A Receptionist Robot: Interaction and Coordination

Manuel José B. C. Malhado
Institute for Systems and Robotics
Instituto Superior Técnico

Abstract—This article presents a project that consists on the development of a receptionist robot for the Institute for Systems and Robotics (ISR), Lisbon. This robot is stationed at ISR’s 6th floor elevator lobby where it waits for approaching visitors. At this point it attempts to interact with them in order to find out whether they wish to be lead to a specific room on this floor.

The followed development methodology focuses on the integration of several modules, featuring navigation and localization capabilities, a graphical interface, speech recognition and synthesis, people detection, face detection, and behavior control, in order to achieve an autonomous system. In order to save time and effort, as well as obtaining a robust solution, “off-the-shelf” software packages are used whenever possible.

This project was divided in two sub-projects. The present one focuses, apart from the conception of the robot’s hardware and software architecture design, on its human-robot interaction capabilities, as well as on all robot modules’ integration and coordination.

Experimental results obtained in order to evaluate the employed speech recognition engine robustness and the integrated system overall performance, are also presented in this article.

This project autonomous robot development process is described in much more detail in this thesis [1], specially in the areas focused by this article.

Index Terms—Receptionist robot, human-robot interaction, graphical interface, speech recognition, behavior control.

I. INTRODUCTION

AUTONOMOUS robotics is a research field which has been in development since the middle of the 20th century and it is currently one of the main areas of interest within the field of Robotics. Even though great break-throughs have been achieved throughout the years, this area still has a long way to go, as much in terms of sensory, mechanical, and mobility capabilities as well as in the artificial intelligence and decision making domain, before it can achieve efficient and flexible behaviors comparable to the ones observed in animals and humans.

Current real life applications using robot agents are relatively scarce and usually restricted to particular areas (such as industry and space exploration) where the use of human labor is not possible or is inconvenient, either because the task at hand is life threatening or inaccessible to human beings.

A common requirement for the environment where the robot shall operate, is that it has to be relatively predictable, since current robot agents’ capacity to adapt to new and unexpected situations is still very limited. This fact is a major reason why there are still so few successful initiatives that use robotic agents to assist and interact with regular people, since these can be extremely unpredictable and different from each other.

People’s unpredictability is very much related with the different reactions they can express towards an unusual and unknown identity such as an automaton. For this reason, today’s key for developing a successful people interacting agent might not be employing an extremely complex decision system that seeks to cover all possible situations, but rather to use a human being’s almost unlimited self adaptive capacity to adjust itself to the robot platform.

This can be achieved by providing, on the one hand, the means to help them feel comfortable with the whole situation, and on the other hand, to guide them through the process of interaction by initially taking the initiative to start a “conversation”/communication, and then directing and narrowing it through an expected list of reasoning.

In order to make a person feel more at ease while interacting with the robot agent, besides presenting an intuitive and enjoyable interface, the automaton might feature human/animal like characteristics, with whom people are accustomed to deal with.

The current project falls under the mentioned field of applications, consisting on the development of a demonstration robot targeting an audience of people who manifest a certain curiosity for the field of robotics. The robot should behave as a receptionist that socially interacts with approaching people, being capable of guiding them to on-the-spot requested locations within a known environment. Adequate dimensions and hardware support are required features for the robot to navigate around the intended environment and to interact/communicate with people using speech and visual interfaces (some hardware was readily available from previous projects, but extra devices had to be acquired).

The desired robotic agent requires several specific capabilities, covering a set of different fields of research and development. Since various promising initiatives (which may or may not have been originally designed to be implemented on a robotic platform) capable of solving particular robot requirements are available as commercial or open-source software packages, it is of this project’s interest to find the most robust and powerful solutions and adapt them to the proposed goals.

