El siguiente artículo ha sido aceptado para ser publicado en el vol. 30, no. 1 de la revista Ciencia e Ingenieria Neogranadina. Esta versión es preliminar y puede contener algunos errores.

Cómo citar:

Muñoz Peña, K., & Bacca Cortes, B. (2019). GUI3DXBot: An Interactive Software Tool for a Tour-Guide Mobile Robot. Ciencia E Ingenieria Neogranadina, 30(1). https://doi.org/10.18359/rcin.3644

GUI3DXBot: An Interactive Software Tool for a Tour-Guide Mobile Robot

GUI3DXBot: Una herramienta software interactiva para un robot móvil guía

Kevin Muñoz Peña[*]

Bladimir Bacca Cortes[**]

Fecha de recepción: 13 de septiembre de 2018

Fecha de aprobación: 6 de noviembre de 2019

Cómo citar:

Muñoz Peña, K., & Bacca Cortes, B. (2019). GUI3DXBot: An Interactive Software Tool for a Tour-Guide Mobile Robot. Ciencia E Ingenieria Neogranadina, 30(1). https://doi.org/10.18359/rcin.3644

[*] Universidad del Valle. E-mail john.k.munoz@correounivalle.edu.co. ORCID: 0000-0002-4280-6358

[**] Universidad del Valle. E-mail bladimir.bacca@correounivalle.edu.co. ORCID: 0000-0003-0113-4134

RESUMEN

Actualmente, los robots móviles inician a aparecer en lugares públicos. Para realizar estas tareas adecuadamente, los robots móviles deben interactuar con humanos. Este artículo presenta GUI3DXBot, un aplicativo para un robot móvil guía. Este artículo se enfoca en el desarrollo de los diferentes módulos software necesarios para guiar a usuarios en un edificio de oficinas. GUI3DXBot es una aplicación cliente-servidor, donde el lado del servidor se ejecuta en el robot, y el lado del cliente se ejecuta en una tableta de 10 pulgadas Android. El lado servidor de GUI3DXBot está a cargo de la percepción, localización-mapeo y planificación de rutas. El lado cliente de GUI3DXBot implementa la interfaz humano-robot que permite a los usuarios solicitar-cancelar un servicio de guía, mostrar la localización del robot en el mapa, interactuar con los usuarios, y tele-operar el robot en caso de emergencia. Las contribuciones de este artículo son dos: se propone un diseño de módulos software para guiar a usuarios en un edificio de oficinas, y que todo el sistema robótico está bien integrado y completamente probado. GUI3DXBot fue validada usando pruebas de integración y de campo. Las pruebas de campo fueron realizadas en un periodo de 2 semanas, y una encuesta a los usuarios fue llevada a cabo. Los resultados de la encuesta mostraron que los usuarios piensan que GUI3DXBot es amigable e intuitiva, la selección de metas fue fácil, pudieron entender los mensajes de interacción, 90% de los usuarios encontraron útil el ícono del robot sobre el mapa, encontraron útil dibujar la ruta planeada en el mapa, 90% de los usuarios encontraron útil la vista local-global del mapa, y la experiencia de guía fue muy satisfactoria (70%) y satisfactoria (30%).

Palabras clave: Robot guía, Robot móvil, Interacción humano-robot, Robótica de servicio

ABSTRACT

Nowadays, mobile robots begin to appear in public places. To do these tasks properly, mobile robots must interact with humans. This paper presents the development of GUI3DXBot, a software tool for a tour-guide mobile robot. The paper focuses on the development of different software modules needed to guide users in an office building. In this context, GUI3DXBot is a server-client application, where the server side runs into the robot, and the client side runs into a 10-inch Android tablet. The GUI3DXBot server side is in charge of performing the perception, localization-mapping, and path planning tasks. The GUI3DXBot client side implements the human-robot interface that allows users requesting-canceling a tour-guide service, showing robot localization in the map, interacting with users, and tele-operating the robot in case of emergency. The contributions of this paper are twofold: it proposes a software modules design to guide users in an office building, and the whole robot system was well integrated and fully tested. GUI3DXBot were tested using software integration and field tests. The field tests were performed over a two-week period, and a survey to users was conducted. The survey results show that users think GUI3DXBot is friendly and intuitive, the goal selection was very easy, the interactive messages were very easy to understand, 90% of users found useful the robot icon on the map, users found useful drawing the path on the map, 90% of users found useful the local-global map view, and the guidance experience was very satisfactory (70%) and satisfactory (30%).

Keywords: Tour-guide robot, Mobile robot, Human-robot interaction, Service robotics.

INTRODUCTION

People guidance is a common service where mobile robots have begun to appear. This kind of services is performed in public places such as hospitals, hotels, expositions, supermarkets, office buildings and museums [1]. For example, a mobile robot can be used to guide patients to the examine locations or doctor rooms in a hospital; in a hotel mobile robots can guide customers to their rooms and to carry their luggage; mobile robots can be used for newest customers can find their products in supermarkets; and tour-guide robots can be used to reduce the risk of missing connection in airports [2]. Using mobile robots to perform people guidance tasks has the following advantages: people guidance and information providing could be repetitive tasks, then employees can be in charge of more important responsibilities; mobile robots can load more weight without health risk; employees can concentrate on your jobs without interruption; mobile robots can be used as shared surveillance systems meanwhile they perform the people guidance task; and mobile robots can be used as tele-presence platforms.

Many components are important to perform a people guidance task. For instance, the ability to self-localize and navigate in a dynamic environment; this means the mobile robot has to use its own sensors to estimate its position and guarantee a safe guidance to the people's goal. Also, a human-robot interface must to provide enough information to people, and collect feedback data from users. And, an emergency system must be aware of situations such as unreachable goals, battery level and robot position lost.

This paper focuses in the development of GUI3DXBot [3] software tool, and all the software modules needed to guide users in an office building. Specifically, providing a guidance service in the Electrical and Electronic Engineering School of Universidad del Valle such that newest users can find professor's office, electronics labs, and research labs. To do so, GUI3DXBot was conceived as a client / server application, the server side is in charge of performing the perception, localization, mapping, and path planning tasks. These tasks use as a main sensor the LMS200 laser range finder, and they were implemented on the ROS framework [4]. The GUI3DXBot client side implements the human-robot interface in order to bring a friendly, and interactive user interface. The GUI3DXBot user interface allows users requesting/canceling a tour-guide service, showing the current robot localization in the environment map, interacting with users using a robot avatar, communicating with the GUI3DXBot server side, and tele-operating the mobile robot in case of emergency.

This paper is structured as follows: the related works are presented in Section 1, the experimental platform is detailed in Section 1.1, the GUI3DXBot server side is described in Section 2, the corresponding client side is described in Section 3, Section 4 shows the results obtained, and conclusions are detailed in Section 5.

1. RELATED WORKS

Nowadays, there are many academic and commercial implementations of tour-guide solutions, Table 1 shows a revision of previous works related with this paper. Table 1 compares each work considering the sensors, the type of map used, the localization method, the obstacle avoidance method, and the human-robot interface (HRI) used; also, the type of environment of deployment, and the main application.

Table 1. State of the art summary of tour-guide solutions.

