Supervision and Interaction

Analysis of an Autonomous Tour-guide Robot Deployment

Aurélie Clodic, Sara Fleury, Rachid Alami, Matthieu Herrb, Raja Chatila
LAAS - CNRS
7, Avenue du Colonel Roche,
31077 Toulouse, France
Email: firstname.lastname@laas.fr

***Abstract*— This paper presents the design and the implementation of a new tour-guide robot and reports on the first results that have been obtained after its deployment in a permanent exhibition. The project is conducted so as to incrementally enhance the robot functional and decisional capabilities based on the observation of the interaction between the public and the robot.**

Besides robustness and efficiency in the robot basic navigational abilities in a dynamic environment, our focus was to develop and test a methodology to integrate human-robot interaction abilities in a systematic way.

We first present the robot and some of its key design issues. Then, we discuss a number of lessons that we have drawn from its use in interaction with the public and how that will serve to refine our design choices and to enhance the robot efficiency and acceptability.

I. INTRODUCTION

Today, one of the challenges of robotics is to have robots that achieve long-term missions and are actually helpful to humans.

Rhino[4] and Minerva[17] have been the precursors of a series of tour-guide robots in various museums and exhibition halls [15], [11]. These robots had various degrees of autonomy and were using more or less sophisticated techniques. However, they have all pointed out that studying human-robot interaction was necessary, in its definition as well as its implementation.

It appeared that robots must obey to some “social” clues [6] and led to the development of service robots (e.g. Pearl [12], Care-O-bot II [7], CERO [8], Lino [9] and BIRON [19]).

To study human-robot interaction, an experimentation environment must be found, out of a laboratory and its standard rooms and halls. . . and its robotics scientists who know very well how their ”creatures” work. We have decided to deploy our robot for periods of 2 weeks every 3 months in an exhibition center in Toulouse.

The robot, named Rackham, has already been used at the exhibition for hundreds of hours (between July 2004 and May 2005), accumulating valuable data and information for future enhancements. The project is conducted so as to incrementally enhance the robot functional and decisional capabilities based on the observation of the interactions between the public and the robot.

Besides robustness and efficiency of the robot basic navigational abilities in a dynamic environment, our focus was to develop and test a methodology to integrate human-robot interaction abilities in a systematic way.

In this paper, we describe this tour-guide robot. We begin with a presentation of the exhibition context. Next, we describe the LAAS software architecture ([1]) and the various tools already developed by our group to implement Rackham functionalities. Then, we show how the system is supervised. We conclude with experimental results, comments and analysis.

II. THE EXPERIMENTAL CONTEXT AND SCENARII

A. Mission Biospace

Mission BioSpace is an exhibition designed by the “Cité de l’Espace”[1] in Toulouse to illustrate what could be an inhabited spaceship. It presents 14 interactive elements that propose to visitors a vision of the future.

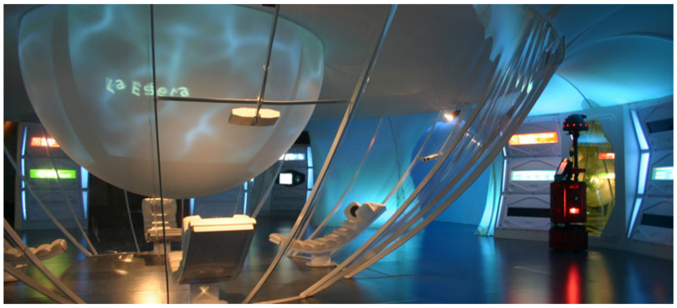


Fig. 1. The Tsiolkovski spaceship: a difficult environment for navigation.

B. A difficult context for navigation and interactions

The exhibition simulates the interiors of an imaginary spaceship (25x10 square meters), including visual and acoustic atmosphere. Hence it represents a difficult context for navigation and interaction (see Figure 1):

- ambient noises make speech synthesis difficult to hear,
- the room is dark with changing background colors,
- supple walls made of tight cloth difficult to model,
- prominent obstacles on the ground and at the head level: not visible by the robot proximity sensors,
- some translucent obstacles: not perceived by the laser range finder,

---

[1] The “Cité de l’Espace” is a space adventure park (http://www.cite-espace.com)

- some narrow passages, which require a precise positioning of the robot to navigate through them,
- clouded environment.

C. A typical Rackham mission

When Rackham is left alone with no mission, it tries to find people to interact with. As soon as a person is detected, thanks to visual face detection, it introduces itself through the virtual 3D face "I'm Rackham and I can guide you in the spaceship" or alternatively it explains how to use its services : "Select your destination using the touch screen".

If the visitor finally selects a destination Rackham first confirms its new mission "OK, I will guide you to…", then plans and displays its trajectory and invites the visitor to follow it.