The Receptionist’s development process consisted of several individual steps, starting with the research for featured capabilities on similar initiatives, followed by the conception of the robot’s software and hardware architectures, the implementation and testing of individual modules, and
finally the progressive integration of each developed subsystem into a fully working system. This project’s work was divided into two separate theses which shared the tasks of research and conception, but from that point onward different responsibilities were assigned to each other. This thesis is responsible for the development of all the Robot’s human-robot interaction capabilities, as well as for the implementation of the receptionist’s behavior and integration of all system’s individual parts. The companion sub-project [2] is devoted to the implementation of a navigation and localization solution, capable of fulfilling the requirements set, and to handle all Robot image processing, necessary to implement people and face detection functionalities.

II. CONTEXT SCENARIO

The Receptionist robot is stationed in the elevators lobby of ISR’s 6st floor, waiting for a person to approach it. Upon detection of their presence, the robot approaches the person, facing them and then it starts dialog interaction by introducing itself and offering its services. If the person shows interest, by acknowledging the robots introductory intervention, the receptionist inquires about the room/location the person would like to be guided to, and subsequently it starts to move towards the destination indicated by the person. Upon arrival to the requested destination, the receptionist announces the arrival and inquires whether further assistance is needed. If the person shows to be already satisfied with its help, the robot returns to its starting position, where it awaits for the arrival of another person.

III. SYSTEM ARCHITECTURE

Considering the problem at hand, it was decided that a modular architecture would be the most fitting. This architecture is very flexible, permitting the segmentation of the development process (design, implementation and testing) into separate and somewhat independent modules (since the work effort is divided into two different theses, each of them is responsible for the development of specific modules), and easing the task of future development by allowing the replacement of specific modules and introduction of new ones without the need to alter the entire system.

The modules which are fully covered by this thesis (i.e., Coordination, Speech Synthesis and On-Screen Interface, and Speech Recognition), are described in detail in the following sections.

A. Coordination

Responsible for the top level system coordination between modules, it controls all the receptionist robot’s reactions to external stimuli, ultimately resulting in the robot’s overall behavior.

This module runs over a hierarchical finite state machine, implemented using the UML StateWisard toolkit framework\(^1\).

B. Navigation and Localization

As its name implies, this module covers all the robot’s navigation and localization requirements.

This module is implemented over Carnegie Mellon Navigation toolkit (CARMEN)\(^2\), an open source software package for mobile robot control which performs the referred tasks, using the data provided by the receptionist’s laser sensor and odometry board, and a previously generated map.

A detailed description of this module can be found in [2].

C. On-Screen Interface and Speech Synthesis

A graphical interface was developed with the use of wxWidgets [4], a Cross-Platform GUI programming toolkit, and Xface [5], a toolkit for the creation of embodied conversational agents. It has several GUI elements that can be accessed through the touch-screen, and it is responsible for all non-voiced interaction with the user.

Speech synthesis is also this module’s responsibility, and it is performed by Microsoft’s Speech Application Programming Interface (SAPI)\(^2\), which is incorporated in Xface for lip-synchronization purposes.

For a more detailed description on this module, refer to section VII.

D. Speech Recognition

By use of a set of different predefined grammars (with a limited lexicon), which are employed according to the current context of operation, speech recognition is performed through Microsoft’s SAPI SDK.

For a full description of this module, refer to section V.

E. Face Detection

For the robot to be able to maintain eye contact with the user, an algorithm that performs face detection is required. Thus, OpenCV’s [6] face detection algorithm is used as a base for the development of this module.

A more detailed description of this module can be found in [2].

F. People Detection

Omni-directional vision systems are not commonly used for the task at hand, hence no readily available algorithm has been found for this purpose and one had to be developed from scratch. In a general way, this algorithm starts by performing motion detection (through background subtraction) and, by analysis of the image region where movement was detected, it evaluates the region’s geometry by matching it to the geometry features of a person’s legs and feet.

OpenCV’s libraries are widely used for this module’s image processing necessities.

\(^1\) \url{http://www.intelliwizard.com/} (last retrieved in 09/2008)

\(^2\) \url{http://www.microsoft.com/speech/speech2007/default.mspx} (last retrieved in 09/2008)
This module is fully discussed in [2].