Ref.	Sensors	Map	Loc. Method	Obst. Av.	HRI	Env.	Application
[1]	Camera, LRF	Sub-mapping	Sub-map - SLAM	Pedestrian based	Touch-screen	Indoors	Reception, guidance
[2]	RGBD, LRF	Occupancy grid	SLAM, CTMap	- - -	Touch-screen, bar code reader	Indoors	Airport guidance
[5]	Camera and LRF	Occupancy grid	Particle Filter	µDWA	Touch-screen	Indoors	Museum tour guide
[6]	Camera	Topological	Bayes	Reactive algorithm	Push buttons	Indoors	Museum tour guide
[7]	RGBD, LRF	Occupancy grid	Particle Filter	DWA	Touch-screen	Indoors	Office tour guide
[8]	Smartphone-camera, Ultrasonic	Topological	QR recognition	Reactive algorithm	Mobile app Touch-screen	Indoors	Exhibition tour guide
[9]	Camera and LRF	Sub-mapping	Sub-map - SLAM	- - -	Voice commands	Indoors	Campus tour guide
[10]	LRF	Metric map	SLAM-EKF	- - -	Touch-screen, voice comm., WEB	Indoors	Exhibition tour guide
[11]	LRF, GPS	Occupancy grid	Particle filter	Reactive algorithm	Touch-screen	Outdoors	Pedestrian assistant
[12]	Sonar, RFID	Topological	RFID recognition	Reactive algorithm	Touch-screen	Indoors	Exhibition tour guide
[13]	RFID, LRF	Metric map	EKF	Fuzzy logic	Touch-screen, Mic.	Indoors	Exhibition tour guide
[14]	Camera, LRF	Occupancy grid	Particle filter	Potential fields	Voice commands	Indoors	Museum tour guide
[15]	LRF, omni-camera	Occupancy grid	Particle filter	VFH	Touch-screen, eyes track.	Indoors	Shopping guide
[16]	Camera, LRF	Occupancy quadtree	Particle filter	DWA	Voice commands, Mobile app	Indoors	Shopping guide
[17]	RFID, Camera, LRF	Occupancy grid	Particle filter	DWA	Touch-screen	Indoors	Campus tour guide
[18]	RFID, LRF	Metric map	LSM-RFID detection	- - -	- - -	Indoors	Exhibition tour guide

Observing Table 1, the main sensor used is the laser range finder (LRF) and different type of cameras such as monocular, RGBD and omnidirectional. However, special localization sensors also are used such as RFID [12], [13], [17], [18] and QR codes [8]. These kind of sensors are used in order to obtain an initial localization hypothesis which is refined by LRF or vision sensors.

Other important part of these tour-guide solutions is the map used, since depending on it, the tour-guide robot can achieve a proper environmental representation, and this map can be updated over long term usage. Then, Table 1 shows that, in decreasing order of importance, occupancy grid, topological, sub-mapping and metric maps are preferred for tour-guide solution implementations.

The map building and localization methods are crucial to implement tour-guide solutions. Table 1 shows that particle filter based methods are the SLAM (Simultaneous Localization and Mapping) approach which better suits in this kind of applications. This is because particle filters perform better in populated environments, which are prone of localization errors due occlusions, and large changes in the illumination at the environment. However, in [2], [7], [14] and [16] the SLAM approach is modified in order to include social awareness for the robot navigation. This property is important in order to compute de motion of robot, and capture the correct motion patterns of pedestrians. Long-term navigation in real environments is another important challenge of tour-guide robots. This aspect is considered in [5], [9], [11] and [15], where the SLAM approach and the planning method were implemented considering full time job with minimal human intervention.

Sensors and technical issues related with SLAM are important, but if all the information generated by these methods cannot be properly communicated to humans, the tour-guidance service cannot be accomplished. For this reason, HRI is a very important part of any tour-guide robotic solution. Table 1 shows how HRI has evolved over many years, starting with simple push buttons [6]; then using virtual maps [10]; afterwards, touch-screen interfaces as reported in most works in Table 1; next, adding the capability of understand simple and complex voice commands as proposed in [9], [10], [14] and [16]; and finally, including special sensors which depend on the application as in [2], as well as detecting pedestrians intensions using eye tracking [15].

In this paper, the GUI3DXBot software tool is presented. GUI3DXBot controls a tour-guide robot with the aim of guiding newest visitors to the Electrical and Electronic Engineering School at the Universidad del Valle. Then, comparing the works presented in Table 1 and GUI3DXBot, it is important to highlight the following remarks:

- GUI3DXBot uses a LRF as main sensor to build the map of the environment, and localize the mobile robot.
- GUI3DXBot uses Gmapping [19] in order to build an occupancy grid map and using a particle filter as localization method.
- GUI3DXBot uses the trajectory rollout algorithm for obstacle avoidance.
- GUI3DXBot uses a touch-screen as human-robot interface. Using this interface, users about emergency situations (unreachable goals, low battery level, or lost robot position), displaying a friendly user interface to select goals, showing to users the path to their goals, and its real-time evolution on a map of the environment.
- GUI3DXBot works in indoors environments as an office tour guide robot application.

Then, this paper has two main contributions: first, it proposes a software module design to guide users in an office building, since most works presented in Table 1 (except [2], [9] and [11], and [15]) have custom software designs without considering standard middleware as ROS [4]. And second, the whole robot system is well integrated and fully tested.

1.1 SYSTEM CONFIGURATION

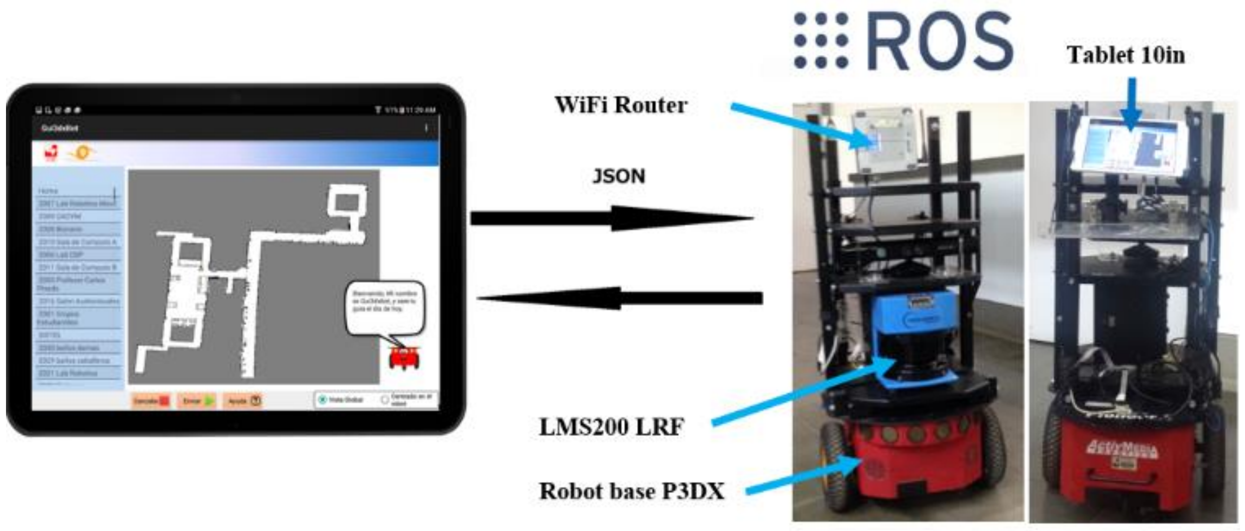


Fig. 1. System configuration for the GUI3DXBot software tool.

Fig. 1 shows the hardware components needed to run the GUI3DXBot software tool. The right side of Fig. 1 shows the Pioneer 3DX robot base, the LMS200 LRF as the main mapping and obstacle avoidance sensor, the WiFi router as the remote communication interface, and the 10in Android device which displays the user interface depicted in the left side of Fig. 1. The Pioneer 3DX robot base was built by ActivMedia robotics, and it has the following technical specifications: differential drive mobile robot, two incremental encoders of 500 ticks/rev, eight frontal sonar sensors, one on-board computer running Ubuntu 14.04 and ROS Hydro, a three battery pack of 12V / 7Ah, and a metal based support to hold sensors on top. The communication protocol between the mobile robot and the Android device is performed using JSON objects through Web-sockets, and using the WiFi router as communication interface.

2. GUI3DXBot SERVER SIDE SOFTWARE DEVELOPMENT

Mobile robot localization and mapping are essential software modules used by service robots. To do so, in the context of office tour-guide robots, this section describes the development of the GUI3DXBot server side software which includes the global localization, the communication with the client side, the path planning, the local and global navigation tasks.

