Intelligent Service Robotics (2019) 12:319–332
https://doi.org/10.1007/s11370-019-00285-8

ORIGINAL RESEARCH PAPER

GidaBot: a system of heterogeneous robots collaborating as guides in multi-floor environments

Oihane Parra[1] · Igor Rodriguez[1] · Ekaitz Jauregi[2] · Elena Lazkano[1] · Txelo Ruiz[3]

Received: 14 February 2019 / Accepted: 13 July 2019 / Published online: 12 August 2019

Abstract
GidaBot is an application designed to set up and run a heterogeneous team of robots to act as tour guides in multi-floor buildings. Although the tours can go through several floors, robots are not allowed to use the lift, and thus, each guiding task requires collaboration among several robots, one per floor. The designed system makes use of a robust inter-robot communication strategy to share goals and paths during the guiding tasks. A user-friendly GUI helps untrained users or new visitors to easily choose target locations or define a list of locations to be visited sequentially. A prototype has been implemented using ROS, and the system robustness has been tested in a Gazebo-based simulated robot/environment and using real robots at the Faculty of Informatics in San Sebastian. The whole application is available together with a simulated world so that the system functioning can be checked further.

Keywords Service robots · Tour-guide robots · Distributed robotic system

1 Introduction

Robots are moving out of industries, and they are ready to take part in our daily lives. The International Federation of Robotics (IFR) defines service robots as robots that aim to assist and/or perform useful tasks for humans and categorizes them according to the type of interaction they are able to have [6]. Instances of service robots have been set up for different purposes, for example, in restaurants, offering coffee to the clients [14]; in hospitals, transporting critical patients to the surgery room [25]; in warehouses, moving material from one location to the other within distribution centres [26]; in retail stores, guiding the customers to the products of their choice [4]; robots acting as shopping assistants [7]; or even working in nursing-care situations [11]; and assisting the elderly at home [2].

Despite the variety of already commercially available service robots, most of them have little presence in our lives yet. The most well-known exception is the iRobot's Roomba vacuum cleaner [3], which was designed to carry out the domestic task of cleaning the floor of our houses.

A different application scope of service robots is that of tour-guide robots, application that heavily relies on autonomous navigation capabilities. The research here presented focuses on a distributed heterogeneous robot navigation system, which we have called GidaBot, that enables robot communication for cooperative guiding tasks in different floors. The peculiarity of the system is that in the guiding task proposed individual robots are restricted to function as guides in a unique floor, and multirobot communication and cooperation are used to fulfil tours where locations in different floors need to be visited. Nevertheless, multiple robots could coexist in each floor and behave cooperatively by extending the information broadcasting system employed for the current robot-environment configuration. The design of the system's general features is detailed, together with the description of the GUI and the information required to be exchanged among the involved robots. A ROS-based prototype of the designed system has been implemented and tested for a particular configuration in both simulated and real robot/environment system. Albeit the system has been set up to solve the 4 floor navigation problem of the Faculty

✉ Igor Rodriguez
igor.rodriguez@ehu.eus

✉ Elena Lazkano

[1] Computer Science and Artificial Intelligence, Faculty of Informatics, UPV/EHU, Donostia-San Sebastian, Spain

[2] Computer Languages and Systems, Faculty of Informatics, UPV/EHU, Donostia-San Sebastian, Spain

[3] Computer Architecture and Technology, Faculty of Informatics, UPV/EHU, Donostia-San Sebastian, Spain

of Informatics in San Sebastian, the system can be adapted to a different building and robot configuration.

2 Tour-guide robots

The literature review reveals several instances of robots acting as tour guides. Minerva [19] is very likely the first robot that acted as such in the Smithsonian's National Museum of American History in Washington, and by far the most cited one. In [16], the navigation capabilities of CoBot are evaluated acting as a guide by cooperating with the visitor, helping each other to fulfil the task. More recently, a robot that performs guided tours was designed, built and set up at the Eureka Science Museum of San Sebastian [18]. Some authors emphasize the need for social interaction in such platforms. Robovie assisted visitors at the Osaka Science Museum Exhibit [17], and the humanoid robot Robotinho [1], mounted on a wheeled platform to reduce mobility constraints, showed such capabilities guiding visitors in the Deutsches Museum of Bonn.

Trahanias et al. [21] present a different approach in which robots are teleoperated over the internet and act as interactive agents in populated environments as museums and exhibitions. In addition, Hristoskova et al. [5] propose a distributed system in which two robots cooperate in guiding tasks. Robots share profiles and tour information with the aim of automatically exchanging the group members in order to optimize the amount of interesting content each time robots are in the neighbourhood.

Looking forward in a near future, one crucial challenge to be considered is the usefulness of those robots—initially intended to be used in single floor buildings—in public places with multiple floors. A possible solution to the multi-floor navigation problem is the use of a single robot that can navigate through different floors using lifts, as the robot Charlie [22] is able to do. A similar work extended to multiple robots, also using elevators, is proposed in the GuideBot tour guide [8] and BellBot hotel assistant [9] systems. Those systems need a central server and behave as a master–slave computer, and do not have knowledge about other agents. But robots that get into lifts are supposed to have the necessary abilities to interact with the lift interface, inside and outside, to execute precise actions. That kind of technology is not commonly available in buildings. The lack of proper actuators can be overcome by interacting with humans as CoBot does [16,24]. This symbiotic collaboration approach has been further expanded to a homogeneous team of up to 4 robots that are also able to perform delivery tasks [23]. Instead of robots that communicate with humans, in the proposed system robots communicate with robots. A fully distributed and heterogeneous robot team cooperates sharing services, without a central computer. In the specific experimental set-up, we limited the work scope of each robot to a single floor and do not allow the robots to enter the elevators. Although several platforms are needed, robot navigation is more secure (robot paths don't collide) and several tours can be run concurrently. In this way, the lift remains available for people involved or not in the guided tour and for people with reduced mobility.

The design of the multirobot guide system described in this research paper relies on the assumption that basic robot navigation is available, i.e. robots can maintain localization, plan paths and execute them. This navigational system will run underneath GidaBot.

Taking that in mind, GidaBot's design aspects can be categorized as follows:

1. The operation modes: the choices the system will offer to the visitor.
2. The Graphical user interface (GUI) that will be the basic interaction tool.
3. The information exchanged among robots, i.e. the definition of the different types and formats of the messages required for the global system.

Those three elements are further explained hereon in separated sections.

3 GidaBot's operation modes

Buildings open to the general public offer several alternatives for visiting them, and thus, the system has to provide different operation modes to satisfy those alternatives. A visitor may require to be guided to a particular location/office, whereas groups of visitors may prefer to be driven to a sequence of interesting locations to get an overview of the place. The multirobot guide system is designed to allow for two operation modes: single target mode and tour mode.

3.1 Single target mode

In single target mode, the user can only select a single target from all available locations. The robots must cope with different situations:

1. The user and the desired goal are on the same floor. In this situation, only one robot will guide the user from the beginning to the end of the navigation. Therefore, the only action the robot must perform is to reach the goal.
2. The user and the desired goal are on different floors. In this case, two robots are involved in guiding tasks; one is located in the floor where the trip starts and the other one in the floor where the navigation ends.