G. YARP (inter module/device communication)

The middleware, especially designed for robots, used in this project is known as Yet Another Robot Platform (YARP) [7], and consists of a set of open-source libraries, protocols, and tools which are able to perform communication between different software modules and hardware devices in a decoupled and accessible way.

YARP is designed to be operating system independent, and allows communication between modules/devices that coexist in the same computer (using the operating system’s shared memory) or that are running in different machines on an IP network, through the use of carrier protocols like UDP (for data streaming), TCP (for data that absolutely needs to arrive to the destination, like commands) and multi-cast.

IV. Physical Platform

The robotic platform adopted for the Receptionist robot consists of a modified version of a Nomatic SuperScout II [8], a commercial unicycle robot. This platform holds a set of devices that offer suitable hardware support, such as a Pentium 3 computer running on Linux, wheels motor controllers, odometry board, and an omni-directional vision system. However, several additional devices had to be acquired in order for the robot to perform as intended: A Flybook v33i tablet PC featuring a touch-screen and running on Windows XP; a Philips ToUcam Pro webcam, used for face detection; a Labtec PC Mic 333 microphone; and an Hokuyo-URG-04LX laser range finder.

In Figure 2 a representation of the system architecture, from a physical point of view, is presented: The gray and red rectangles represent the computers running on Linux and Windows XP, respectively; the light blue boxes are the systems modules, which are implemented in either of the available computers; the dashed bordered boxes represent hardware devices (the ones in red bordered boxes are built-in to the Windows XP computer); the black and orange arrows represent the data flow of communication between modules/devices either supported by built-in or YARP connections, respectively; and the yellow boxes represent the processes that implement the Linux Monitor, used for the Linux computer control through the tablet PC.

V. Speech Recognition

This module is responsible for recognizing speech sequences within an expected limited set of context dependent sentences, uttered by any person that the Receptionist Robot might interact with (expectantly, fully grown adults with good knowledge of the English language).

A. SAPI SDK 5.1 speech recognition software

SAPI SDK 5.1 was chosen as the speech recognition toolkit that would implement this module’s speech necessities. This software is application development oriented, providing easy-to-use interfaces to develop Windows applications with speech recognition support. It intends to mask all the complexity associated with the task of speech recognition, providing an already trained, speaker independent, and mature speech recognition engine.

SAPI can use Finite State Grammars as language models, which are configured through XML grammar files. These grammars permit the definition of the phrases the engine is able to recognize, through a sequence of words contained in tags that define: if that set of words has to be expressed at that particular point of the sentence; if one of the set of words in a group is expected at that point; or if a set of words is optional (i.e., this set of words may or may not be uttered by the speaker; the recognizer will reckon the rest of the sentence either way). XML grammar files syntax and lexicon is fully described in SAPI’s help documentation.

In order for an application to be able to deal with new recognition data, SAPI generates a Windows event that includes useful information concerning the recognition result. This includes, but is not restricted to:
The recognition’s successfulness, which can be one of three values: successful recognition, unsuccessful recognition, and interference detected; The kind of interference that was detected, if perceived. It can be one of the following: no signal, noise, too loud, too quiet, too fast, or too slow; The index of the recognized output; The exact sentence that the recognizer believes it was uttered; The confidence level of the recognition, which can be one of three values: high, normal and low;

B. Speech Recognition Control

This module has two possible states of operation: it is either waiting for control commands, or waiting for new sound input, in order to perform a recognition. In figure 3, a Mealy finite state machine [9] representation of this module is presented.

VI. Experimental Results

In order to evaluate SAPI’s recognition robustness, a set of experiments using 6 different speakers were performed. It was asked each speaker to utter the same sequence of sentences in two different scenario, each with a specific goal:

1. This scenario’s aim is to be as close as possible to this module’s intended context of operation, in order to evaluate its robustness in realistic conditions. The hardware configuration that is used is the one available in the Receptionist platform (the tablet PC’s sound card is used to acquire the sound captured by the on board microphone), and the speakers where asked to speak while standing up and about one meter behind the Robot;

2. This scenario is defined as a reference to understand how much of the recognition performance is conditioned by the Receptionist’s context of operation. To do so, a different (less noisy) hardware configuration was employed, using a SilverCrest Bass Vibration Headset and a Toshiba Tecra A3X laptop’s sound card for data acquisition.