While navigating, the robot keeps on giving information about the progress of the ongoing travel : a congestion will require to temporarily stop or even to compute an alternative trajectory while a too important uncertainty on the position might call for a relocalisation procedure; temporary "disappearances" of the guided visitor are also detected and indicated by sentences such as "Where are you ?", "Here you are again!". Using various buttons displayed on the interface, the visitor may stop and change the ongoing mission.

III. RACKHAM

A. The robot

Rackham is a B21r robot (iRobot). It is a 4-feet (52 cm) tall and 20-inches (118 cm) wide cylinder topped with a mast supporting a kind of helmet. It integrates 2 PCs (one mono-CPU and one bi-CPUs running P3 at 850 MHz). We have extended the standard equipment with a pan-tilt Sony camera EVI-D70 attached under the helmet, a digital camera mounted on a Directed Perception pan-tilt unit, a ELO touch screen, a pair of loudspeakers, an optical fiber gyroscope and wireless Ethernet.

In order to integrate all these components in a robust and pleasant way the "Cité de l'Espace" has designed a "head" on a mast, the whole toped by an helmet which represents a kind of one one-eyed modern pirate or an African art statue (see Figure 2). The eye is materialized by the EVI-D70 camera fixed upside-down above the helmet, the second camera is hidden in the helmet and one loudspeaker is integrated in the "mouth". The "nose" is only decorative. The mast has been designed as high as possible to keep the cameras away from children's hands.

B. The software architecture

The software architecture is an instance of the LAAS[2] architecture [1]. It is a hierarchical architecture including a supervisor written with openPRS[3] (a Procedural Reasoning System) that controls a distributed set of functional modules.

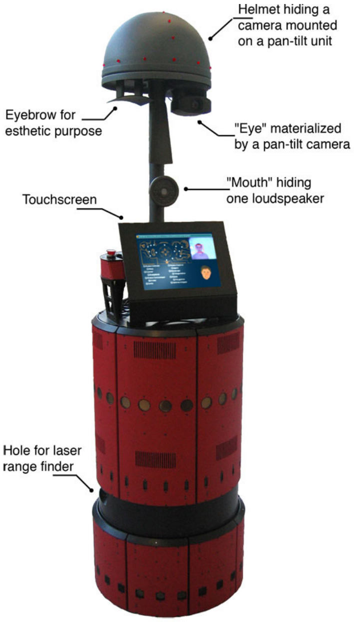


Fig. 2. Rackham and its equipment.

A module is an independent software component that can integrate a set of functions with various time constraints or algorithmic complexity: control of sensors and actuators, servo-controls, monitoring, data processing, trajectory computation, etc.

Each module is created using a module generator called GenoM and thus presents standard behavior and interfaces [5]. The functions encapsulated in a module can be dynamically started, interrupted or (re)parameterized upon asynchronous standard requests sent by the supervisor.

Once started, a service runs autonomously. A final reply that qualifies how the service has been executed is returned to the supervisor with the end of the service. During the execution a module can export data in structured public entities called posters and read data from posters produced by other modules (eg, robot positions, trajectories, maps and so on). The set of posters represent a distributed database of the state of the functional level of the architecture.

For Rackham, we have implemented 15 modules. We now present them according to their role in the system (see Figure 3).

1) Localization: Several modules are involved in the localization of the robot.

First the `rflex` module, which interfaces low level software provided by the manufacturer and which exports in a poster the position computed by the odometry and corrected by the gyroscope. This position gives a good estimate of the motions of the robot. It is associated with a covariance matrix deduced from a probabilistic model error.

To localize itself within its environment the robot uses a SICK laser, controlled by the `sick` module, that exports at the required rate the laser echoes together with segments deduced from aligned echoes. Another module, `segloc`, matches

---

[2] LAAS stands for: "LAAS Architecture for Autonomous Systems".

[3] The set of tools used to build an instance of this architecture (GenoM, openPRS, pocolibs, etc) are freely distributed at the following url: http://softs.laas.fr/openrobots.