(a) In the floor where the navigation starts, the robot will guide the user from the starting location to the lift or the staircase (the user is free to choose) and it will indicate the visitor what floor to go. The next robot will be waiting for the user in the goal floor's meeting point to solve the remaining path to the goal location.
(b) In the floor where the navigation ends, the robot needs to meet the user before guiding him/her, so it has to move to the previously chosen lift or staircase (the meeting point). Afterwards, when the user arrives, he/she will notify the robot to go on. Then, the robot will guide him/her to the goal and the navigation will finish.

3.2 Tour mode

Often, it can be interesting to follow a predefined goal sequence, for instance, to get an overview of the inside of the building. This means that robots must conduct guided tours. For this purpose, the system must allow to create tours as a collection of location goals in the desired sequence. Tours are saved in a local directory and can be edited. New and edited tours are automatically shared among all the available robots involved in the system.

Of course, the *Tour mode* relies on the *Single target mode* and can face the same situations at each target of the sequence.

4 The graphical user interface

Tour-guiding robots need to interact with humans. The simplest way of interaction requires a graphical user interface so that tasks and goals can be set by the user and information can be provided to him/her in a practical manner.

Figure 1 shows a general overview of the interface developed.

The main window comprises a tab per floor, showing in each tab buttons of the most interesting locations and the current position of the robot on the floor's blueprint. That position is obtained after translating the robot's localization in the navigational map (provided by the navigation system) into the interface's floor blueprint (the scale of the map and its relation with the displayed blueprint must be known). Three more tabs (*Offices*, *Guided Tour* and *Information*) together with the current navigation information (text information about the current location and the starting and goal points) complete the interface. The *Offices* tab shows information about the available destination points per floor, in order to help the user find the desired target locations. The *Guided Tour* tab allows the user to select and follow a predefined goal sequence. And the *Information* tab offers information about the operative robots and details like velocities and battery level of the current robot. Moreover, the *Information* tab also shows the number of pending requests of each operative robot (see Fig. 2).

GidaBot's graphical interface has been designed to offer the user:

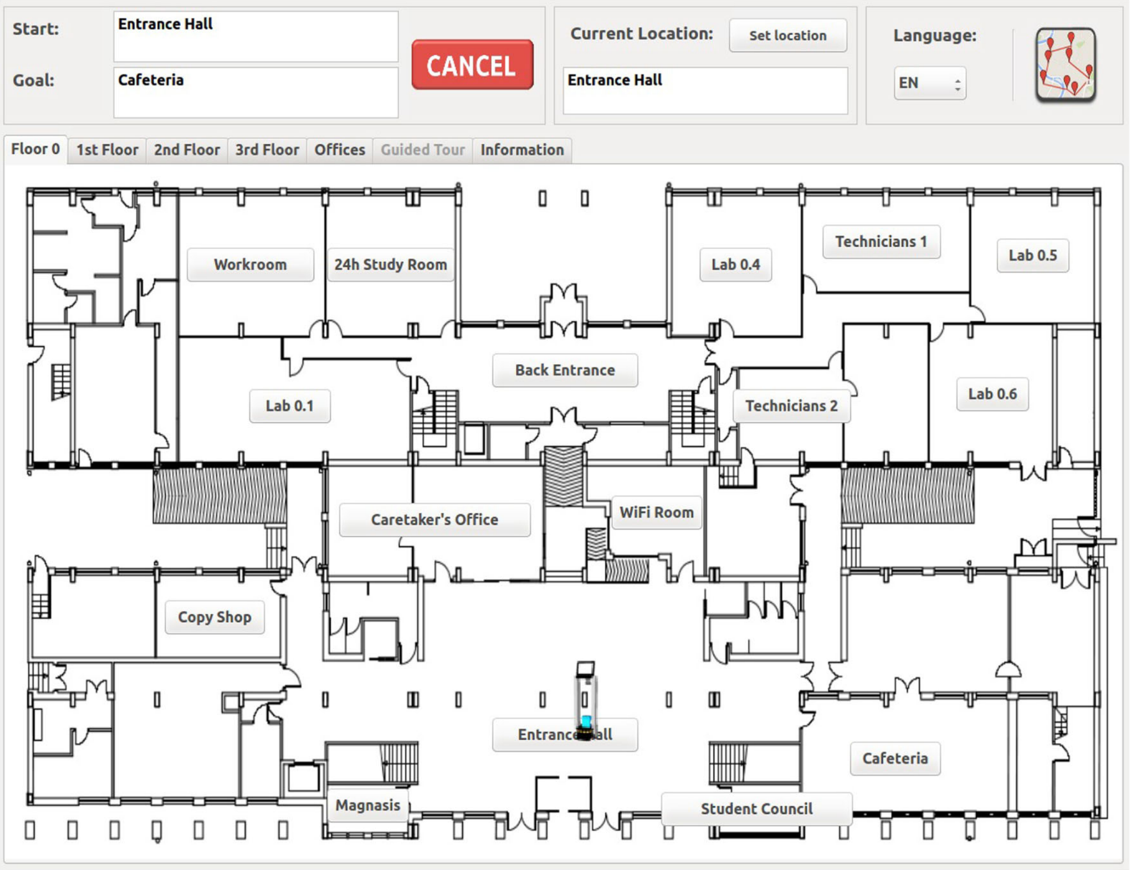
The option to choose among the different operation modes available The upper right button offers the user the option to change the operation mode. The icon of the button changes according to the selected mode.

The option to select the language of the system A pop-up menu allows to select the graphical interface's language. Options are Basque, English and Spanish.

Visual detailed information about the different locations and easy definition and management of tours On the one hand, in single target operation mode, after selecting the tab corresponding to the floor of interest, the user has to click on the destination button, accept the confirmation message and choose the way to move between floors (if there is a floor change). On the other hand, the *Guided Tour* tab (Fig. 3) offers the user the option to select and follow an already predefined tour. In tour mode, the user can start a tour by clicking *Start again* button. The visitor must confirm that he/she is ready to go on by clicking on the *Continue* button. Alternatively, he/she can opt to skip any goal by selecting the next desired one from the list.

Information about the current state of the robot during any guiding task When the robot moves, the GUI shows its location on the floor's map during the whole trip. But, at the same time, on the appropriate floor tabs, the location of the rest of the robot troop is continuously updated. Moreover, the system always keeps the user informed, via text pop-ups and verbal messages, about the trip, the robots' navigation state and the actions needed to perform. For instance, the robot notifies the user when the target point is reached, or in case of a floor change, it informs about the floor he/she has to move to in order to continue with the trip. Besides, if a goal is unreachable, for instance, when every possible path is closed, the robots involved in the navigation will be informed and the user will be told that the robot cannot help him/her.

Option to cancel the task at any time Regardless of the mode, if the user wants to finish the navigation on the way to the goal, the "Cancel" button must be clicked. The robot will stop, and it will immediately become ready to process a new goal. In case the navigation comprises several floors, the rest of the affected robots' navigation will also finish. Besides, each time the user wants to send a goal, the GUI informs him/her about the number of pending requests of the goal robot. Therefore, if he/she thinks it will take long to wait, the visitor can always choose to cancel the task.