2.1 DESIGN REQUIREMENTS

GUI3DXBot was developed using the RUP methodology [20]. Only the functional, non-functional requirements, and class diagrams are presented due to space limitations. The GUI3DXBot server side functional and non-functional requirements are described as follows:

- *Functional requirements*: the GUI3DXBot server side must be able of building a map of the environment using range sensors, localizing itself in the map, planning a path to the goal requested by users, performing local planning in case of presence of obstacles, navigating to the home location at the end of the office tour-guide service, and issuing to users alerts in case of emergency situations.
- *Non-functional requirements*: the GUI3DXBot server side works on Ubuntu 12.04, it was developed on ROS Hydro, the robot base is a Pioneer 3DX, and the range sensor was the SICK LMS200.

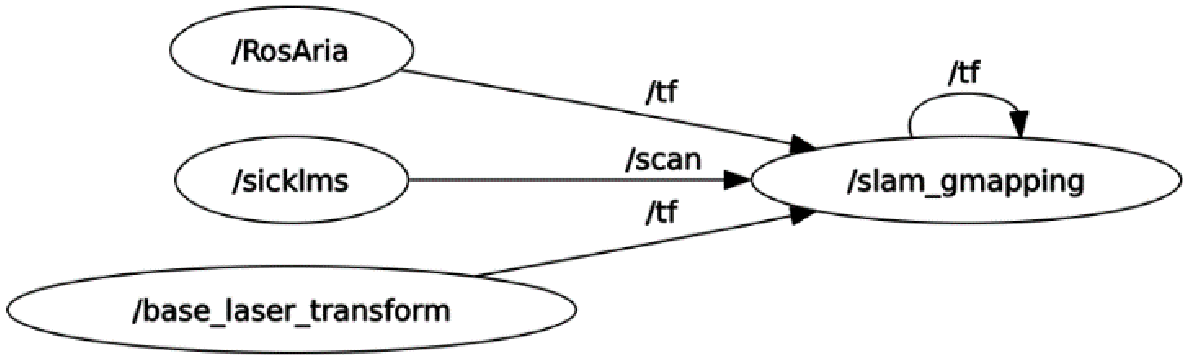
2.2 MOBILE ROBOT MAPPING, LOCALIZATION, LOCAL AND GLOBAL NAVIGATION

The functional requirements described above were implemented using a set of software modules which run in the GUI3DXBot server side. Then, the GUI3DXBot server side is in charge of the localization, local and global navigation tasks, as well as communicating with the HRI.

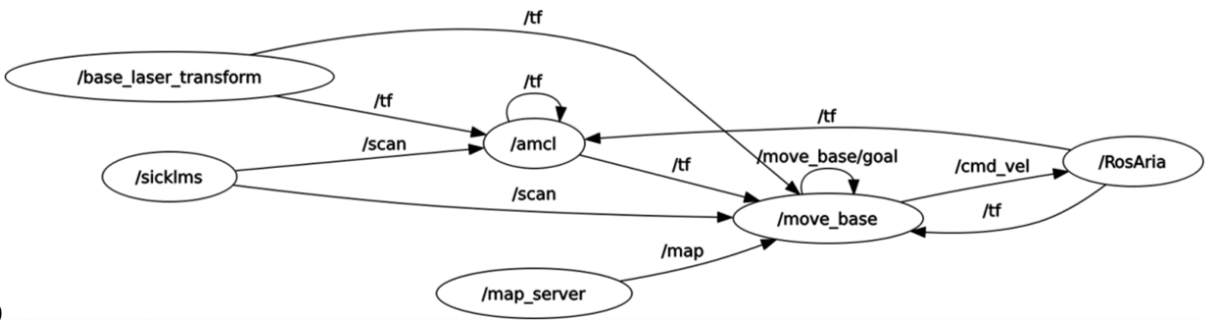
In order to start offering guiding services using a mobile platform, a previous mandatory step is needed: mapping the environment. To do so, Fig. 2a shows the computational graph which represents the mapping process. This figure, as well as Fig. 2b, are common ways to represent the algorithm implemented when ROS is used. Then, the mapping process was performed using the *GMapping* package, which requires the odometry information provided by the mobile robot through the *RosAria* package, the range data provided by the laser sensor, and the transformation between the laser sensor, and the mobile robot platform. The mapping process is performed once, unless the environment structure changes. Then, this map is further used for the people guidance service.

The people guidance service which implements the GUI3DXBot software tool is constantly localizing the robot in the map, planning the path to the goal requested by users, and avoiding obstacles. Fig. 2b shows the computational graph that relates the software modules in charge of these tasks. As can be observed in the Fig. 2b, the *move_base* node is an important part of the robot navigation software, it requires the odometry readings provided by the *RosAria* package, the map of the environment, the current sensor readings, and the current robot position. The *move_base* node is in charge of robot navigation, which includes local and global path planning, however, these features are useless without the robot current position.

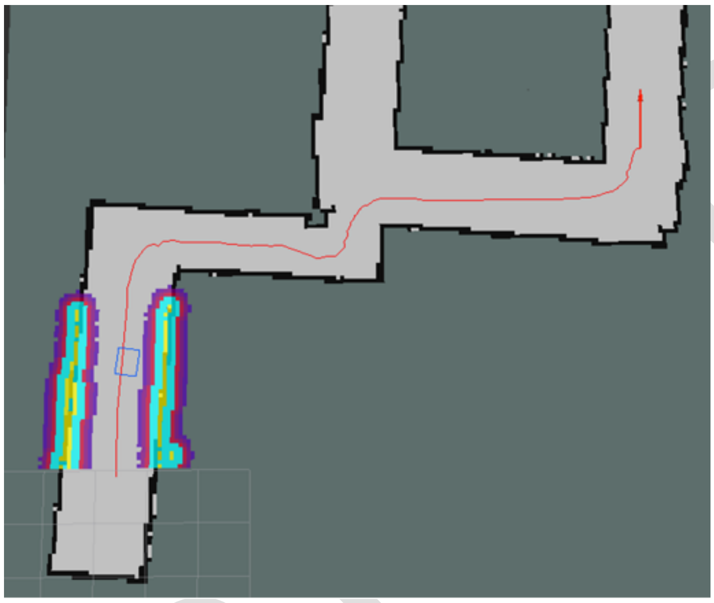
Then, the first thing the GUI3DXBot software tool must to do is to localize the mobile platform. To do so, the package AMCL is used; it implements the adaptive Monte Carlo Localization approach [4], which requires the robot odometry readings provided by the *RosAria* package, the laser scan readings provided by the *SICKLms* package, and the relative transformation between the laser sensor and the mobile base to localize the mobile robot.



a)



b)



c)

Fig. 2. a) Computational graph for the mapping process. b) Computation graph for the localization and navigation process. c) Cost map for local path planning and global path planning.

In second place, the GUI3DXBot software tool must plan the robot path from the current position to the goal requested by users. To do this, the package *Navfn* [4] was used as part of the node *move_base*. Once the user's goal is known, this information as well as the current robot positon are introduced to the *Navfn* package to compute the global path. The global path is computed in the *Navfn* package using the Dijstra algorithm. An example of this global path is shown in the Fig. 2c.

People guidance service is commonly deployed in public spaces, and obstacle avoidance is essential to guarantee a safe route to the user's goal. Then, the global path planning could be modified locally when the mobile robot faces the obstacles. To implement this feature, the *move_base* node has another important package called *costmap_2d* which implements the trajectory rollout algorithm [21] in order to compute the local robot trajectory considering the obstacles in the environment. Fig. 2c shows an example of the local cost map. Using this algorithm, at each robot position different trajectories are evaluated considering the obstacles distribution, and the robot kinematic model. Each trajectory is evaluated using a cost function which includes the sensor readings, the local static map, the inflated obstacles relative position with respect to the robot, the robot footprint, the size of the local map, between others. At the end, the low cost trajectory is selected.