Two different language models are used during the Receptionist’s regular operation, one to capture a positive or negative response from the user, and the other to discern to which room the user would like to be guided to. Considering that these grammars feature rather different characteristics, since one of them only has two possible outcomes and short recognizable sentences, and the second is considered more challenging for featuring 25 different possible outcomes and considerably longer recognizable sentences, their recognition performance is evaluated separately and three different sentences to be spoken by the test subjects are defined for each language model.

In the model defined by yes_no.xml case, the sentences (in this case, words) to be spoken by the speakers are “yes”, “no”, and “maybe”. While the first two represent the shortest possible recognizable sentences and cover both possible recognition outcomes, “maybe” is not included in the set of recognizable sentences and is used to evaluate how well the recognizer handles sentences that are not supposed to be recognized. Since it would be impractical to evaluate all 25 destination_rooms.xml model’s possible recognition outcomes, two random outcomes were chosen for evaluation: one is represented with a standard size sentence – “Guide me to room six oh seven” – and the other features the longest recognizable sentence supported by this grammar – “Could you please show me the way to the Evolutive Systems and Biomedical Engineering Lab” – and possibly the most challenging to recognize. This model’s defined third sentence is “lead me to nowhere”, and has the same purpose as the “maybe” utterance in the previous grammar.

In figure 4 the obtained recognition results for each sentence is presented, for both the realistic and reference scenarios.

By analysis of both scenario results it can be concluded that this module’s recognition performance is clearly affected by speech capturing conditions. In the reference conditions scenario case, the recognition rate was of 100%
for all users except one, a result that is distinctly different from the one obtained in the realistic scenario, where the recognition rate is drastically lower, especially in the S4 and S5 sentences case. As for the S3 and S6 sentences, the recognizer clearly has trouble distinguishing a sentence that is not covered by the grammar as unrecognized, this situation is clear in both scenarios, where these results cannot be conclusively compared since they are inconsistent and the number of test samples is relatively small.

VII. On-screen Interface and Speech Syntheses

For the user to be able to communicate with the Receptionist in a non-verbal way, it was necessary to develop a graphical interface, which is displayed and interacted through the tablet PC screen.

As a way of disseminating the interest for science and technology, one of the key aspects of the interface is that it should present as much information as possible regarding the sensors and mechanisms that condition the Robot’s behavior. It is also imperative, for demonstration purposes that the interface features the necessary controls to perform direct commands (e.g., choosing a destination by manually selecting a room).

In order to attain an interface accessible to as many people as possible, an effort was made in developing an interface that respects general usability principles like the ones defined by Jakob Nielsen [10], as well as the interface design principles presented by Bruce Tognazzini’s (both authors are software consultants specialized in user interface usability).

The GUI programming toolkit wxWidgets [4] was employed in this module’s development.

A. Interface Layout

The Interface consists of two equally sized notebook windows (a type of window that features a set of selectable tabs), placed side by side, completely filling the screen environment (figure 5). These notebooks feature the same combination of tabs (implying the duplication of each corresponding panel), except for the “Face” panel (described in section B) which is only presented in the notebook on the left, since all the animation and voice handling mechanism responsible for the face control is implemented in this window’s class.

At the system startup, the default combination of displayed panels is the one presented in figure 5, since these are sufficient for the Receptionist to fully operate in Autonomous mode (which is described in section VIII), and are considered more inexperienced user-oriented.

B. Face Panel

In order for the Robot to present suitable human-robot interaction, it was determined that it would require an animated virtual face, integrated in the graphical interface, capable of expressing emotions (this way the Receptionist would be able, for instance, express joy or sadness, depending on whether a requested task had been performed successfully or not).