Start: Entrance Hall

Goal: Cafeteria

Current Location: Entrance Hall

Language: EN

Fig. 1 Main view of the GUI

Those options are available for any visitor. However, it is possible to launch the GUI as administrator, and as such, it provides facilities to control the following aspects:

The possibility to set up the robot's location The graphical interface provides the option to indicate the robot's position and orientation in a simple manner, using the current floor's map in the GUI (see the "Set location" button in Fig. 1)

The option to easily manage guided tours The superuser can create, edit and delete existing tours when needed using the buttons on the bottom part of *Guided Tour* tab (see Fig. 3).

5 Robot communication: information exchange

The multirobot guide system proposed here is mainly designed to work with several robots interconnected carrying out collaborative navigation tasks. Of course, this implies that single navigation is also supported. When multiple robots are involved, information exchange among the robots is required.

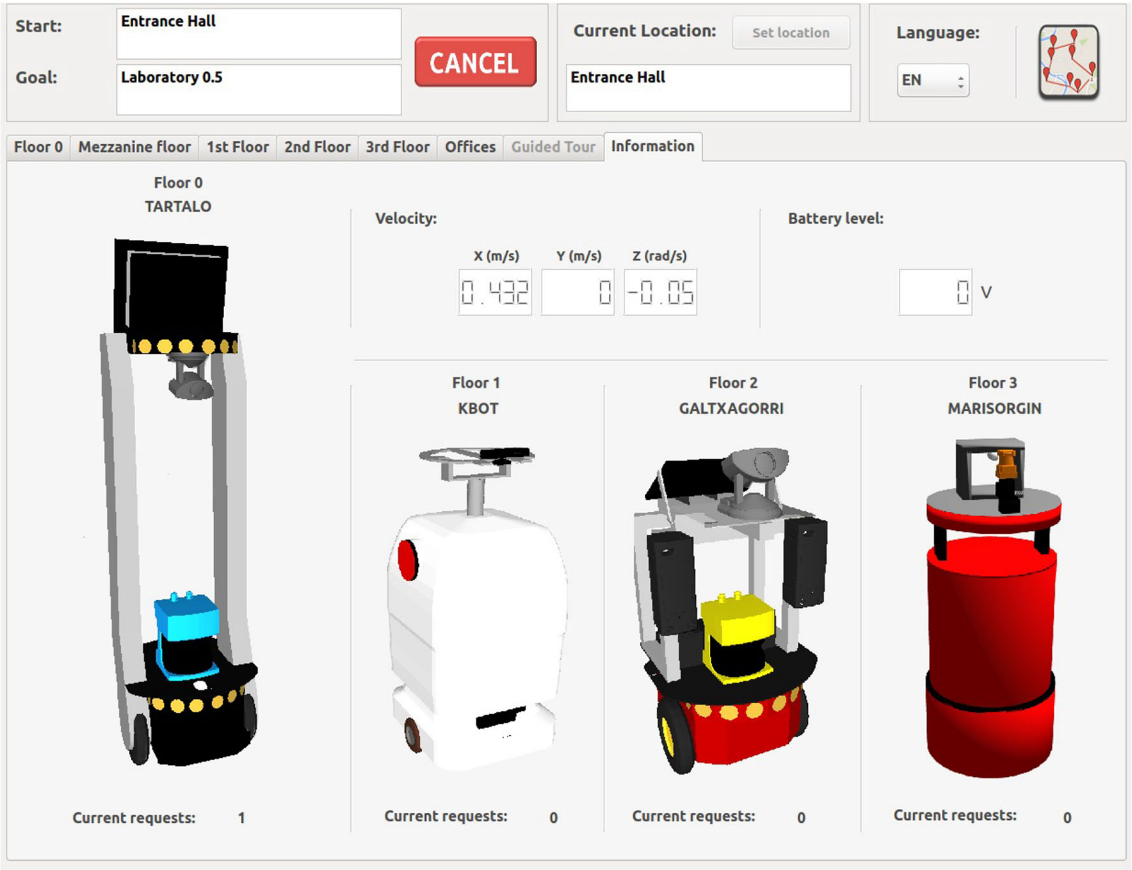
No matter how many choices the system offers, if the system is to be robust and efficient, then it is mandatory to ensure an adequate communication among robots. In order to make the system work as desired, each robot has to inform the others, on the one hand, about user's requests and, on the other hand, about its state—current location and navigation state. This communication must be fluent over time.

Context exchange among robots relies on different type of messages explained below.

5.1 Goal descriptions

A goal comprises the information related to the start and end points of the navigation. Note that one or several robots can be involved in guiding tasks:

- Number of the initial floor, where the visit starts.
- Coordinates of the initial robot location.
- Number of the goal floor, where the visit ends.
- Coordinates of the goal location.
- The way to be taken to move between floors (the lift or the staircases).
- Start point identifier.
- Goal point identifier.
- Language to be used for (verbal and text) communication.



Start: Entrance Hall

Goal: Laboratory 0.5

Current Location: Entrance Hall

Language: EN

Floor 0 TARTALO

Velocity: X (m/s) 0.432 Y (m/s) 0 Z (rad/s) -0.05

Battery level: 0 V

Current requests: 1

Floor 1 KBOT Current requests: 0

Floor 2 GALTXAGORRI Current requests: 0

Floor 3 MARISORGIN Current requests: 0

Fig. 2 Information tab showing the operative robots and the available information

5.2 Tour description

Tours comprise a predefined goal sequence and are defined as local text files in a three column format with the following content: *floor, room number, room description*. Here an example of a tour definition text file:

```
Tour
0, 014, Computer Technicians 1
0, 008, Caretaker's Office
1, 101, Secretary's Office
1, 104, Auditorium
2, 223, Mikel LARREA ALAVA
2, 274, Kepa SARASOLA GABIOLA
3, 325, Itziar IRIGOIEN GARBIZU
3, 305, RSAIT
```

When several robots are involved in a new tour, the tour is shared among the platforms involved, which means that each robot has to update the list of available tours and show it in the corresponding tab of its GUI. The tour information is shared via a message that contains the following fields:

- Robot id: robot where the tour was created.
- Tour name: the name given to the tour.
- Tour file name: the name of the file where the tour is saved.
- Tour information: contains the goal sequence information, in string format.