As can be observed in Fig 2a and 2b, the graph nodes define a server-side software module design which can be replicated in most platforms compatible with ROS. In this work, this design is used to conduct people guiding service in an office building, but it can be adapted to many other applications. In addition, all the server-side software module was implemented using new and existent ROS functionalities. The new ROS functionalities include: transform nodes, odometry sensor handler, robot base handler, and all sensor topics (laser and point cloud). Finally, the server-side software module has no GUI, then all parameters are configured either by command line or using the GUI3DXBot user interface.

3. GUI3DXBot USER INTERFACE

Section 2 described the GUI3DXBot server side, which handles information that mobile robots can understand. This section shows the software modules of GUI3DXBot user interface, which is used to send user requests, and to receive feedback and status information from server side.

3.1 DESIGN REQUIREMENTS

The GUI3DXBot user interface was also developed using the RUP methodology [20]. Only the functional, non-functional requirements, and class diagrams are presented due to space limitations. The GUI3DXBot client side functional and non-functional requirements are described as follows:

- *Functional requirements*: the GUI3DXBot user interface must show the current map to users, showing all available goals to users, receiving the user's goal, showing the path to the user's goal on the map while the guidance service is running, showing the current robot position on the map while the guidance service is running, showing the local or global view of the environment, receiving the user's cancel order if available, showing status information using a robot avatar, and showing emergency information using a robot avatar.
- *Non-functional requirements*: the mobile device minimum S.O. version is 5.0.2, the server side was implemented on ROS Hydro, and the mobile robot platform was a Pioneer 3DX.

3.2 GUI3DXBOT USER INTERFACE DEVELOPMENT

The functional requirements described above were implemented as software modules of the GUI3DXBot HRI. This HRI interchanges messages between the GUI3DXBot server side (Action server), and the GUI3DXBot user interface (Action client). All these messages are shown in Fig. 3a. As described in Section 2, GUI3DXBot server side needs to know the requested user goal, or the user start/cancelation order to begin with the planning and navigation process. In turn, the HRI needs to know status information such as the current robot position, the current local / global map, or the list of different goals to go; in addition, the HRI needs to know feedback information such as the messages to display through the robot avatar (start / cancel / successful end of a guidance mission), and the alerts of emergency situations (low battery, unreachable goals).

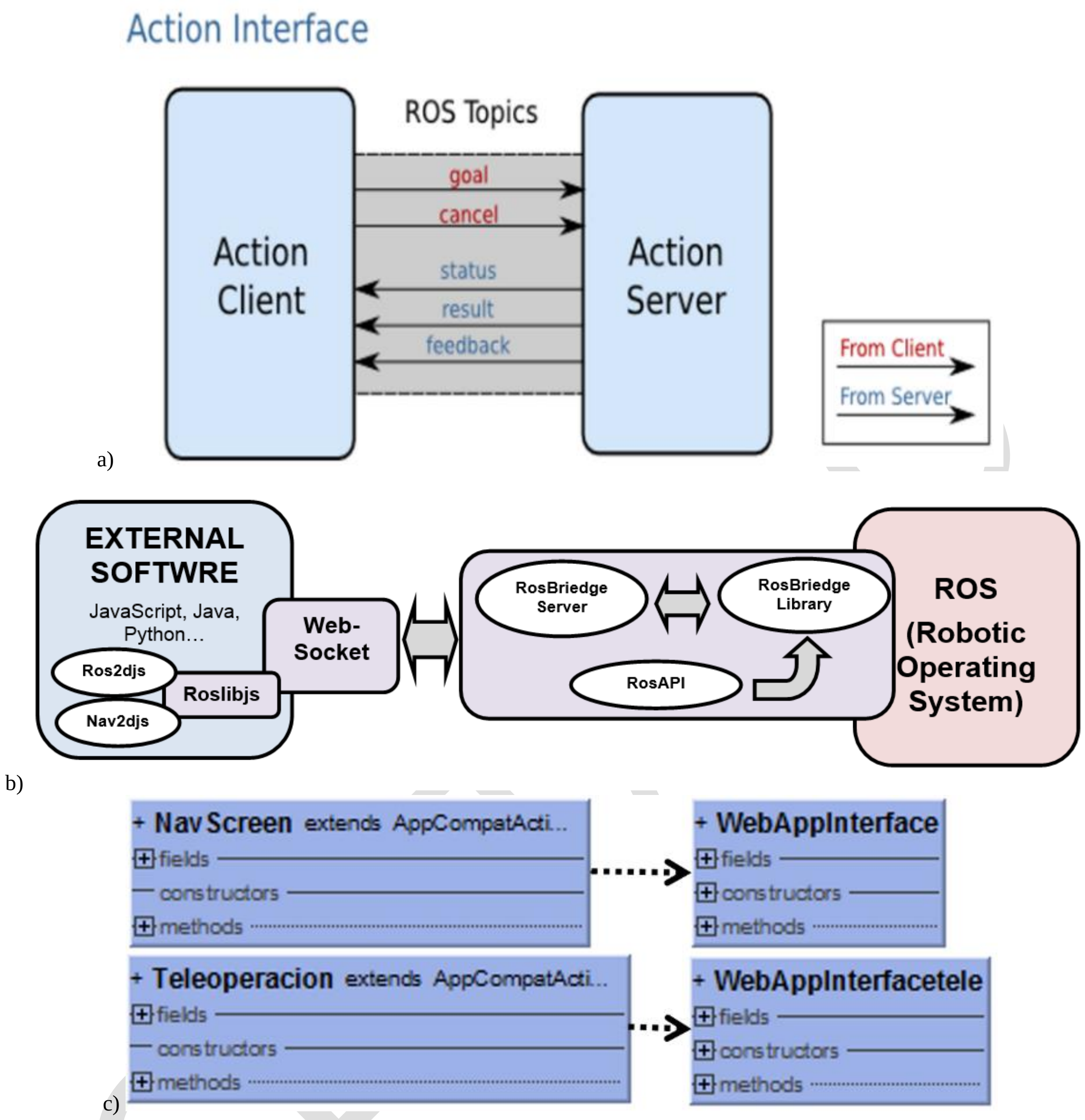
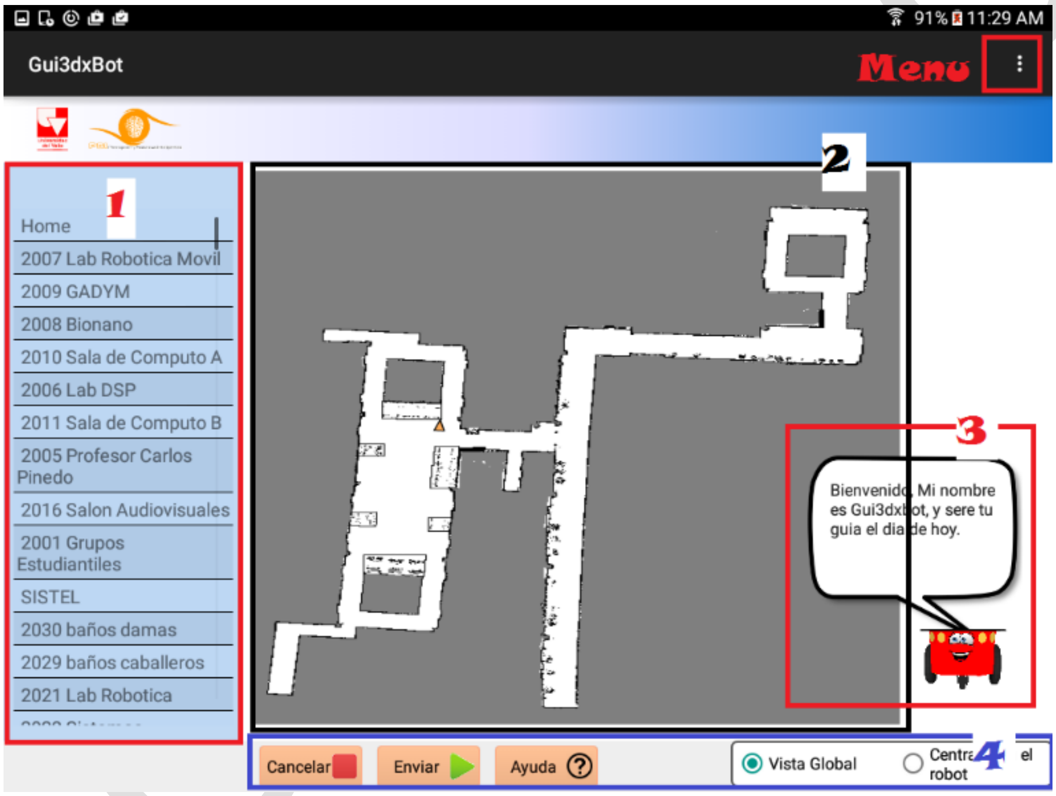