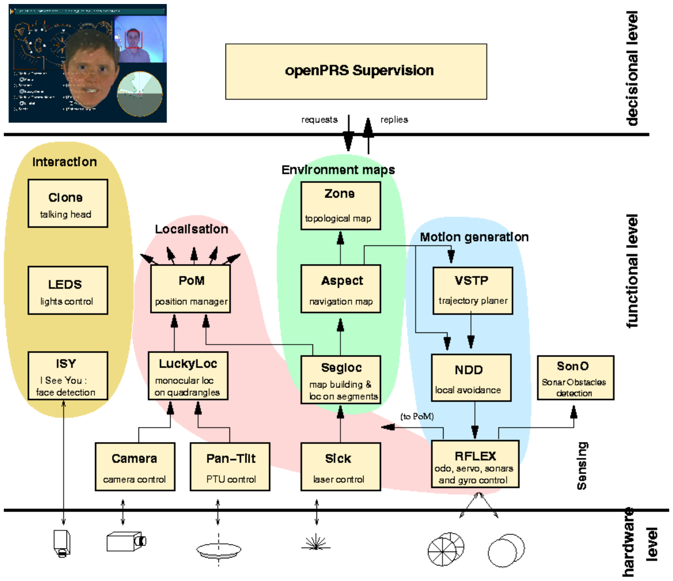


Fig. 3. The functional level of Rackham and its 15 modules.

these segments with segments previously recorded in a map thanks to a classical SLAM procedure. However the map (see Figure 4) is effectively updated only during exhibition closing time (no public).

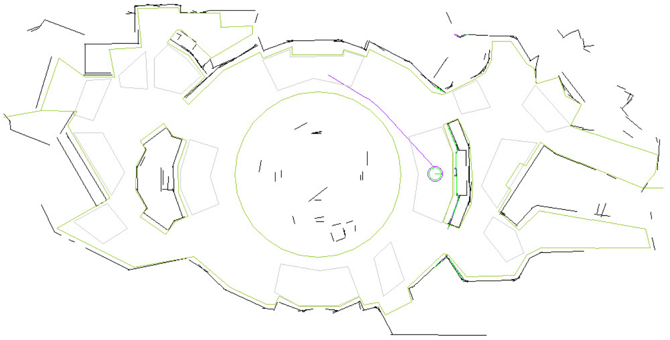


Fig. 4. The map of the environment built by Rackham contains 232 segments (black). It has been augmented with virtual obstacles (green or dark-grey) and target zones (light-gray).

The localization being a very critical ability, a third localization modality, based on vision, has been designed. The camera is controlled by the module `camera` that produces images to be processed by another module called `luckyloc` that extracts, identifies and localizes planar quadrangles that appear on the furniture. However, this function is not yet totally functional.

Finally, the various uncertain positions exported by the modules `rflex`, `segloc` and `luckyloc` are merged by `pom`, the position manager module. This module is able to integrate positions computed at various frequencies and even to propagate "old" position data. Various fusion strategies can be selected like Kalman fusion or integration of the measured motions relatively to the most reliable positions. The supervisor is informed in case of localization problems with one of the modules, fusion difficulties or significant uncertainties on the position. Depending on the problem and the context, various strategies are applied.

It is important to note that the `pom` module allows to centralize the robot positions and to export one and only one reference position. All the other system components do not need to know how this position is obtained. This procedure can change dynamically without disturbing the position consumers. It is a very important mechanism to manage redundancy and an essential feature for this critical function.

On top of this geometric positioning, several topological zones corresponding to places of special interest ("TARGETS"), to dangers for the navigation ("OBSTACLES") not always visible by the robot sensors like prominent or transparent furniture), or to other special areas ("SPECIAL") have been defined in the environment. The `zone` module continuously monitors entrance and exit of the robot from these zones and informs the supervisor.

2) Obstacles and people detection: Obstacle detection is a critical function both for security reasons and for interaction purposes. The most efficient sensor is once again the laser. However Rackham's laser can only look forward (over 180 degrees) in an horizontal plan.

To partially overcome these limitations, the laser data are integrated in a local map by the `aspect` module and filtered using knowledge about the global map (segments and the virtual obstacles[4]).

Thus, `aspect` exports, every 40 ms, a local map of the surroundings of the robot which represents the free space and which distinguishes static (ie, that belong to the environment or the virtual obstacles) and dynamic obstacles (probably visitors). This local map is permanently displayed on the bottom right of the interface (see Figure 5).

Using this representation, `aspect` is able to inform the supervisor when the robot is surrounded by unpredicted obstacles. The red leds on the helmet flicker at a frequency proportional to the obstruction density of dynamic obstacles.

To reinforce the assumption of presence near the robot, the supervisor can use the services of the `sono` module that detects motion all around the robot using the ultrasonic sensors. Unfortunately our ultrasonic sensors produce some audible noise which seem to disturb visitors interacting with the robot.

A much more robust people detector is offered by the module called `isy` (or, "I See You") which is able to detect faces in real time from a color camera image. The detector uses a cascaded classifier and a head tracker based on a particle filter [3]. `Isy` controls the camera orientation in order to track the detected face. It informs the supervisor when it detects or looses a face.

From the direction and the size of the face it is able to estimate the 3D position of the detected person with a sufficient precision (about 10 cm for the height and 20 cm for the range).