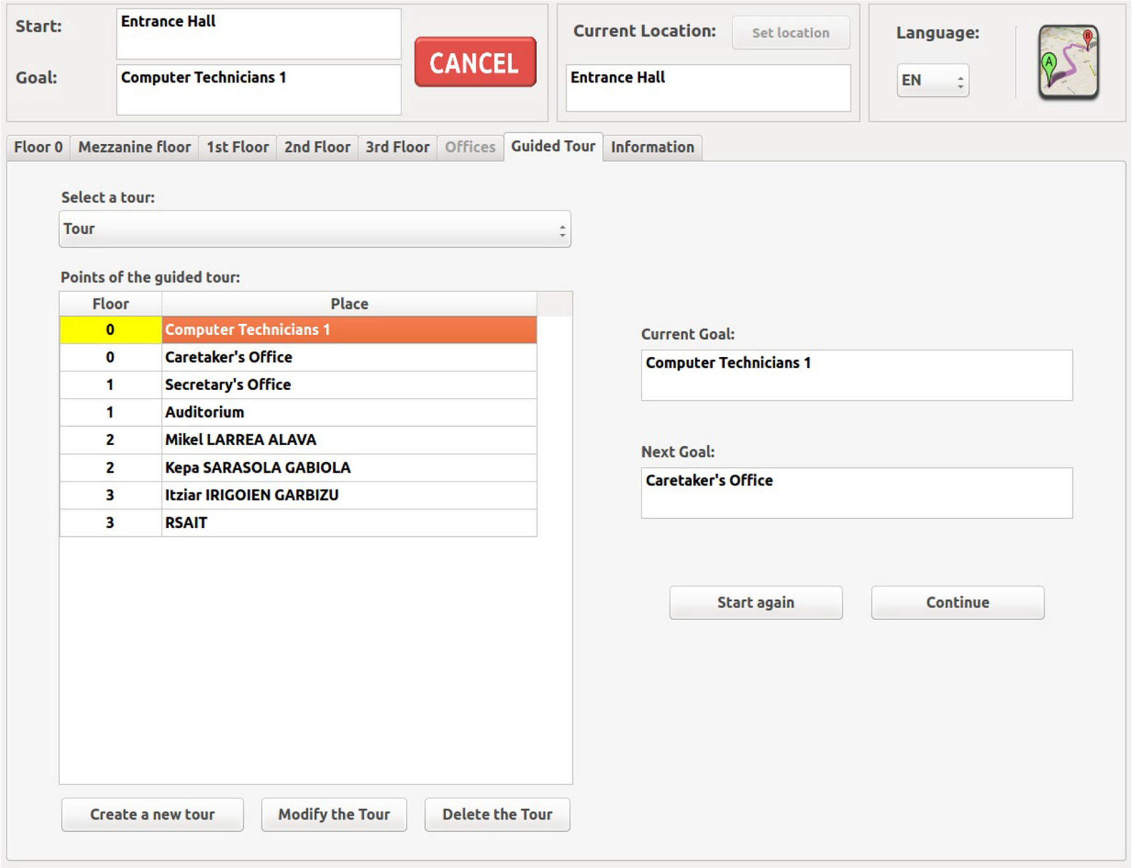
5.3 Robots' pose

As the GUI shows every available robot's localization in the map, it needs to keep updated each robot's position in the corresponding tab. Thus, robots continuously interchange their pose:

- X: the X coordinate of the robot.
- Y: the Y coordinate of the robot.
- Orientation: the orientation of the robot.

5.4 Pending requests

As mentioned before, when a user needs to be guided to a destination, she/she is informed about the pending requests of the robots involved in the task, so that he/she can decide



Start: Entrance Hall

Goal: Computer Technicians 1

Current Location: Entrance Hall

Language: EN

Select a tour: Tour

Points of the guided tour:

Floor	Place
0	Computer Technicians 1
0	Caretaker's Office
1	Secretary's Office
1	Auditorium
2	Mikel LARREA ALAVA
2	Kepa SARASOLA GABIOLA
3	Itziar IRIGOIEN GARBIZU
3	RSAIT

Current Goal: Computer Technicians 1

Next Goal: Caretaker's Office

Fig. 3 GUI tab for guided tours

to abort the task or to go on. This type of messages is defined as:

- The number of pending requests.
- The list of pending goals.

6 GidaBot's prototype

The preceding sections described the design aspect of GidaBot. We have implemented a ROS-based prototype to show the viability of the approach. The developed ROS nodes, their communication and execution are described hereon, as a showcase.

The multirobot guide system has four components:

1. The graphical user interface.
2. The ROS navigation stack.
3. The multirobot subsystem, composed by two nodes: *multirobot_navigation* node and the *goal_manager* node.
4. The communication subsystem.

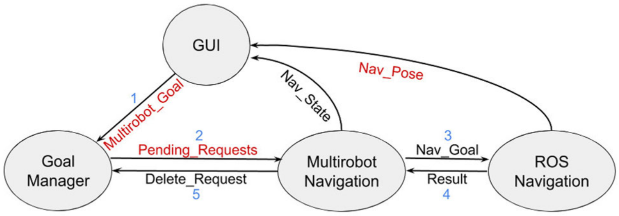


Fig. 4 Communication among system nodes

All those elements are executed in every robot. Figure 4 describes the connection among these elements. The numbers on the links denote the sequence to process a goal, i.e. the communication flow among the nodes during the navigation task. Note that the text highlighted in red color refers to those messages shared among the robots. When a user confirms that a goal must be reached, a *multirobot_goal* message is created and broadcast to every robot. The *goal_managers* then will add the goal to their *pending_requests* only when the corresponding robot is involved. As the *multirobot_navigation* node is permanently checking the pending requests list, when the list changes it will activate the navigation process (see Algorithm 1 for a more detailed description).

We will focus on the description of the last three elements.

6.1 Robot navigation in ROS

The navigation subsystem constitutes the pillar of any guide system. Robot navigation implies that the robot is able to determine and maintain its own position and plan a path towards some goal location, avoiding dangerous situations, such as collisions with objects. Probabilistic robot localization methods make use of probability distributions to represent and maintain uncertainty in robot localization and feature identification over time in order to determine the path the robot must follow to fulfil a task [20]. These probabilistic approaches have shown to perform well in semi-static environments and thus are being widely used in multiple robotic systems. ROS[1] provides a navigation stack initially developed for the PR2 robot by Willow Garage [10] that has been adapted for many robots.[2] This navigation stack offers tools for constructing a global map of the robot's environment by means of SLAM (self-localization and mapping) techniques. Besides, robot localization during navigation is maintained using particle filter-based AMCL (augmented Monte Carlo localization) algorithm. Together with the map and a robot localization mechanism, the navigation stack needs a planner to find and select the path to be followed by the robot. ROS allows the user to configure the stack by choosing among several planners the one that better fits to the robot/environment system. The default navigation function makes use of Dijkstra's algorithm for planning purposes.

The interfloor multirobot guide system developed in this work makes use of the ROS navigation stack, adapted to each of the platforms involved in the system, but here, we will not get into details on the navigation parameters set-up because it is out of the scope of the work. Note that due to the morphology of our robots crossing doors is insecure and thus, for the time being the map of each floor does not include the internal description of the rooms. Therefore, the guide system limits the robot navigation to the front of the doors that give access to the target locations.

6.2 Multirobot navigation

Two ROS nodes have been implemented that make possible the multirobot navigation:

The goal_manager node This node manages the list of pending requests. If a robot receives more than one request, these are queued in order of arrival and managed using a first-in-first-out (FIFO) queue. Thus, pending goals (with initial or final point in the robot's floor) are processed in the same order they are requested.