Fig. 3. a) Communication structure of the action interface between move_base node and the GUI3DXBot user interface [4]. b) Conceptual diagram of the communication structure software. c) Class diagram of the GUI3DXBot user interface.

All this information is interchanged using JSON objects, which are sent using a web-socket connection between the GUI3DXBot server and client side. This is depicted in Fig. 3b. Fig. 3b also shows the server-side technologies involved in this communication namely: *RosBridge Library*, which is responsible of taking JSON objects and sending them to ROS, and vice-versa; *RosBridge Server* which implements the web-socket connection using the *RosBridge* protocol [22]; and *RosBridge API* which is uses by clients to access reserved services of ROS libraries. On the client-side, *Roslibjs* implements the *RosBridge* protocol in order to provide the ROS functionalities implemented in this work such as: topic publication /subscription, calling ROS services, action servers, performing transformation, etc. Internally, the HRI uses *Ros2djs* and *Nav2djs* to show the environment map, the robot position, the robot path and the robot goals. These functionalities were implemented in the GUI3DXBot user interface.

Fig. 3c shows the class diagram of the software module that implements the GUI3DXBot user interface. This class diagram shows two Android activities namely: the navigation or main activity, and the teleoperation activity. The *NavScreen* class implements all the functional requirements described in Section 3.1. And, the Teleoperation class implements the robot tele-operation functionality, which is used in case the mobile robot requires human assistance. In this case, the GUI3DXBot user interface requests a password, then the mobile robot can be moved manually. Both activities use the *WebAppInterface*, which is used as a bridge between the GUI3DXBot user interface, JavaScript and Android. In this way, the *WebSocket* communication between the GUI3DXBot user interface and server side can be established.



Bienvenido Mi nombre es Gui3dxbot, y sere tu guia el dia de hoy.

Fig. 4. Navigation activity (NavScreen) of GUI3DXBot

The GUI3DXBot user interface has two activities namely: the navigation activity (*NavScreen* class) which is shown in Fig. 4, and the tele-operation activity shown in Fig. 5. The navigation activity is divided in four parts namely: goals list (number 1 in Fig. 4), the environment map (number 2 in Fig. 4), the robot avatar (number 3 in Fig. 4), and the controllers bar (number 4 in Fig. 4). The four parts of this activity contain a total of 8 graphical components and their corresponding event handlers which were developed to implement the GUI3DXBot user interface.

The scrollable goals list contains all the places where the mobile robot can guide people. Here, the user just selects the goal location and the process of guiding starts. The environment map is continuously received from the GUI3DXBot server side, and it also shows the mobile robot trajectory, the current robot location, and the goals location.

The robot avatar is used to interact with users, through this avatar GUI3DXBot displays and issue messages using the sound synthesizer of Android. These messages include: asking users if they want the guiding service, issuing alerts of emergency situations (low battery, unreachable goals), and notifying users the goal requested was successfully achieved.

Using the controllers bar, users are able to start a guidance service ("Enviar" button), users can cancel a guidance service at any moment ("Cancelar" button), users can read how the guidance service works ("Ayuda" button), and users can change the environment map view to global or local at any moment of the trajectory.

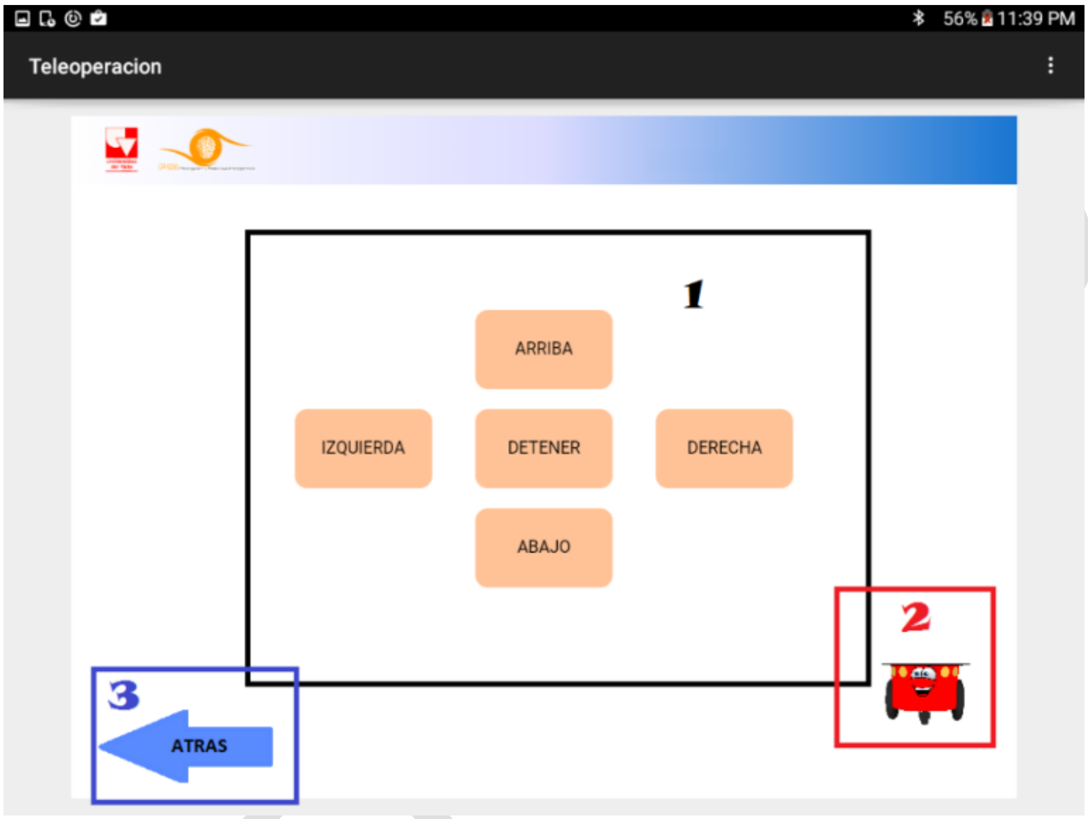


Fig. 5. Tele-operation activity of GUI3DXBot.

The tele-operation activity is show in Fig. 5, and it is activated using the Menu, which is placed at the right top of GU3DXBot GUI. The Menu also allows to update the goals list in case the environment has changed. The tele-operation activity is accessible once the user password is validated. This Android activity is composed by 3 parts namely: the tele-operation controls (number 1 in Fig. 5), the robot avatar (number 2 in Fig. 5), and the back button (number 3 in Fig. 5). The three parts of this activity contain a total of 7 graphical components and their corresponding event handlers which were developed to implement the GUI3DXBot user interface.