---

[4] Let us recall that in the context of Mission BioSpace with prominent and transparent obstacles this notion of virtual obstacles is very important.

The ambient light (weak and changing) of the space ship does not provide enough light; a ring of white leds fixed around the lens provide a range of detection of about three meters.

3) Trajectory and motion: Rackham being a guide, it must be able to take visitors to various places in the exhibition and that are displayed on the interactive map. For the robot they correspond to polygonal `target zones` (see §III-B.1) and to the position of the associated element of interest (which can be itself out of the polygon) that the robot will have to comment.

The robot motion involves mainly three modules:

- `rflex` that manages the lower servo-control loop, transmitting the reference speeds at the micro-controller.
- `ndd` integrates a local avoidance procedure based on an algebraic instance of Nearness Diagrams [10]. The input obstacles are provided by the aspect map (see §III-B.2).
- `vstp` is a Very Simple [but very efficient] Trajectory Planner based on an algebraic visibility graph optimized with hash tables[5]. A main visibility graph is precomputed for the static segments of the map. Dynamic obstacles can be added and removed in real-time upon supervisor requests.

The strategy used to coordinate the implied modules is dynamically established by the supervisor. The objective is of course to reach the target zone while avoiding obstacles. The planned trajectory is an Ariadne's thread for `ndd`: the vertices of the broken line are sub-goals. Usually the supervisor has to intervene only if `ndd` does not make progress towards the goal. In such case, various strategies can be applied: computing of a new trajectory taking into account the encountered obstacles, waiting for a while, starting an interaction with people around, etc. The motion is over when the robot is inside the target zone.The maximum speed that the robot can achieve in this mode is about 0.6 meters per second.

4) Interactions: For now, the interactions are mainly established through the following components:

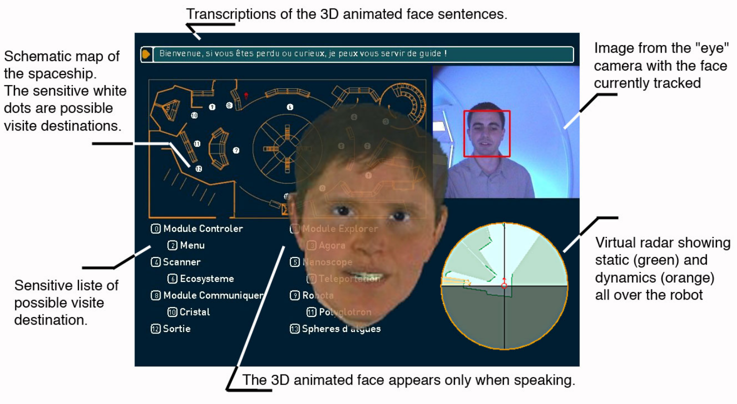
- the dynamic "obstacles" detectors (`aspect` and `sono`),
- the `isy` face detector,
- the 3D animated face with speech synthesis,
- displays and inputs from the touch screen,
- control of the robots' lights.

While the first two allow to detect the presence or the departure of people, the last ones permit the robot to "express" itself and thus establish exchanges.

The vocal synthesis is highly enriched by a 3D animated head displayed on the screen. This talking head, or `clone`, is developed by the Institut de la Communication Parlée (http://www.icp.inpg.fr). The clone is based on a very accurate articulatory 3D model of the postures of a speaking locutor with realistic synthetic rendering thanks to 3D texture projection. From a given text, the speech synthesizer produces coordinated voice and facial movements (jaw, teeth, lips, etc.).

The directions of the head and of the eyes can be dynamically controlled. This capability is important as it allows to reinforce an interaction, looking towards the interlocutor face detected by `isy`, or to point out an object or a part of the exhibition currently mentioned by the robot.

The clone appears in front of the touch-screen each time the robot has to speak (see Figure 5). Meaningful messages have been prepared, corresponding to the various situations encountered by the robot or to the places that need to be described during the visit.



Transcriptions of the 3D animated face sentences.

Bienvenue, si vous êtes perdu ou curieux, je peux vous servir de guide !

Schematic map of the spaceship. The sensitive white dots are possible visite destinations.

Image from the "eye" camera with the face currently tracked

Sensitive liste of possible visite destination.

Virtual radar showing static (green) and dynamics (orange) all over the robot

The 3D animated face appears only when speaking.

Fig. 5. A view of the interface of the touch-screen.

The robot interface, written in Java, is made of independent components or microGUIs directly controlled by the supervisor through a dedicated communication channel.