The multirobot_navigation node This main node receives the list of the pending requests (which includes the navigation goals), processes this information and then sends the robot to the pertinent place. Once a robot has finished processing a request, if its queue is not empty, it will start navigating to the next goal (the first in the queue). It also provides the GUI with all the information required to keep the user informed; the trip, the robot's navigation state and the actions needed to be performed, as well as the failures detected by the navigation subsystem are exchanged. Algorithm 1 describes more in detail the *multirobot_navigation* node.

6.3 Messages and broadcasting

The GidaBot system is designed to work with several robots interconnected carrying out collaborative navigation tasks although single navigation is also supported. The approach employed for robots' communication relies on a specific information exchange. Table 1 shows the definition of the information to be interchanged in terms of ROS messages.

Those messages need to be available to all robots, and afterwards, each one will decide whether the received information is relevant or not. However, ROS is not designed for distributed systems in which information must be shared among all the entities. Hopefully, there are two packages that facilitate a solution. Our first attempt was to use the *multimaster_fkie* package, which allows to establish and manage a multimaster network using multicast protocol, but our faculty's network firewall does not allow many-to-many distribution. For that reason, our system was developed using the *multimaster* package, which—though deprecated—enables communication between two ROS masters. What it does, exactly, is to register topics or offer services at a different ROS master, and/or subscribe to topics or call to services of the same master. In this manner, topics/services managed by different masters are selected to be shared. Thus, the use of the *multimaster* package is a one-time necessity, a tool that makes possible to distribute topics among machines in ROS.

In the developed system, each robot has its own ROS master and all robots are interconnected on a complete graph network, resulting in a low latency messaging platform. In order to share information using the *multimaster* node, it is enough to execute this node in just one of the two masters we want to connect. This means we need $\frac{n(n-1)}{2}$ multimasters, where n is the number of robots. Before running the node, the foreign master must be specified, together with the local publications we wish to share and the foreign publications to be received.

Figure 5 shows the system architecture for 4 robots. Notice the multimasters on the bottom row. M_{ij} denotes that the mul-

[1] http://www.ros.org.

[2] http://wiki.ros.org/navigation/RobotsUsingNavStack.

```
Function NavigResultActions(nav_resp):
if success then
    inform GUI(success);
    delete request from pending_list;
end
else
    inform GUI(nav_failure);
    delete request from pending_list;
end
EndFunction

Function MainFunction:
get pending request list
while pending_list not empty do
    /* Get current. Message type: Goal                */
    goal = read current;
    /* Navigation starts with the user and
       the robot in the same floor                    */
    if goal.init_floor == current_floor then
        /* The goal is on initial floor               */
        if goal.goal_floor == current_floor then
            send robot to the goal;
            wait until robot's navigation ends;
            NavigResultActions(nav_result);
        end
        /* The goal is on another floor               */
        else
            choose lift or stairs;
            send the robot to chosen point;
            wait until robot's navigation ends;
            NavigResultActions(nav_result);
        end
    end
    /* Navigation starts without the user.
       The robot has to meet the user that
       comes from a different floor                   */
    else if goal.init_floor != current_floor and goal.goal_floor
    == current_floor then
        /* STEP 1: Send the robot to the front
           of lift/stairs                             */
        send the robot to the front of chosen lift/stairs;
        wait until robot's navigation ends;
        NavigResultActions(nav_result);
        /* STEP 2: Wait to the user                   */
        wait until the user presses "Continue" button;
        if button is not pressed after a lapsed time then
            delete request from pending_list;
            return;
        end
        /* STEP 3: Send the robot to the goal
           */
        send robot to the goal;
        wait until robot's navigation ends;
        NavigResultActions(nav_result);
    end
end
```

Algorithm 1: *multirobot_navigation* node

Table 1 Messages description

Message type	Data type	Name
Goal	float32	initial_floor
	geometry_msgs/Point	initial_pose (x, y)
	int	goal_floor
	geometry_msgs/Pose	goal_pose (x, y, θ)
	string	way
	string	start_id
	string	goal_id
	string	language
Tour	float32	floor
	string	tour_name
	string	file_name
	string[]	tour_goals_info
Pose	geometry_msgs/Pose	nav_pose (x, y, θ)
Pending requests	uint8	num
	Goal[]	goals

timaster communicates masters i and j, i.e. robots i and j. For the experimental set-up used in this work, 6 multimasters are needed in order to have full intercommunication among the 4 robots.

7 Experimental set-up

The developed guide system has been tested with four different robots navigating on the four floors of the Faculty of Informatics. The experimental set-up is described below.

7.1 Robotic platforms and environment description

Kbot is a differential drive robot supplied with a Sick S3000 laser scanner built by Neobotix [12] in 2004. It was recently renewed, with a new PC and a Kinect sensor. This robot is now ROS operative, since the necessary ROS drivers have been developed for it [15].

MariSorgin is our heirloom robot, a synchro-drive robot that dates from 1996. It is a B21 model from real-world interface provided with a ring of ultrasound, infrared and tactile sensors for obstacle avoidance. In addition, a Hokuyo URG-30 laser, a Kinect sensor and a Heimann thermopile have been placed on top of the enclosure.

Last but not least, *Galtxagorri* and *Tartalo* are two differential drive robots, a Pioneer-3DX and PeopleBot models from Omron Adept MobileRobots [13], respectively. Both are provided with ultrasounds and a Cannon VCC5 camera; the former has a Leuze RS4 mounted on its top and the later a Sick LMS200 laser sensor on its base.