The tele-operation controls allow users to command the motions of the mobile robot. The robot avatar displays and issues sound alerts while the users is commanding the robot motion. And, the back button returns to the navigation activity. It is worth noting that the tele-operation activity is executed in case of emergency, when the mobile robot needs human assistance.

4. RESULTS AND DISCUSSION

The functionality of GUI3DXBot and its software modules was verified using the following tests: integration tests using the R.U.P. methodology in order to validate the functional requirements, and real world trials over a period of two weeks in the Electrical and Electronic Engineering School of the Universidad del Valle. After the guiding service was offered, the user asked to answer a survey about his experience with the GUI3DXBot. This section describes the experimentation setup conducted, and the results obtained with the real world trials where the robot system shows evidence of being well integrated. An example of the guidance service offered can be found in https://www.youtube.com/watch?v=bUg2RNfpRDQ

4.1 EXPERIMENTATION SETUP

At the very beginning, the mobile robot has no map of the environment. Then, in order to offer the guidance service this map must be computed in advance. This section describes the procedure to do that as follows:

1. The mobile robot platform depicted in Fig. 1 is powered on, and the position where it starts is assumed the home location.
2. Using a command terminal, the *RosAria* and *sicklms* nodes are launched in order to be able to obtain the laser data, and command the robot motion.
3. Using a command terminal, the *teleop_nav* node is launched in order to get the user keyboard inputs and move the robot over the environment, and save all the data gathered in a *bagfile*.
4. Once the data acquisition finishes, the software module shown in Fig. 2a is executed to compute the environment map. This map is an occupancy grid map with resolution of 0.08m.
5. Using an image processor, the PNG file that contains the environment map is opened and refined, it means: adding forbidden zones, clearing known zones, etc.
6. Finally, using *rviz* the home position is configured, as well as the goals coordinates.

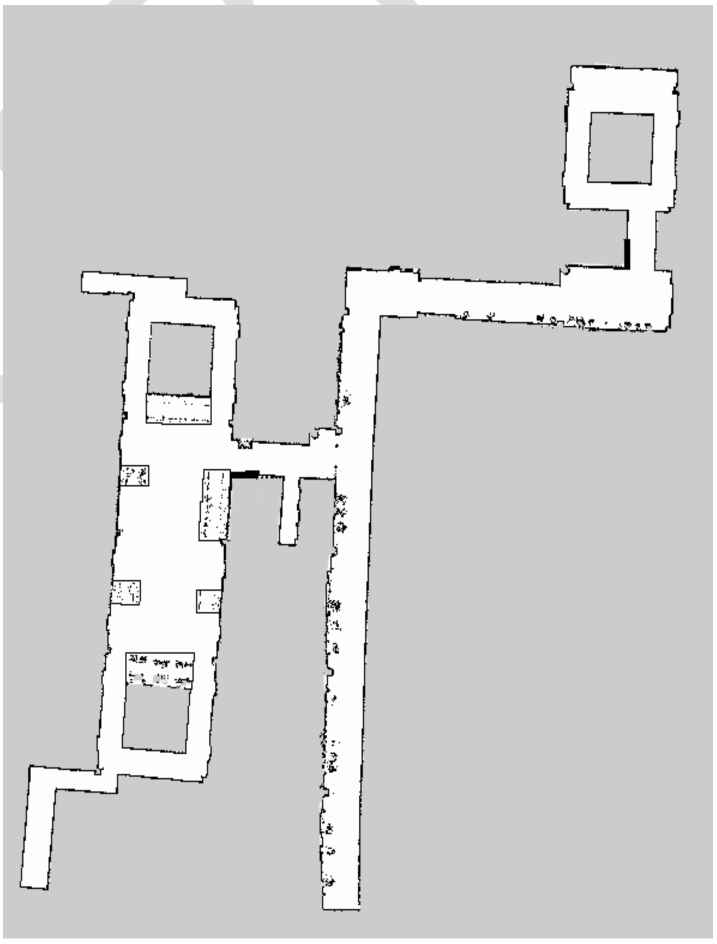


Fig. 6. Map of the second floor of buildings 353 and 354 of the Electrical and Electronic Engineering School of Universidad del Valle.

Using this procedure, the map corresponding to the second floor of buildings 353 and 354 of the Electrical and Electronic Engineering School of the Universidad del Valle was computed, and it is depicted in Fig. 6.

4.2 TESTS AND RESULTS

The procedure described in Section 4.1 is performed only once. But, if the environment changes, this procedure needs to be performed again. The real world trials reported in this section were carried out over two weeks. The environment, where the guiding service was offered, was not altered. It means people walked normally over these places and doing their normal activities. The trials were performed in business days in order to guarantee real conditions of experimentation, which in turn guarantee the robot system is well integrated and fully tested.

Over these three weeks 33 interactions with humans were registered. At the end of each human interaction, the user is asked to answer a survey. This survey was oriented to evaluate the experience with the guiding service. The questions asked are described as follows, as well as their answer and quantitative value:

1. In general, what do you think about the GUI3DXBot GUI? **Answers**: Very nice (5), Nice (4), Little nice (3), Unpleasant (2).
2. Do you think that the GUI3DXBot GUI interaction is intuitive? **Answers**: Very intuitive (5), Intuitive (4), Little intuitive (3), Nothing intuitive (2).
3. Do you think that the GUI3DXBot GUI provided enough information about your goal? **Answers**: Completely (5), Fair (4), Few information (3), Any (2).
4. How easy it was to choose a goal from the list in the GUI3DXBot GUI? **Answers**: Very easy (5), Easy (4), Difficult (3), Very difficult (2).
5. Do you think that the robot avatar provided enough information to use the GUI3DXBot application? **Answers**: Completely (5), Fair (4), Few information (3), Any (2).
6. How easy it was to understand the messages issued by the robot avatar? **Answers**: Very easy (5), Easy (4), Difficult (3), Very difficult (2).
7. Do you think that the robot icon on the map was useful? **Answers**: Very useful (5), Useful (4), Little useful (3), Nothing useful (2).
8. Do you think that the robot trajectory depicted on the map was useful? **Answers**: Very useful (5), Useful (4), Little useful (3), Nothing useful (2).
9. Do you think that the global and local view of the map were useful? **Answers**: Very useful (5), Useful (4), Little useful (3), Nothing useful (2).
10. How often the guiding service presented difficulties? **Answers**: Many times (2), Repeatedly (3), Rarely (4), Never (5).
11. In general, how do you rate the experience with the guiding service provided? **Answers**: Very satisfactory (5), Satisfactory (4), Unsatisfactory (3), Nothing satisfactory (2).
12. What would you change or add to the GUI3DXBot application?

Respondents were distributed as follows: 33% were Electronic engineering students, 12% were Master Science students, and 55% were non-engineering students (basic sciences, industrial designers and medicine students). It can be observed that most users were non-engineering students, which is important to consider different perceptions of GUI3DXBot.

Fig. 7 shows the results obtained from the users answers. These answers were analyzed considering seven aspects namely: GUI, goals, human-robot interaction using the robot avatar, computed trajectory, map visualization, problems over the guiding service, and user comments. Using questions 1 and 2 is possible to know how friendly is the GUI3DXBot GUI.

Results of question 1 show users like the appearance and structure of the GUI3DXBot GUI. But, to understand the results of question 2, it is worth saying that 52% of users who rated the GUI as *Intuitive* have some confusion with the button “Cancel”, however once they experienced with the GUI3DXbot, they resolved quickly their doubts.

Questions 3 and 4 provide evidence of how easy is to choose a goal in the GUI3DXBot software tool. The results shown are satisfactory, and this is due the goals list is compact, the goals are always present in the GUI then users can cancel a guiding service and selecting other goal, and goals description are meaningful. Human-robot interaction is important in a guiding service, then questions 5 and 6 can be used to know if users could understand the GUI3DXBot feedback messages. As can be observed in Fig. 7, users liked the robot avatar and they could understand its messages.