The available microguis are (Figure 5):

- a map of the environment including the current robot position and trajectory,
- the local "aspect" map displayed as a radar,
- the image of the "eye" camera with the faces currently detected by isy,
- the clone or talking head,
- pop-up warning messages,
- top messages,
- localization window (init).

5) Controls of the functional level: All the presented modules are controlled by the supervisor according to the ongoing mission, the context and the robot state.

This set of modules offers a good degree of redundancy for several functions as shown on Figure 3. It greatly helps in making the robot's behavior robust, and provide tools for the supervisor to adapt itself to a large set of varying situations. This supervisor is briefly described in the next section.

IV. SUPERVISION

A. General description

Rackham is used in a context where there is no need for a high level planner i.e. a system that synthesizes a partially ordered set of tasks to be performed to reach a given goal. Consequently, the highest level of decision consists in

---

[5] VSTP is freely distributed: http://softs.laas.fr/openrobots/.

selecting and refining the most adequate tasks according to the current situation (availability of visitors, energy level…).

Hence, the supervisor must be endowed with the following capabilities:

- task selection,
- context-based task refinement,
- adaptive task execution control.

In its current configuration, Rackham, as a tour-guide in the exhibition, has basically to deal with two main tasks: *the search for interaction* (the robot, left alone in the exhibition, tries to attract a visitor in order to interact with him), *the mission* (the robot, according to the visitor's choice, brings him to a selected place).

Depending on the context and the level of abstraction, task execution is based on three aspects:

1) the definition of a state space, an action space and the construction of a policy[6],
2) the construction of robot primitives (based on the action space definition),
3) the execution of the policy and the control of this execution.

Various schemes are proposed in the literature in order to refine and execute robot tasks in the presence of uncertainty. Mainly they partition the state space [16] , or the action space [13], or both [18].

We do not intend to discuss here the various means that may be used to build policies. What we want to stress is the importance of the models (variables, primitive actions and their parameters, primitive observations) and the ability to decompose them. One key aspect is to construct efficient and robust motion execution primitives that are able by themselves to deal with local contingencies. This allows to reduce the burden of the higher levels and limits the complexity of the associated policy.

Our choice was to use relatively low level observation and action primitives in order to leave as much flexibility as possible at the supervision level. Indeed, as we will see in the sequel, the performance of tasks in the vicinity and/or in interaction with humans is not compatible with a "black-box" strategy.

Another interesting aspect on which we focus is how the task execution process is influenced by the need for human-robot interaction.

B. Executing tasks in presence of humans

When a task is given, our robot not only needs to execute it, but it also needs to be able to explain it (by exhibiting a legible behavior or by displaying relevant information) and it should allow humans to act on the course of its actions during their execution.

For instance, during the *mission* task, Rackham should not only be moving toward its goal and avoiding obstacles, it also has to maintain the interaction with the humans: waiting for possible inputs like abort or change the mission and displaying any relevant information that may be needed.

TABLE I

EXAMPLE OF POLICIES FOR THE *mission* TASK

move	**trajectory**	**localization**	**navigation action**	**interaction action**
nothing	unavailable	good	trajectory planning	explain action
nothing	available	good	trajectory following	wait
begin	available	good	wait	follow me !
moving	available	good	wait	wait
	(...)		(...)	(...)
moving	available	bad	motion trajectory stop	explain state
stopping	available	bad	wait	explain state
stop	unavailable	bad	re-init position manager	ask freeing the way
	(...)		(...)	(...)
moving	available	good	wait	wait
blocked	available	good	motion trajectory stop	explain state
stopping	available	good	wait	explain state
stop	unavailable	good	wait	explain state
nothing	unavailable	good	trajectory planning	explain action

There are various speech-based or visualization-based functions that allow to provide feedback to the user mainly in terms of messages. Other information such as trajectory, robot position, etc, are displayed directly by the interface as soon as there are elaborated by the modules (without direct intervention of the supervisor, we will come back to this point in section V-C.2).

Possible robot actions can be partitioned in two sets:

- navigation related actions: trajectory planning, trajectory following, motion trajectory suspend, motion trajectory resume, motion trajectory stop, change speed, re-initiliaze position manager, re-initiliaze position estimator, end, wait, error.
- interaction related actions: we do not list here all the actions because there are many of them, but fundamentally they are of three kinds: actions explaining the current navigation action (one for each navigation action), actions explaining the state of the robot ("I'm lost.", "We are blocked",…) and actions trying to engage people with the robot ("follow me !", "Please free the way before me to help me re-localize").

The action selection is essentially performed on the basis of a common state space. Three state variables are needed:

- `localization quality`: *good, average, bad*
- `trajectory`: *unavailable, available, error*
- `move`: *nothing, begin, moving, blocked, error, stopping, stop, ended*

This state space is an abstraction, for the supervision, of the modules feedback. For instance, if the `zone` module has already notified an entrance in the target zone, then the notification of the end the motion by `ndd` module will switch the value of `move` to *ended*.