Xface [5] was the toolkit employed to implement this virtual face (figure 5). This software integrates Microsoft SAPI 5.1 in order to perform speech synthesis while simultaneously synchronizing the face’s lips according to the spoken phonemes.

To ensure a better human/robot interaction the Receptionist’s face should be able to maintain eye contact with the user, using input data from the Face Detection module. Unfortunately this is not possible since, as studied and concluded in [11], visual perception of images represented in a planar surfaces (in this case the tablet PC screen) remains largely unchanged regardless of the vantage point, resulting in the impression that, if the face looks straight ahead, it will seem it is looking straight at the viewer, independently of their position relative to the screen (for instance, Da Vinci’s Mona Lisa seems to be looking directly at us, regardless of the point of view from which we view the painting). The opposite is also true: if the face is looking anywhere else, the viewer will always feel the face is looking elsewhere. Considering this last statement, the aim of rotating the virtual head in a reactive way according to the user face position is to give the impression that the Receptionist is paying attention to them, inviting them to further interact with the Robot.

In figure 6, an example of the face rotation process is presented, consisting of two rotations of the virtual head model around its center. In this figure’s top left corner, a frame captured by the Robot’s camera that corresponds to the Receptionist face’s point-of-view is presented. Considering the represented referential, the face primarily performs a rotation around the Y axis, followed by a rotation around a vector (dependent on the first rotation) in the XZ plane. The rotation angles are calculated using the user’s face position and manually adjusted coefficients.

To avoid that the Receptionist’s face instantly “looks” at a detected face when, in the previous instant, it was facing the other way, a discrete low pass filter is employed, resulting in smoother and realistic head movements. This filter is employed by use of $1$, where $c$ is a gain that has
been hand adjusted to the value of 0.5.

$$\begin{align*}
\text{NewPosX} &= c \times \text{UserPosX} + (1 - c) \times \text{LastPosX} \\
\text{NewPosY} &= c \times \text{UserPosY} + (1 - c) \times \text{LastPosY}
\end{align*}$$  \hspace{1cm} (1)

C. Dialog Panel

This panel is represented in figure 5, and it features the following components:

- A text control window where a log of the conversation between the user and the Receptionist is maintained. Red text represents the Receptionist speech (in the beginning of each sentence, the emotion expressed by the Receptionist’s face while speaking it is presented between brackets), and the blue text represents the user’s speech lines. This window might be useful in case the user fails to hear/understand what the Robot says.
- A list box where the user’s currently available options of speech are presented, serving as an alternative means of communication with the Receptionist, as well as a reference to what the user can say that will be recognized.
- A button labeled “Submit Answer”, which posts the currently selected option in the list box;
- A check box labeled “Use Speech Recognition” is used to activate or deactivate speech recognition.

D. Command Panel

All available buttons on this panel (figure 7), activate the Receptionist’s Manual operation mode, except the button labeled “Resume Autonomous Mode” which triggers the Autonomous operation mode (see section VIII for the definition of both these modes). The remaining buttons perform as follows:

- “Room” and “Person” buttons – Both trigger a pop-up list that features the available rooms and persons, respectively, which the user can select as a destination;
- “Base” button – Instructs the Robot to go to its default location where it waits for a person to approach, while in Autonomous mode;
- “(Pinpoint)” button – Switches to the Map Panel tab (section F), and activates its “Pinpoint Destination” button;
- “Pause”/“Continue” button – If pressed while its label is “Pause” the Receptionist interrupts its current route, while if the button’s label is “Continue” the Robot proceeds to the last defined destination. This button is generally larger than the others in order to make it more accessible, since it is primarily used to interrupt the current locomotion to a specified goal, making it harder to use the interface;
- “Reset Autonomous Mode” button – Resets the Autonomous mode’s state machine;
- “Return To Base And Turn Off” button – Sets the “Base” position as destination, and, as soon as it arrives, turns off the whole system (including both computers);
- “Turn Off” button – Turns off the whole system;
- “Reboot” button – Reboots the whole system.

