Prototype to Product

The whole RoboX development started in February 2001 after the decision of developing 10 robots for the *Robotics* exhibit which is currently running at the Swiss National Exhibition in Neuchâtel. The Swiss National Exhibition takes place once in about 40 years. The current edition, Expo.02, goes from May 15 to October 21, 2002. It is a major national event with 37 exhibitions opened ten and a half hours per day, seven days per week. Within the thematic *nature and artifice*, *Robotics* is intended to show the increasing closeness between man and machine. The central visitor experience is then the interaction with ten autonomous, freely navigating tour-guide robots on a surface of about 320 m^2.

5.1 Production

The first step of the project was the mechanical, electronic and esthetic design which was finished in the middle of May 2001. This lead to the order of most of the components which was delivered starting from June. The next three month were dedicated to the assembly of the two prototypes and the development of drivers. At the beginning of September the prototypes were ready for the one month validation tests. This lead to a small redesign phase. The most sensible parts were the proprietary printed board circuits, which have been simplified and optimized.

In the middle of October 2001 the production started. The RoboX family was ready at the end of April 2002, just three weeks before the start of the *Robotics* exhibit.

5.2 Software Development

Within the team the *Extreme Programming* [3] philosophy has been partially adopted. The goal were to be as dynamic as possible for the development of the application software because the application requirements were very bad defined and the lack of time needed very fast replanning cycles. Fur-

Runs	389
Run time	242 h
Movement time	32 h
Travelled distance	16.8 m
Average speed	0.15 m/s
Failures	55
MTBF	4.4 h

Table 1: *Results of the validation tests for the navigation system. The low MTBF value is due to the continuous introduction and testing of new software.*

thermore, testing and fast integration of new software was a key point for such a complex mechatronic system.

Two main functionalities were to be developed: navigation and interaction. The team was also mainly divided into two sub-teams dedicated to the two main problems. Interaction has been developed from scratch. Navigation has been extended with a new obstacle avoidance and localization technique which are more suitable for an highly dynamic human environment.

5.3 Application: *Robotics* at Expo.02

Due to the various delay in the development of both the hardware and software, but especially to the late delivery of the exposition surface, the exhibit start was still in the test phase. The most important results of the navigation system are shown in table 2 for the first 30 days of operation.

Run time	2447 h
Movement time	1750 h
Travelled distance	582.9 km
Average speed	0.09 m/s
Failures	117
MTBF	20.9 h
Visitors	124031

Table 2: *One month of operation. The navigation system has a MTBF of 20.9 hours. Due to various delays in the development, this is expected to grow in the next weeks.*

Until now six to eight robots operate simultaneously each day with a MTBF of 20.9 hours. They served more than 120000 visitors. Failures are defined as errors which stop the operation of the robot.

6. Conclusions and Outlook

This project represents a milestone in the field of mobile robotics: for the first time tour-guide robots are produced (11 robots) and used for long time (five months) as real products instead of prototypes as in former projects. The paper presents their characteristics first, then goes into details about the research relevant application software. The experiments section is dedicated both to the development and testing with the prototypes and to the first results at the *Robotics* exposition in Neuchâtel, Switzerland. In the next months, this exposition will allow to improve the software and hardware robustness of the whole system.

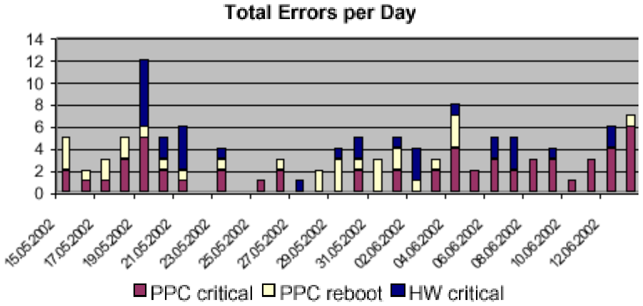

Figure 6: In case of critical errors the robot requires human intervention to continue its task. Sometimes (max three time a day) errors require the reboot of the navigation system. Hardware errors also cause critical interventions.

References

[1] Arras, K. O., N. Tomatis, B. Jensen, and R. Siegwart (2001). "Multisensor On-the-Fly Localization: Precision and Reliability for Applications." Robotics and Autonomous Systems 34(2-3): 131-143.

[2] Arras, K. O., J. A. Castellanos, and R. Siegwart (2002). Feature-Based Multi-Hypothesis Localization and Tracking for Mobile Robots Using Geometric Constraints. IEEE International Conference on Robotics and Automation, Washington DC, USA.

[3] Beck, K. (2000). Extreme Programming Explained, Embrace Change, Addison-Wesley.

[4] Brega, R., N. Tomatis, K. Arras, and R. Siegwart (2000). The Need for Autonomy and Real-Time in Mobile Robotics: A Case Study of XO/2 and Pygmalion. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000), Takamatsu, Japan.

[5] Burgard, W., A. B. Cremers, et al. (1999). "Experiences with a Interactive Museum Tour-Guide Robot." Artificial Intelligence 00(1999): 1-53.

[6] Grimson W.E.L., Lozano-Pérez (1987). "Localizing Overlapping Parts by Searching the Interpretation Tree." IEEE Trans. on Pattern Analysis and Machine Intelligence 9(4): 469-82.

[7] Crowley, J. L. (1989). World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging. IEEE International Conference on Robotics and Automation, Scottsdale, AZ.

[8] Fox, D., W. Burgard, et al. (1997). "The Dynamic Window Approach to Collision Avoidance." IEEE Robotics & Automation Magazine: 23-33.

[9] Graf, B., R. D. Schraft, et al. (2000). A Mobile Robot Platform for Assistance and Entertaiment. International Symposium on Robotics, Montreal, Canada.

[10] Hilti, A., I. Nourbakhsh, B. Jensen, and R. Siegwart (2001). Narrative-level Visual Interpretation of Human Motion for Human-robot Interaction. IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii.

[11] Latombe, J.-C. (1991). Robot motion planning. Dordrecht, Netherlands, Kluwer Academic Publishers.

[12] Nourbakhsh, I., J. Bodenage, et al. (1999). "An Effective Mobile Robot Educator with a Full-Time Job." Artificial Intelligence 114(1-2): 95-124.

[13] Quinlan, S. and O. Khatib (1993). Elastic bands: connecting path planning and control. IEEE International Conference on Robotics and Automation.

[14] Schlegel, C. (1998). Fast local obstacle avoidance under kinematic and dynamic constraints for a mobile robot. IEEE International Conference on Intelligant Robots and Systems, Victoria, B. C., Canada.

[15] Thrun, S., M. Beetz, et al. (2000). "Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva." International Journal of Robotics Research 19(11): 972-99.

[16] Tomatis, N., I. Nourbakhsh, and R. Siegwart (2001). A Hybrid Approach for Robust and Precise Mobile Robot Navigation with Compact Environment Modeling. IEEE International Conference on Robotics and Automation, Seoul, Korea.

[17] Willeke, T., C. Kunz, et al. (2001). The History of the Mobot Museum Robot Series: An Evolutionary Study. Florida Artificial Intelligence Research Society (FLAIRS), Florida.