Considering that the two activities, navigation and interaction, are of different nature and that one can speak and move at the same time, we decided to separate navigation and interaction policy treatment. This seems also convenient in order to provide a "legible" robot behavior (not all robot actions need to be displayed). Table I, shows an example of the policies.

---

[6] By policy we mean : "a solution that specifies what the agent should do for *any* state that the agent might reach" [14], no matter the way it has been computed or deduced.

Another key advantage of this separation is to build reusable policies. For instance, in the current system, the navigation policy can be used alone, when the robot has to perform a navigation task without interaction (e.g. when it is heading to its re-charging station) or in conjunction with a different human interaction policy.

C. *Executing tasks taking humans into account*

In a second step, we have extended the state space of the interaction policy.

In order to take into account the observation functions dedicated to the human-robot interaction (detection, tracking of human faces, detection of humans blocking the path...), two state variables have been added to the interaction state space:

- `path state in front of the robot`: *free, blocking, blocked, freeing,*
- `face detection state`: *I see you, I don't see you, I see you again.*

Notice that it expresses in fact in which direction the things go: better or worse because it is the interesting information for the robot (and for the visitor to whom this will be shown).

Currently, we just use this information to give feedback to people with messages: are you there ?, here you are !, we are blocked, we can pass again...

In the future, they will be used to select actions that will influence the task execution itself (slowing down, suspending or even aborting execution, etc). This imposes to build task execution policies that are able to react to such inputs.

This kind of information will also be useful to choose the best media or the best combination of them to be used to transmit messages to the humans.

V. RESULTS AND ANALYSIS

A. *Quantitative results*

Between March 2004 and February 2005, Rackham has spent ten weeks at the "Cité de l'Espace" in five venues[7].

During the last two stays, the robot was sufficiently robust to be operated by the personnel of the Cité de l'Espace without our intervention.

We collected various data for analysis purposes: all the requests to the modules and their reply, the covered distance, the visitors interactions, etc.

The results presented below are a synthesis of the data collected during the periods from October 5th to October 15th and from February 7th to February 20th. During these stays Rackham was put on mission by the organizers for a total of 58 hours. The robot is then permanently in interaction and was requested for 1565 visits to a place of interest. About 20% of these missions were voluntarily interrupted by the visitors before the end. For about 2% of the missions the supervisor has detected a problem (mainly an abnormal delay to refresh a low level data and few logical inconsistencies) and called for the operator for security reasons.

The total covered distance is about 16 km (for an environment of 25x10 square meters). The average speed of about 0.55 km/h integrates all the disturbances (crowds, immobile and dense group of visitors) during the motion.

From October 5, 2004 to October 15, 2004					
day	number of missions	distance in meters	duration hh:mn (motion)	number of requests	average speed (km/h)
1	17	71	0:34	379	0.44
2	63	543	2:39	2100	0.57
3	46	495	1:27	2210	0.61
4	9	100	0:11	318	0.63
5	76	815	2:15	2377	0.63
6	97	802	2:20	2967	0.54
7	54	542	2:12	2081	0.52
8	89	904	3:41	2810	0.59
9	54	607	2:19	1751	0.60
10	58	681	1:57	2019	0.58
11	170	1611	5:37	5084	0.57
	733	**7171 m**	**32:12**	24096	**0.57**

From February 7, 2005 to February 20, 2005					
day	missions	distance	duration	requests	speed
1	40	395	1:25	2801	0.51
2	49	555	1:32	2719	0.56
3	44	487	1:18	2557	0.62
4	82	851	3:32	4338	0.44
5	82	881	2:28	4209	0.58
6	70	739	1:49	3609	0.56
7	85	884	2:14	4338	0.50
8	71	815	2:24	3984	0.53
9	55	663	1:31	3154	0.60
10	78	912	2:29	4742	0.49
11	71	872	2:08	4214	0.54
12	91	994	2:49	4632	0.53
13	14	161	0:27	733	
	832	**9209 m**	**26:06**	46030	**0.54**

The results presented above are a synthesis of the data collected during the last two stays. Rackham has executed 1575 missions requested by the visitors of the exhibition and traversed nearly 16.5 km.

B. *Visitor behaviors*

1) Human robot interaction: It is striking to notice how the behavior of the visitors highly depends on their age.

Kids immediately identified Rackham as a robot (although it is very different from cartoons robots). They are not afraid at all and even often too effusive, catching the camera when it does not look at them or pushing the robot when it does not move fast enough.