E. Room and Person Pop-up Lists

The Room and Person pop-up lists (presented in figure 8) are used to manually define a specific room, or the room where a specific person might be found, as a destination, and are accessible through the “Room” and “Person” buttons in the Commands panel.

These two lists feature a room/person per row and, in the Room list case, two different columns featuring a room’s illustrative image and room code, and the room description; while in the Person list, three columns are presented, holding a persons photography and name, their work phone extension, and the room where they might be found.
At the bottom of these lists two buttons can be found: the “GO!” button that is used for submitting the currently selected destination; the “Cancel” button which hides the this pop-up list window.

**F. Map Panel**

This panel (figure 9) is inspired on today’s GPS navigation devices’ interfaces. It features an image of the environment where the Receptionist can navigate (in the current case, ISR’s 6th floor), where several objects are represented:

- The Receptionist – Represented by an orange circle with a black line segment indicating its orientation;
- The laser sweep – Represented by the area covered with intersecting green lines;
- The person’s position as perceived by the People Detection module – Represented by a blue circle;
- The current destination – Represented by the drawing of a a red “target”;
- Waypoints and trajectory plan – Represented by blue circles and lines, respectively.

This panel also provides two buttons – “Pinpoint Destination” and “Place Robot” – that while selected and upon pressing a location in the map and dragging to select an orientation, submit a destination goal or the Robot’s believed position.

The other available controls are zoom related: with the “Zoom” button one can zoom in and out of the map; with the “x2” and “x4” buttons, the zoom level can be shifted between two and four times the map image’s original size, respectively; while activated, the “Track Robot” button sets the Robot’s current position as the zoom focus point, keeping the Receptionist in the center of the viewable zoomed map; one can manually change the zoom focus point by pressing and dragging the map image, while in zoom mode.

![Fig. 9. Two representations of the “Map” Panel. left – no zoom; right – zoom 4x](image)

**G. Status Panel**

In this panel (figure 7), several system related data is presented, from the Receptionist’s point of view:

- In the center, a representation of the Robot is presented;
- The numbered black circumferences with increasing radii and centered on the Robot represent distance ranges in meters;
- The green area represents the last laser sweep;
- The blue circle represents the person’s position as perceived by the People Detection module;
- The blue vertical and red horizontal arrows, both with origin in the center of the panel, represent the current linear and angular velocities, respectively. These are also numerically displayed in the upper right corner of the panel;
- The drawn representation of a battery represents the tablet PC’s current battery capacity;
- In the bottom left corner, Coordination module’s status are presented: the current mode of operation, the current Autonomous mode’s active state, and the last transition that led to the referred state.

**H. Cams Panel**

This panel (figure 7) exhibits, almost in real time, the images captured by both cameras that are featured in the Receptionist’s platform (the one present in the omnidirectional vision system – bottommost image – and the one used for face detection – topmost image –, see section IV for details on these cameras).

On the images captured by the face detection camera, the detected faces are encircled by a red circumference with the same radius as the detected face.

**I. Random Expression generator**

A person’s face, even when it is not speaking or expressing any emotion in particular, is not a static element “carved like stone” – there are always involuntary expressions present. While playing a SMIL-script, Xface’s engine introduces random head movements and blinking, but when no animation is being played the face is static.

To avoid the referred situation, the ExpressionGenerator process was developed. This process randomly submits SMIL-scripts corresponding to expression animations, using the C programming language’s pseudo random number generator function (rand) where the CPU time is used as the first generated number’s seed, and the previously generated numbers are used as seeds for the following generation.

**VIII. Coordination**

Through communication with all other system’s modules but the Face Detection module, the Coordination module is responsible for controlling the Receptionist’s overall behavior.