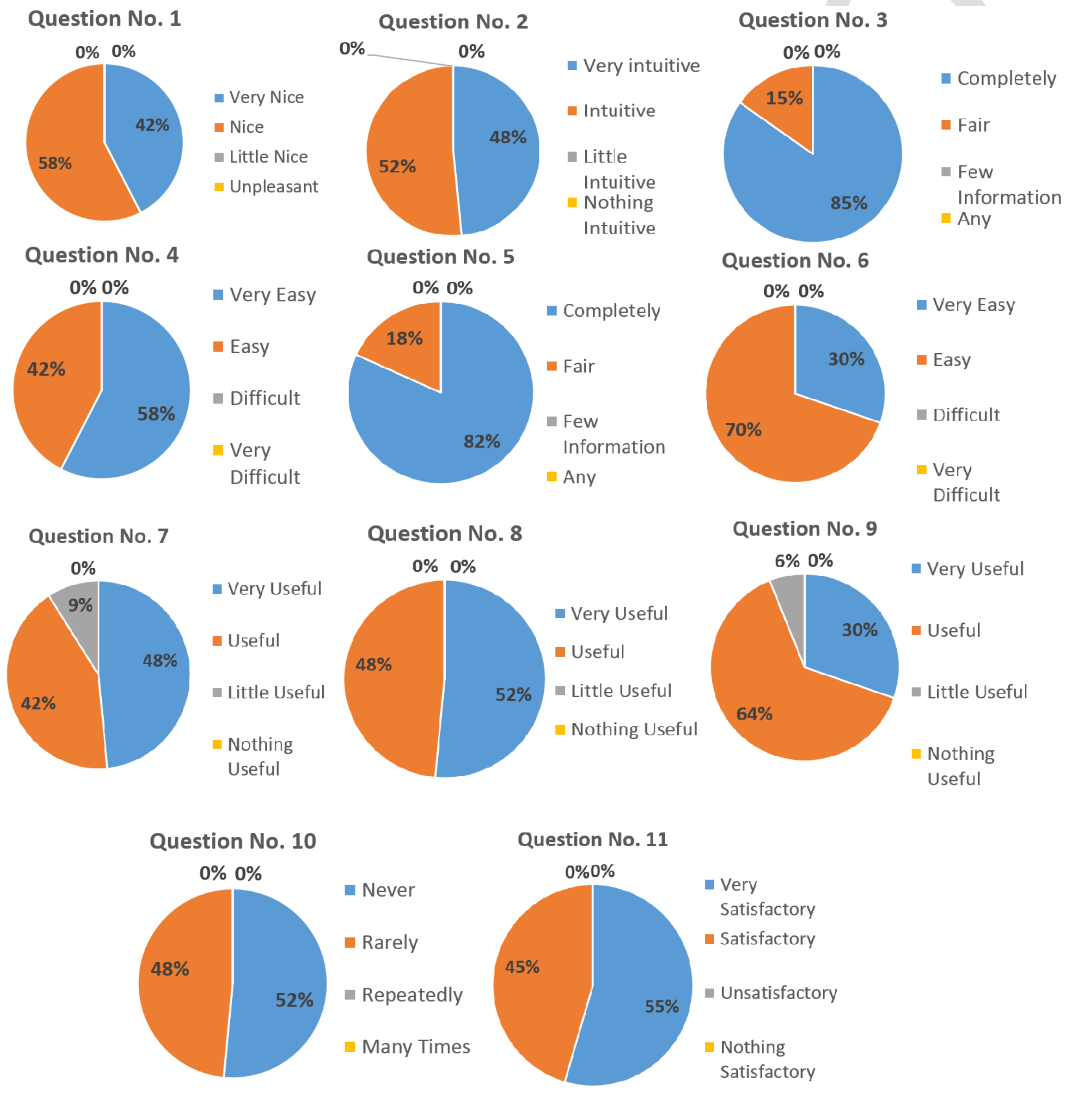


Fig. 7. Survey results.

Another type of human-robot interaction to test is proper status information. Here, question 7 helps to understand if the robot icon on the map was meaningful. Fig. 7 shows that 90%

of surveys found useful the robot icon, and it could be used to locate users in the environment. However, 10% found the robot icon little useful, these users argue this icon should be more attractive to people. Afterwards, question 8 shows evidence of how important is to show what actions are planned by the guidance robot, here Fig. 7 shows satisfactory results, and users said that plotting the robot trajectory helps them to memorize the path for further opportunities.

Guiding services not just allow users reach their goals, also encourage users to locate themselves. Question 9 shows the results to evaluate if users liked the local and global view of the environment map. Fig. 7 shows users used the functionality of changing the local and global view of the environment for the guiding service. However, 6% of them argue it is little useful, since they preferred the local view instead of the global view.

Evaluating the user experience to interact with service robots is important to observe the acceptance of this kind of applications. Questions 10 and 11 allow to perceive this aspect. The less failures the more acceptance in the guiding service. In this case, the survey reports GUI3DXBot 52% users had not failures in the service. Since the environment was not altered, and trials were performed in business days, corridors and halls were full of people crossing by. Then, failures perceived by users were robot motion stop, and start-stop motions due trajectory re-planning, but at the end the guiding robot reaches its goal. These are special cases, but in general users reported a very satisfactory (55%) and satisfactory (45%) experience of guidance as can be observed in the results of question 11.

Finally, additional comments made by users were: “Increasing the robot motion speed”, “Increasing the size of buttons for better access while the robot is moving”, “Adding voice commands”, “Improving the robot icon on the map, it is not friendly and it is not visible”, and “When the mobile robot is moving, it is difficult to access to the GUI buttons”. Reading these comments, most of them can be solved using a bigger touch-screen. The motion speed depends on the CPU power, since it has to compute the local cost maps for obstacle avoidance, then using a more powerful CPU could resolve this issue. And, changing the human-robot interaction using voice commands is an interesting improvement for further versions of GUI3DXBot software tool.

Table 2 shows the statistical results of survey about GUI3DXBot. To validate these results, it is expected that the maximum value deviation is less than 25% as depicted in (1).

$$Cv = \frac{|5-\bar{x}|}{5} * 100 < 25\% \qquad (1)$$

Where, $\bar{x}$ is the arithmetic mean, which in our case it is desirable that $\bar{x} > 4$. In addition, it is expected that the results mode corresponds to the maximum value (5). Then observing the results in Table 2, the deviation coefficient (Cv) is less than 25% to all results, the mean is bigger than 4 in all results, and the mode is 5 in the 63% of these results.

Table 2. Statistical results of survey about GUI3DXBot.

Question	***Mode***	***f_{MO}(%)***	$\bar{x}$	***Cv***
1	4	57.6%	4.4	13.0%
2	4	51.5%	4.5	11.5%
3	5	84.8%	4.8	3.1%
4	5	57.6%	4.6	9.3%
5	5	81.8%	4.8	3.8%
6	4	69.7%	4.3	16.2%
7	5	48.5%	4.4	13.8%
8	5	51.5%	4.5	10.7%
9	4	63.6%	4.2	17.9%
10	5	51.5%	4.5	10.7%

11	5	54.5%	4.5	10.0%

This work used an LRF as main sensor as the 81% of the related works shown in Table 1 providing good results for safe robot navigation and people guidance; also, this sensor allows creating an occupancy grid map as the 44% of works in Table 1, which in our knowledge is a better human-robot interface to represent the environment. In addition, this work used a Bayes based framework as the 75% of works in Table 1, which in this case is a proper choice given the dynamism environment facing the mobile robot. Finally, the touch-screen which is the human-robot interface selected to capture the human-robot interaction was a proper choice, since observing results of question 2, it obtained a mean of 4.5.