Teenagers and young adults try to find out how it works or how to make the system fail, blocking its path or clicking on all the buttons. They are also very attracted by their own

---

[7] See http://www.laas.fr/ sara/laasko.

Fig. 6. Head of Rackham emerging from a crowd of kids.

image displayed by the face tracker.

While adults are anxious to understand every thing (technological exhibitions serve to transmit knowledge), the elderly sometimes do not even imagine that this thing can move or that they can communicate through the (tactile) screen.

2) Interface misunderstanding: Among the various data displayed on the interface, we have implemented a “radar-like” representation of the local map and of the proximity data, in order to show how the robot models the dynamic obstacles (visitors), and static ones (from the map). Many visitors, looking at the robot, interpret that element as a “virtual” joystick to make the robot move and are disappointed on the robot inactivity.

3) How does it work ?: There is a difference between what people think the robot can do and what it really does. Generally, people do not understand that the robot is deaf, especially because it has an animated face that can speak.

Besides, they do not comprehend how the robot localizes itself or detects obstacles. They generally think that the robot uses the ultra-sonic sensors that they see (but that we do not use) or its cameras. They do not understand why the robot does not stop when they put their hands on them.

That brings out the difficulty for a robot to be understood. People will gradually understand better how a robot works, but we have to dedicate special efforts to make the robot abilities and behaviors more “readable”.

C. Towards control and data-flow for interactivity

Rackham, with its current functionalities and limitations has been an attractive tour guide for many visitors. However, by observing how people interact with it and analyzing the cases where it failed to accomplish its mission (either because of a software failure, or because the humans did not understand or follow what the robot was expecting from them – following it, or freeing the way when necessary), we are able to draw some lessons and define future work.

1) Functional data relevance: A module often carries out complex computations and manipulates a large amount of data to accomplish the required functionalities. However the output result is generally simplified to be easily used and interpreted. For example we would naturally expect a face detection module to return a flag indicating if there is somebody or not. . . and that’s what it does. But this binary information is a strong impoverishment of the result. Indeed in real systems results are rarely certain and most probably the algorithm has an idea of the confidence on its result. This level of confidence is a a key issue for a correct control. The problem is: can we measure and express such an uncertainty in a coherent, standardized and comprehensible manner for all the data produced by the modules ?

Other data, hidden within the modules, can be relevant for the system control. For instance a module that localizes the robot using proximity data has an idea of the kind of environment encountered (eg. corridor, cluttered, etc.)

2) Interface limitations: Nor the supervisor nor the interface have access precisely to the other’s state. This brings limitations and complications.

First of all, it is difficult for the supervisor to treat messages given back by the interface.

More annoying for the interaction abilities, the interface is loaded at the beginning and although the information displayed on it is regularly updated, it is not possible to put forward particular information when it is necessary (or hide it when it is not), to display elements bigger or not according to their current importance. For example, when the visitor has to choose his destination, the important things is the map and the possible destinations, not the face detection that distracts him.

In fact, we missed the possibility to control directly what is displayed on the interface at a supervisor level.

We have now enhanced and redesigned this part of the system to allow better communication between the supervisor and the interface and better control from the supervisor on the interface.

VI. CONCLUSION AND FUTURE WORK

In this paper we have proposed an instance of the LAAS architecture that is adapted to human-robot interaction. We have built a supervisor which represents the interaction tasks in addition to traditional navigation tasks and is able to explain its behavior and to interact with the people in the vicinity of the robot during its mission. We have combined this supervisor with a functional level (implementing localization, motion control with collision avoidance and interaction with users) that provide enough flexibility and redundancy to achieve a certain level of robustness in an relatively “hostile” environment.

Rackham now has solid foundations which allows it to navigate in a robust manner and to establish a simple interaction with people in a real world environment. It has been effectively used quite intensively and is considered as an attractive and successful component of the overall exhibition.

But this is only the first page of the story. We now work to enhance Rackham’s interaction and perception capabilities.

The integration of such capabilities will be done with the concern of developing a systematic manner to integrate more sophisticated context interpretation and to provide decision-making to synthesize and control interactive tasks at various levels.

ACKNOWLEDGMENTS

The work described in this paper was conducted within the EU Integrated Project COGNIRON ("The Cognitive Companion") and was funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020 and by the French National Program ROBEA. This work was also supported by the Belgian programme on Interuniversity Poles of attraction AMS, initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. Numerous persons have contributed to the project : the authors want to thanks: Aymeric Yvernes, Coline Lelong-Pantel, Florent Lanterna, Stéphane Plais, Guillaume Infantes, Maxime Cottret, Abdelatif Baba, Paulo Menezes, Ludovic Brethes, Frederic Lerasle; from the Space City Museum : Jean-Noël Plachez, Véronique Hallard, Maxime Randisi and Stephan Lauriol; and from ICP Grenoble : Frédéric Elisei.