It has been decided that the Receptionist robot should be able to function in two different modes of operation:

- The Autonomous mode where the Robot should behave as a receptionist, by autonomously performing as mentioned in section II, where the tasks that the Receptionist has to execute are described. While in this mode, the user can interact with the Robot by speech or through the Dialog panel (section C).
• The Manual mode should be considered as a method of demonstrating particular Robot features, providing a way for the Robot to perform direct instructions. In this mode, the Robot is static and waiting for any manual commands submitted through the Command or Map panels (described in sections D and F respectively).

A. Behavior Model

Considering the Receptionist’s requirements for the Autonomous mode of operation, a Finite State Machine (FSM) [9] was considered the best choice as the model of behavior to be used for the Receptionist’s automation needs.

UML StateWisard\(^4\) was the chosen toolkit to implement this module’s FSM. This toolkit acts like a Visual C++ add-in and provides a Unified Modeling Language (UML) statecharts [12] programming mechanism.

It integrates two modeling tools, accessible through the Visual C++ environment. The first is the State Tree (figure 10 in the right), where each of the system’s states is represented with its associated child states, transitions and Entry and Exit actions branching down from it. The second tool is the State Chart, which presents a graphical representation of the system’s states, with child states contained inside their parent states, and the transitions connecting these states.

![Fig. 10. Representation example of the StateWizard's State Chart (in the left) and State Tree (in the right) for a sample applications](image)

The designed UML statechart model features three hierarchy levels with specific conceptual significance.

B. Modes of Operation Hierarchy Layer

The top level layer implements the Receptionists two possible modes of operation, modeled by two states (AutonomousMode and ManualMode, and features a third state (Booting) which is exited as soon as all system modules’ YARP ports are connected, meaning that the Receptionist is ready to operate (figure 11).

Upon exiting the Booting state, the state machine transits to the ManualMode state. In this state, all manual commands emitted by the On-screen Interface and Speech Syntheses module (section VII) are handled by a set of internal transitions.

The behavior performed by the Receptionist robot while in the AutonomousMode state is modeled by a lower level sub state machine.

C. Autonomous Behavior Hierarchy Layer

At this hierarchy level, all individual behaviors that make up the receptionist overall behavior are modeled by states, and transitions between these states are triggered by external events that are fired upon arrival of specific data through this module’s ports. In figure 12 this layer’s statechart is presented.

D. User Feedback Confirmation Hierarchy Layer

In order to handle the uncertainty associated with the Speech Recognition module’s recognized speech, required whenever user feedback is requested by the Receptionist, the general purpose state machine represented in figure 13 was developed. This state machine purpose is to ask the user for a confirmation in case the recognizer is not sure of the speech that was comprehended, in which case the Speech Recognition module returns a confidence value less than high.

Since the confirmation procedure is meant to be employed in several Autonomous behavior statechart’s states, its design was developed so that it could be reused in each of the these states as a sub state machine. Even though UML statechart models do not support multi parenthood, and so the same state machine cannot directly be enclosed in more than one higher level state, this model has been developed in order for the same state structure, transition events, and action routines to be reused.

IX. overall system performance analysis

In order to evaluate how well the Receptionist performs both tasks to which it was designed (to function as a receptionist on the floor where it is stationed, and to serve as a demonstration platform of its robotic capacities), two different test scenarios were considered, where three users that fit the profile of the Robot’s target audience (section I) are requested to interact with the platform and perform a set of predefined tasks.

The time it takes for each test subject to perform each required step to achieve the desired goals is also registered (Table I), and compared with the set of reference values obtained by an experienced user (that knows the platform and has optimized his interaction with it), while performing the same task.

\(^4\) http://www.intelliwizard.com/ (last retrieved in 09/2008)
A. Receptionist Test Scenario

This test scenario aims at evaluating the Robot’s capacity to function as an interactive identity, which purpose is to address incoming persons and to serve them.

The Receptionist is initially stationed at its “base” position, and the test subjects, which do not have any previous knowledge of the Receptionist behavior or of its interface (besides the fact that the Robot recognizes speech commands and features a touch screen interface), are asked to approach the Receptionist and interact with it in order to request that it takes them to the toilet.