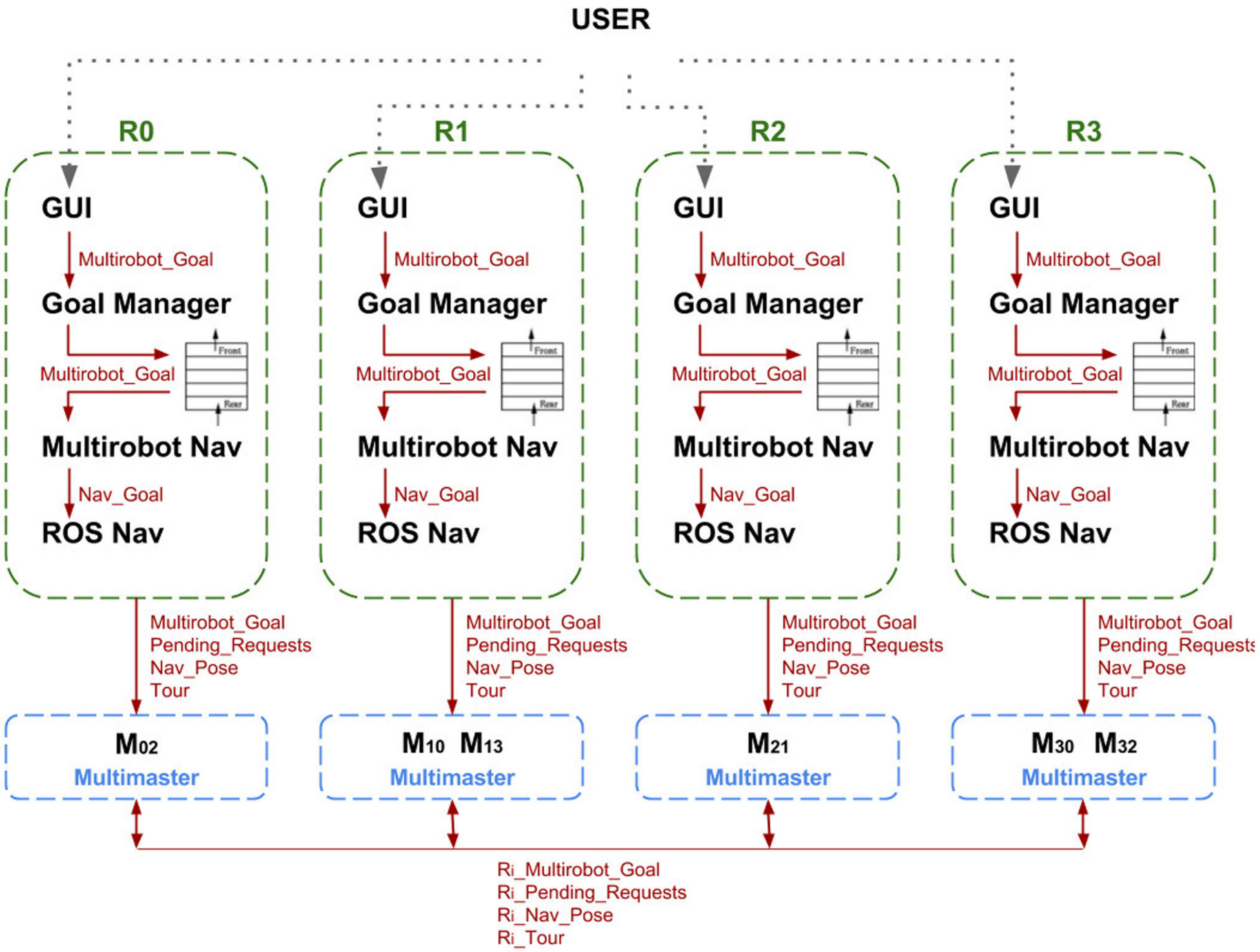


Fig. 5 Multirobot system architecture

All the robots have thus a laser sensor for safe navigation and localization, a touch screen for interaction purposes and speakers to be able to reproduce audio. This brief description gives a hint of the diversity of sensors being used and the dissimilar morphology and sizes, in summary the heterogeneity of the robot team.

Regarding the environment, the Faculty of Informatics of the University of the Basque Country (UPV/EHU) is located in San Sebastian. It is a four-floor building equipped with two side staircases and a single lift that enables people to move between these floors.

The main entrance is in the zeroth and lowest floor, where a few lecture rooms and laboratories are placed . The Dean's Office, Secretary's Office and Auditorium can be found in the first floor, together with more lecture rooms and laboratories, whereas most professors' offices are located in the second and third floors. Research laboratories are also in this upper floor.

7.2 Wireless communication

In order to be able to communicate over time, either robots are in the same LAN or they have a known public IP address externally accessible. The Faculty of Informatics is a public building of the University, and thus, wireless communication options are preset and not all possibilities are authorized. The following alternatives have been considered:

- LAN: requires multiple antennas. Not available/authorized.
- *eduroam* (education roaming):[3] although there are several antennas distributed all over the building, in its current state the connection suffers multiple interruptions and is not reliable at all.
- 3G Mobile Wi-Fi with prepaid SIM cards: occasional communication interrupts may occur, but they are rare.

[3] http://www.eduroam.org.

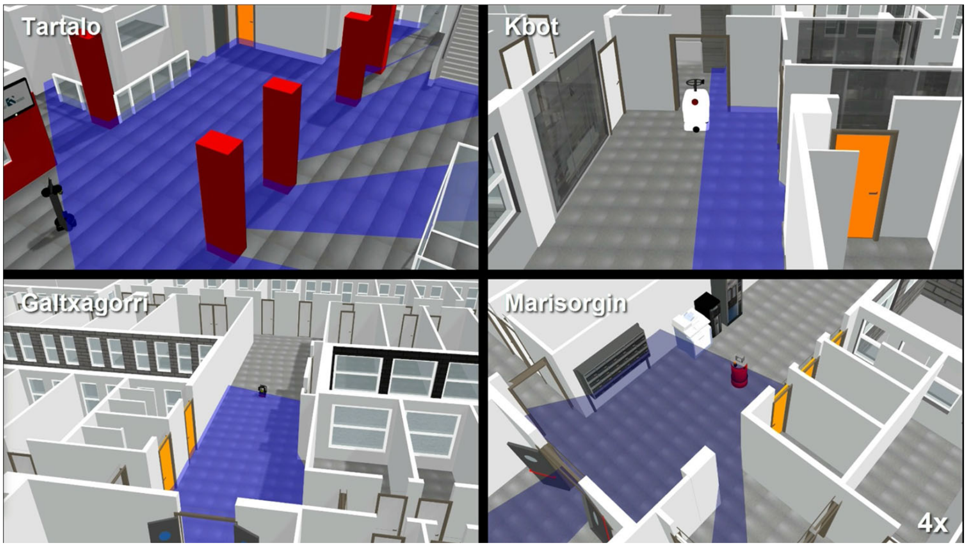


Fig. 6 Screenshots of the simulated robot/environment pairs

Hence, this is the final choice we made. Each robot now uses a 3G Modem that connects to the internet.

With this set-up, considering that the robots share information among each other, they need to have an accessible IP address assigned so that messages can be received properly. This IP address should not be changed while the system is operating, and preferably, neither after each session.

However, the used SIM cards do not allow static IPs; IPs can change at any time, and even more, those IPs are not externally accessible. So after discarding the choice of hiring a static IP internet connection with a telephone provider, we decided to set up a virtual private network (VPN) that enables assigning static IPs to the robots.

As the final solution, we managed to get four static IP addresses from our university's VPN service, creating an LDAP[4] account for each robot and using it when connecting to VPN. This way, an accessible static IP address, in which the ROS master will be running, is set every time the robots initialize.

7.3 System configuration

Each robot has its own launch file, containing several parameters that must be set up before executing the system, and the initialization of the required modules (Fig. 5).

Among these parameters, the global configuration of the system must be set up. Each robot needs to know the floor it will be working on, and for the other robots participating in the task, their name, IP address and the floor they will be working on. It is also possible to have a floor without any robot, which means that the user will have to find the destination by himself/herself. In our case, all of our robots are prepared to navigate in any floor of the Faculty of Informatics of San Sebastian, so changing their location can be easily carried out.

Concerning robot communication, the launch files of each robot describe the configuration of each multimaster node and this can also be set up accordingly.

8 Experimental results

This section describes the different experiments performed to evaluate the robustness of the *GidaBot*. Experiments have been performed in both simulated and real robot/environment systems.