5. CONCLUSION

This paper presented the development and testing of different software modules of GUI3DXBot software tool, which offers an automated guiding service at the Electrical and Electronic Engineering School of the Universidad del Valle. GUI3DXBot is a client/server application, where the server side is in charge of: building a map of the environment, localizing the mobile robot, planning the route to the user goal, avoiding obstacles, always returning to the home location at the end of the tour-guide service, and issuing alerts in case of emergency. In turn, client side has the following functionalities: implementing the human-robot interaction, showing the goal list to users, showing the local and global map view, receiving the users selected goals, showing status information using a robot avatar, and showing emergency information.

The GUI3DXBot server side was developed using ROS, specifically the following packages were used: *RosAria* for the robot motion and sensing, *GMapping* to build the environment map, *AMCL* for robot localization, *Navfn* for local and global planning, and the web-sockets support to interact with the client side. The HRI was developed in Android using a tablet of 10in. Both, server and client side of GUI3DXBot were developed considering the RUP methodology. Fig. 2 shows the software module design which integrates all these packages in order to guide users in an office building.

To validate the functionality of GUI3DXBot a total of 20 human-robot interactions were documented using a survey. This survey was oriented to evaluate the experience with the guiding service. As a result, users qualify GUI3DXBot as *Very Nice* and *Intuitive GUI*, users found very easy to select goals, users could understand the interaction messages from GUI3DXBot, 90% of users found the robot icon on the map useful, users found useful drawing the planned trajectory on the map since it helps to localize themselves, 90% of users found useful the local and global view of the environment map, and in general the guiding service experience was very satisfactory (70%) and satisfactory (30%). This shows that the robot system was well integrated and fully tested in real world conditions.

Future works include increasing the human-robot interaction by adding functionalities such as voice recognition and visual people recognition. Through these functionalities the mobile robot can naturally interact with people, and can follow people in case they lose interest in the guidance service. Also, adding a redundant localization and mapping method based on features present at the ceiling, since LRF sensors cannot work properly in crowd environments.

REFERENCES

[1] K.-T. Song, Y.-H. Chiu, S.-H. Song, and K. Zinchenko, "Scheduling and control of a cloud robot for reception and guidance," in 2017 International Automatic Control Conference (CACS), 2017, pp. 1-6. https://doi.org/10.1109/CACS.2017.8284263

[2] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore, H. Hung, O. A. I. Ramírez, M. Joosse, H. Khambhaita, T. Kucner, B. Leibe, A. J. Lilienthal, T. Linder, M. Lohse, M. Magnusson, B. Okal, L. Palmieri, U. Rafi, M. van Rooij, and L. Zhang, "SPENCER: A Socially Aware Service Robot for Passenger Guidance and Help in Busy Airports," Springer, Cham, 2016, pp. 607-622. https://doi.org/10.1007/978-3-319-27702-8_40

[3] B. Bacca-Cortes, K. Muñoz Peña, and E. Caicedo Bravo, "GUI3DXBOT, Solución para la Asistencia a Peatones en Ambientes de Oficina Usando un Robot Móvil Guía (13-60-371)." Ministerio del Interior, Dirección Nacional de Derechos de Autor, Cali, p. 1, 2017.

[4] M. Quigley, B. Gerkey, K. Conely, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in ICRA, 2009, pp. 1-6.

[5] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva," Int. J. Rob. Res., vol. 19, no. 11, pp. 972-999, Nov. 2000. https://doi.org/10.1177/02783640022067922

[6] I. R. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer, and A. Soto, "An affective mobile robot educator with a full-time job," Artif. Intell., vol. 114, no. 1-2, pp. 95-124, Oct. 1999. https://doi.org/10.1016/S0004-3702(99)00027-2

[7] A. Bellarbi, S. Kahlouche, N. Achour, and N. Ouadah, "A social planning and navigation for tour-guide robot in human environment," in 2016 8th International Conference on Modelling, Identification and Control (ICMIC), 2016, pp. 622-627. https://doi.org/10.1109/ICMIC.2016.7804186

[8] S. J. Lee, J. Lim, G. Tewolde, and J. Kwon, "Autonomous tour guide robot by using ultrasonic range sensors and QR code recognition in indoor environment," in IEEE International Conference on Electro/Information Technology, 2014, pp. 410-415. https://doi.org/10.1109/EIT.2014.6871799

[9] S. Wang and H. I. Christensen, "TritonBot: First Lessons Learned from Deployment of a Long-Term Autonomy Tour Guide Robot," in 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2018, pp. 158-165. https://doi.org/10.1109/ROMAN.2018.8525845

[10] D. Rodriguez-Losada, F. Matia, R. Galan, M. Hernando, J. Manuel, and J. Manuel, "Urbano, an Interactive Mobile Tour-Guide Robot," in Advances in Service Robotics, InTech, 2008. https://doi.org/10.5772/5950

[11] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard, "Autonomous Robot Navigation in Highly Populated Pedestrian Zones," J. F. Robot., vol. 32, no. 4, pp. 565-589, Jun. 2015. https://doi.org/10.1002/rob.21534

[12] K. Yelamarthi, S. Sherbrook, J. Beckwith, M. Williams, and R. Lefief, "An RFID based autonomous indoor tour guide robot," in 2012 IEEE 55th International Midwest Symposium on Circuits and Systems (MWSCAS), 2012, pp. 562-565. https://doi.org/10.1109/MWSCAS.2012.6292082

[13] Ching-Chih Tsai, Shu-Min Shish, Hsu-Chih Huang, Min-Yu Wang, and Chih Chung Lee, "Autonomous navigation of an indoor tour guide robot," in 2008 IEEE Workshop on

Advanced robotics and Its Social Impacts, 2008, pp. 1-6. https://doi.org/10.1109/ARSO.2008.4653591

[14] F. Faber, M. Bennewitz, C. Eppner, A. Gorog, C. Gonsior, D. Joho, M. Schreiber, and S. Behnke, "The humanoid museum tour guide Robotinho," in RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication, 2009, pp. 891-896. https://doi.org/10.1109/ROMAN.2009.5326326

[15] H.-M. Gross, H. Boehme, C. Schroeter, S. Mueller, A. Koenig, E. Einhorn, C. Martin, M. Merten, and A. Bley, "TOOMAS: Interactive Shopping Guide robots in everyday use - final implementation and experiences from long-term field trials," in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 2005-2012. https://doi.org/10.1109/IROS.2009.5354497

[16] Y. Chen, F. Wu, W. Shuai, and X. Chen, "Robots serve humans in public places-KeJia robot as a shopping assistant," Int. J. Adv. Robot. Syst., vol. 14, no. 3, pp. 1-20, 2017. https://doi.org/10.1177/1729881417703569

[17] R. Stricker, S. Müller, E. Einhorn, C. Schröter, M. Volkhardt, K. Debes, and H. M. Gross, "Konrad and Suse, two robots guiding visitors in a university building," Inform. aktuell, pp. 49-58, 2012. https://doi.org/10.1007/978-3-642-32217-4_6

[18] H. Lin and W. Tsao, "Automatic mapping and localization of a tour guide robot by fusing active RFID and ranging laser scanner," 2011 Int. Conf. Adv. Mechatron. Syst., pp. 429-434, 2011.

[19] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," IEEE Trans. Robot., vol. 23, no. 1, pp. 34-46, Feb. 2007. https://doi.org/10.1109/TRO.2006.889486

[20] P. Kruchten, The Rational Unified Process: An Introduction, 3rd ed. Addison-Wesley Professional, 2003.

[21] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, pp. 709-715.

[22] J. Mace, J. Lee, R. Toris, B. Alexander, D. Bertram, and M. Gruhler, "RosBridge Protocol Specification," GitHub, 2018. [Online]. Available: https://github.com/RobotWebTools/rosbridge_suite/blob/groovy-devel/ROSBRIDGE_PROTOCOL.md. [Accessed: 01-Sep-2018].