B. Demonstration Test Scenario

This test scenario focuses on evaluating how intuitive and accessible it is to issue direct commands to the Robot through its on-screen interface, and how well it performs the requested tasks.

Before the test subjects know the tasks they will be asked to perform, they are submitted to a comprehensive explanation/demonstration concerning the Receptionist’s overall capacities and its on-screen interface functionalities.

In this scenario, the Receptionist is booted while stationed at the top left corridor’s corner of the map (facing down). Considering this initial state of events, the user is asked to perform the following tasks:

- Indicate to the Robot its current position correctly (at system startup, the Receptionist’s default position is “base”);
- Instruct the Robot to go to the lower left corridor’s corner of the map and follow it;
- As soon as the Robot passes the 6.09 room door, instruct the robot to stop;
- Instruct the Robot to go to a specific person’s office.
\begin{table}
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Exp. Scenario & Step & Ref. & subjects 1, 2, and 3 \\
\hline
Receptionist & St1 & 12 & 42 & 19 & * \\
& St2 & 9 & 44 & 51 & 51 \\
& St3 & 45 & 46 & 45 & 41 \\
\hline
Demonstration & St1 & 6 & 50 & 41 & 105 \\
& St2 & 5 & 32 & 55 & 25 \\
& St3 & 22 & 28 & 24 & 25 \\
& St4 & 9 & 38 & 48 & 31 \\
& St5 & 69 & 71 & 71 & 73 \\
\hline
\end{tabular}
\caption{System testing time values. All values are represented in seconds. Step executed condition legend: Receptionist scenario: St1 – “yes answer successfully submitted when asked if assistance is required; St2 – “toilet” specification as a destination accepted; St3 – Destination reached. Demonstration scenario: St1 – Robot’s position specified correctly; St2 – Pin-pointed destination submitted; St3 – Robot’s course interrupted; St4 – destination as a person submitted; St5 – arrived to destination}
\end{table}

In the Receptionist experiment scenario, the test subjects did not react to the on-screen interface quite as expected, subjects 1 and 3 generally seemed to overlook it since the first one initially ignored the options of speech available and instantly requested to be lead to the toilet, while subject 3 ignored both what the Robot said and the information available in the Dialog panel, using directly the “room” button in the Command panel (being this the reason why subject’s 3 St1 time is represented by a ’*’ – St2 time value represents the time it took to submit the destination using the alternative method); also, Subjects 1 and 2 were forced to confirm the requested destination. This results reveal that, as the subjects later confirmed: the screen size revealed itself too small to fully catch the users full attention, the displayed available options are not clearly highlighted as such, and the synthesized speech is not completely clear for relatively long sentences.

Where it concerns the Demonstration scenario, all subjects revealed some difficulty in understanding how to correctly define the Robot’s position, specially it’s orientation, and none of them thought to use the zoom functionality to assist the robots positioning. Another situation that revealed itself to be troublesome consisted on using the graphical interface while the robot was moving. Nevertheless, the test subjects reacted well to the rest of this scenarios steps, and confirmed that the next time they performed a similar task they would be surer on how to use the interface.

In other more casual experiments, the overall system revealed it self, to some point, to perform as expected. However, in some situations, the Robot demonstrated unpredicted stability issues covered in [1].

\section{Conclusions}

All predefined requirements to successfully implement this project’s autonomous robot were achieved. A conception plan of the system’s software and hardware architectures was elaborated and pursued. Several software packages were adapted and employed in the Robot’s architecture, resulting in the Receptionist’s Human-Robot interface capacities and a coordination solution.

The Speech Recognition module was successfully employed, but did not reveal it self to be completely reliable.

The On-screen Interface and Speech Synthesis module showed it self to perform as expected, excluding in particular situations where usability issues are clear and should be attended to.

All modules have been successfully integrated through the implemented Coordination module, resulting in an overall working system, that nevertheless demonstrates punctual instability situations that should be resolved in the future.

\begin{thebibliography}{10}
\end{thebibliography}