8.1 *GidaBot* in simulation

The goal of the simulated robot/environment system is twofold:

1. Evaluate the soundness of the application without suffering from issues like battery life and Wi-Fi communication.
2. Offer a tool to other researches/developers to test the application.

With that aim, each floor of the Faculty of Informatics and each robot have been modelled using Blender[5] and, afterwards, these models have been integrated in four different Gazebo[6] worlds. As a result, a complete simulation environment is available together with the *GidaBot* system. Figure 6 shows screenshots of each simulated floor/robot pair.

[4] Lightweight Directory Access Protocol.

[5] https://www.blender.org/.

[6] http://gazebosim.org/.

Table 2 Long duration tour

#	Floor	Place description
1	0	Student Council
2	3	IXA Lab
3	2	C. Rodriguez's Office
4	1	Dean's Office
5	2	J. Abascal's Office
6	3	B. Sierra's Office
7	0	Computer Technicians 1
8	1	Ada Lovelace Auditorium
9	2	Seminar Room 2.3
10	0	Wi-Fi Room
11	1	Laboratory 1.3
12	3	RSAIT Lab
13	2	O. Arbelaitz's Office
14	1	Secretary's Office
15	0	Copy Shop

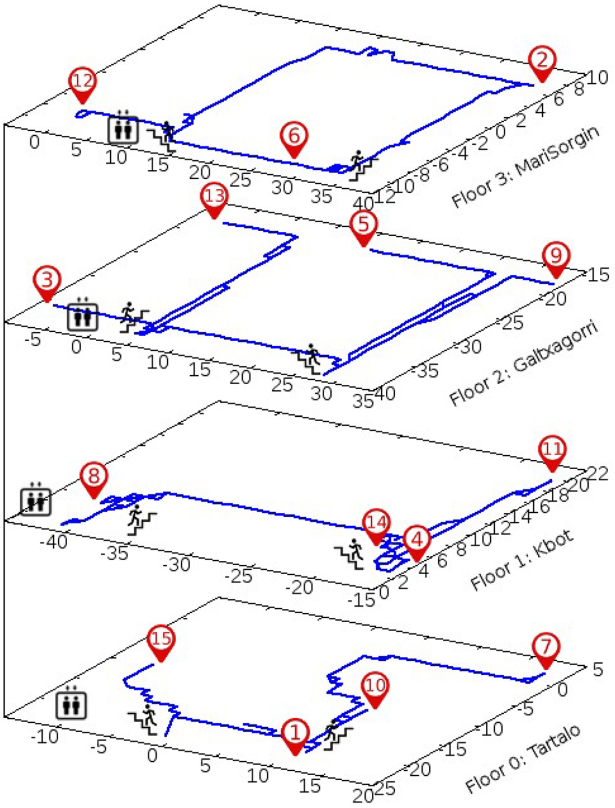


Fig. 7 Path followed by each robot on each floor during the simulated guided tour

A long tour of 15 randomly selected goals covering the whole faculty was performed and recorded in which the goal sequence was randomly chosen (see Table 2).

Table 3 Exchanged goal messages among the robots during the guided tour

Robot	Received	Processed
Tartalo	15	7
Kbot	15	8
Galtxagorri	15	8
MariSorgin	15	6

Table 4 Real-world tour

#	Floor	Place description
1	0	Technicians' Lab
2	0	Caretakers' Office
3	1	Secretary's Office
4	1	Ada Lovelace Auditorium
5	2	M. Larrea's Office
6	2	K. Sarasola's Office
7	3	I. Irigoien's Office
8	3	RSAIT Lab

Table 5 Collected data

Robot	Time (s)	Dist. (m)	Mean vel. (m/s)
Tartalo	353	86	0.24
Kbot	81	32.2	0.4
Galtxagorri	172	60	0.35
MariSorgin	132	35.8	0.27
Tour	738	215	0.29

Four standard PCs (i5 with 4 GB RAM) available at the laboratory were used. Instead of the individual 3G modems, all the PCs were connected to the same Wi-Fi router that gave internet access. Two people were responding to the interface queries during the experiment that lasted 1 h approximately.

Figure 7 shows the paths followed by each robot on each floor during the guided tour. Table 3 summarizes the number of goal messages exchanged among the robots in the course of the tour. Notice that for each floor change, two messages are processed, but only if the robot is involved in the operation: one for picking up the visitor, and one for releasing the visitor. Only the number of messages received and processed by each robot has been considered.

A video of the simulated system running a tour is also available at RSAIT's YouTube channel.[7]

[7] https://www.youtube.com/channel/UCT1s6oS21d8fxFeugxCrjnQ.

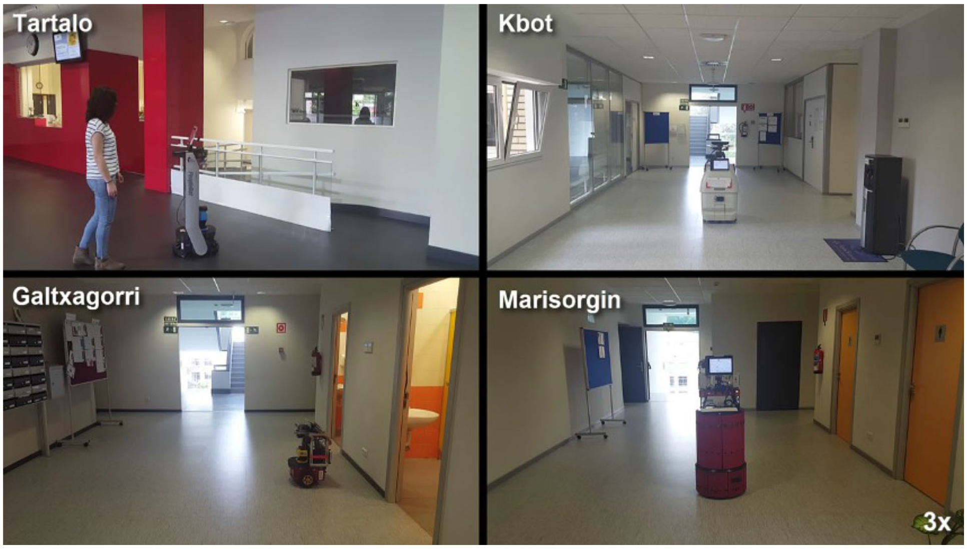


Fig. 8 Snapshot of the real system during a guided tour

8.2 GidaBot in the real world

In order to show the system's performance in the real world, we set up a tour of eight randomly chosen goals that covered the main four floors of the Faculty of Informatics (Table 4). An untrained user was asked to follow the guided tour in a working day during students vacation period.

Information about the required time and the travelled distance to successfully complete the tour has been recorded during the whole process (Table 5). The mean linear velocity varies depending on the characteristics of the environment, navigational capabilities and configuration parameters of each robot. Note that the time and distance have only been taken into consideration when the user was with the robot.

In addition, a video available at RSAIT's YouTube channel[8] shows the whole tour recorded during the guided tour. The main frame is divided into four subwindows, one per floor and robot. The upper left subwindow corresponds to the zeroth floor and *Tartalo*. The upper right one corresponds to the first floor and *Kbot*. The lower left corner corresponds to the second floor and *Galtxagorri*. And the lower right subwindow corresponds to the third floor and *MariSorgin*.