REFERENCES

[1] R. Alami, R. Chatila, S. Fleury, M. Ghallab, F. Ingrand, "An architecture for autonomy", *International Journal of Robotic Research*, Vol.17, N°4, pp.315-337, Avril 1998.

[2] G. Bailly, M. Bérar, F. Elisei, and M. Odisio, "Audiovisual speech synthesis", *International Journal of Speech Technology*, 6:331-346, 2003.

[3] L. Brèthes, P. Menezes, F. Lerasle, and J. Hayet, "Face tracking and hand gesture recognition for human robot interaction," *International Conference on Robotics and Automation, New Orleans, May 27 - June 1 2004*.

[4] W. Burgard and A.B. Cremers and D. Fox and D. Häehnel and G. Lakemeyer and D. Shulz and W. Steiner and S. Thrun, "Experiences with an interactive museum tour-guide robot", *Artificial Intelligence* , 114 (1999) 3-55.

[5] S.Fleury, M.Herrb, R.Chatila, "GenoM: a Tool for the Specification and the Implementation of Operating Modules in a Distributed Robot Architecture", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Grenoble, France, 1997.

[6] T. Fong, I. Nourbakhsh, K. Dautenhahn, "A survey of socially interactive robots," *Robotics and Autonomous Systems, Special issue on Socially Interactive Robots*, 42(3-4), 2003.

[7] B. Graf, M. Hans, and R. D. Schraft. "Care-O-bot II - Development of a next generation robotic home assistant",*Autonomous Robots*, 16(2):193-205, 2004.

[8] H. Huttenrauch and K. Severinson Eklundh. "Fetch-and-carry with CERO: Observations from a long-term user study with a service robot",*In Proc. IEEE Int. Workshop on Robot-Human Interactive Communication (ROMAN)*, pages 158-163. IEEE Press, 2002.

[9] B. J. A. Krose, J. M. Porta, A. J. N. van Breemen, K. Crucq, M. Nuttin, and E. Demeester. "Lino, the user-interface robot",*In European Symposium on Ambient Intelligence (EUSAI)*, pages 264ñ274, 2003.

[10] J. Minguez, L. Montano. "Nearness Diagram Navigation (ND): Collision Avoidance in Troublesome Scenarios",*IEEE Transactions on Robotics and Automation*, p 154, 2004.

[11] I. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer and A. Soto, "An Affective Mobile Educator with a Full-time Job," *Artificial Intelligence*, vol. 114,num. 1-2, pp. 95–124, 1999.

[12] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, "Towards Robotic Assistants in Nursing Homes: Challenges and Results.," *Robotics and Autonomous Systems*, 42(3-4), 2003.

[13] J. Pineau and S. Thrun. "An integrated approach to hierarchy and abstraction for POMDPs," *Technical Report CMU-RI-TR-02-21* , Carnegie Mellon University, 2002.

[14] S. Russell and P. Norvig, *Artificial Intelligence : A Modern Approach,* Prentice Hall, 2003.

[15] R. Siegwart, and et al. (2003) "Robox at Expo.02: A Large Scale Installation of Personal Robots." *Special issue on Socially Interactive Robots, Robotics and Autonomous Systems* , 42 (2003) 203-222.

[16] G. Theocharous, K. Murphy, L. Kaelbling, ‘’Representing hierarchical POMDPs as DBNs for multi-scale robot localization," *ICRA'04 (Intl. Conf. on Robotics and Automation)* , New Orleans, LA, USA, 2004.

[17] S. Thrun and M. Beetz and M. Bennewitz and W. Burgard and A.B. Cremers and F. Dellaert and D. Fox and D. Haehnel and C. Rosenberg and N. Roy and J. Schulte and D. Schulz "Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva", *Journal of Robotics Research* , vol. 19, num. 11, 2000.

[18] W. Turkett, Jr., and J. Rose, "Planning With Agents: An Efficient Approach Using Hierarchical Dynamic Decision Networks," *Proceedings of the Fourth International Workshop on Engineering Societies in the Agent World* , London, England, 2003.

[19] B. Wrede, A. Haasch, N. Hofemann, S. Hohenner, S. Hüwel, M. Kleinehagenbrock, S. Lang, S. Li, I. Toptsis, G. A. Fink, J. Fritsch, and G. Sagerer. "Research issues for designing robot companions: BIRON as a case study", *In P. Drews, editor, Proc. IEEE Conf. on Mechatronics and Robotics,* volume 4, pages 1491-1496, Aachen, Germany, September 2004. Eysoldt-Verlag, Aachen.