Figure 8 shows a snapshot of the video.

The video also shows the pop-up windows displayed to the user at the different steps of the tour, in order to measure the kind of interaction the user has with the robots. As the user chooses to reach upper floors by the stairs, it can be appreciated how the next robot in each phase moves to the stairs when it receives the message from the currently acting robot, in order to meet the user and continue with the tour.

As mentioned before, the configuration of the system can be adapted to a different set-up. The code is available at RSAIT's GitHub.[9]

In addition, the application has been used for several times since 2016 in the open door event hold at our faculty every year. About 100 candidate students come every year to visit the facilities and divided into groups they follow a tour in which our robots show them the most interesting rooms and places of the faculty. Figure 9 shows Kbot and Marisorgin making a guided tour with bachelor students in the first and fourth floors of the faculty, respectively.

9 Conclusions and further work

The GidaBot system described in this paper is an application to set up and run multiple robots in tour-guiding tasks over multi-floor environments. The developed guide system makes use of a robust inter-robot communication among different mobile platforms and allows them to carry out guiding tasks along the different floors of the building. The application also includes a graphical user interface that helps the user interact with the robot in an intuitive manner.

It is important to emphasize the heterogeneity of the robot team where the application is being tested. But also the fact that an standard software framework is being used. These two features make possible to conclude that it is not a tour-guide system developed just for a specific robot/environment system, but that it is applicable (with some adjustments) to other instances. This would require some work such as adapting the maps shown on the tabs of the interface and the coordinates of the interesting locations of the new environment.

[8] https://www.youtube.com/watch?v=i1UtxrGieks.

[9] https://github.com/rsait.

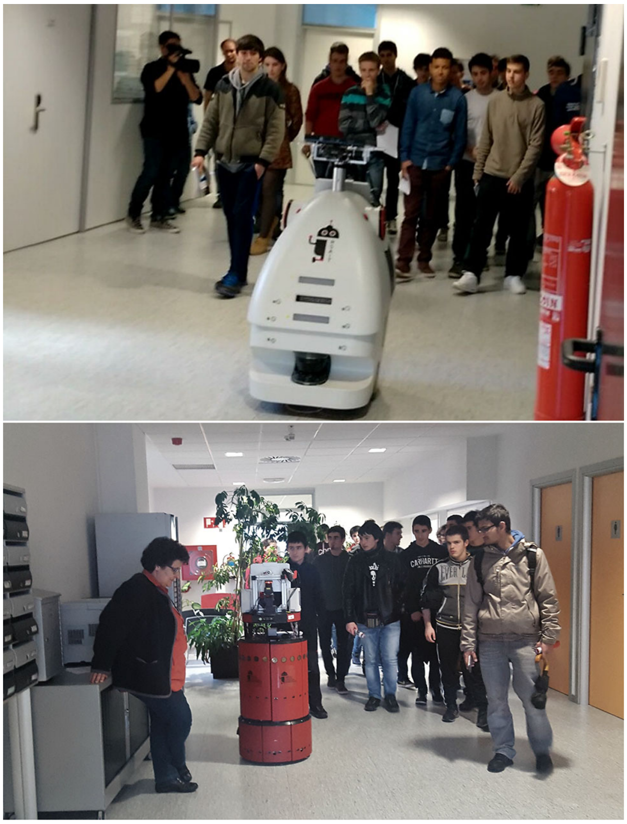


Fig. 9 Snapshots of guided tours with bachelor student visitors

The system relies on the ROS navigation stack that needs to be tuned on each robot. The performance of the navigation stack varies depending on the hardware. Some parameters could be better tuned in some cases to avoid bizarre behaviour such as giving several turns on the spot when localization fails. Also, bothering the robot, blocking its way and preventing robust localization can entail a failure. Further versions of this ROS package may overcome this problem. Visitors also must be advised to keep at the back of the robot to minimize sensor uncertainty and overcome localization problems. This should be explicitly expressed by a pop-up window in the GUI.

Regarding robot communication, for the application to be run, each robot must have a known static IP and all the robots must share the network. Though the used network resources are irrelevant for the application itself, if the system is going to be used in a public building as it is the case, in a near future, it would be desirable to avoid the use of prepaid SIM cards and make the system run with *eduroam*.

A relevant issue concerns the *multimaster* package used for context sharing among robots. ROS was designed for single robot systems, and the *multimaster* package is a side solution. ROS 2.0 is being designed to overcome this problem offering the possibility of building multirobot systems. But ROS 2.0 version is still in a beta stage, and the migration from ROS 1.0 to 2.0 promises to be anything but trivial with many incompatibilities among packages.

As further work, the system can be tuned in several aspects. The most immediate issue is to enhance the usability of the GUI, offering to the visitors general instructions about the functioning of system. In larger environments, two or more robots could share a floor, and then, robot availability should also be managed. But depending on the field of application—and it is the case of tour-guide robots—service robots should be able to interact with users in a human like manner and show social skills. Currently, we are integrating a face recognition system so that single visitors are recognized while sharing a goal among robots. User images must be shared among robots, but first, the degree of acceptability by the potential users must be measured. Besides, we intend to extend the system so that the robots respond to spoken orders.

Acknowledgements The funding was provided by Ministerio de Economía, Industria y Competitividad, Gobierno de España (Grant No. TIN 2015-64395-R) and Eusko Jaurlaritza (Grant No. IT900-16).

References

1. Faber F, Bennewitz M, Eppner C, Görög A, Gonsionr C, Joho D, Schreiber M, Behnke S (2009) The humanoid museum tour guide Robotinho. In: IEEE international symposium on robot and human interactive communication (RO-MAN), pp 891–896
2. Fischinger D, Einramhof P, Papoutsakis K, Wohlkinger W, Mayer P, Panek P, Hofmann S, Koertner T, Weiss A, Argyros A et al (2016) Hobbit, a care robot supporting independent living at home: first prototype and lessons learned. Rob Auton Syst 75:60–78
3. Forlizzi J, DiSalvo C (2006) Service robots in the domestic environment: a study of the roomba vacuum in the home. In Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction. ACM, pp 258–265
4. Gross H.-M, Boehme H, Schroeter C, Müller S, König A, Einhorn E, Martin C, Merten M, Bley A (2009) TOOMAS: interactive shopping guide robots in everyday use-final implementation and experiences from long-term field trials. In: International conference on intelligent robots and systems (IROS). IEEE, pp 2005–2012
5. Hristoskova A, Agüero CE, Veloso M, De Turck F (2012) Personalized guided tour by multiple robots through semantic profile definition and dynamic redistribution of participants. In: Proceedings of the 8th international cognitive robotics workshop at AAAI-12, Toronto, Canada
6. International Federation of Robotics. IFR. http://www.ifr.org/service-robots/. Accesed 24 Jan 2017
7. Kanda T, Shiomi M, Miyashita Z, Ishiguro H, Hagita N (2009) An affective guide robot in a shopping mall. In: Proceedings of the 4th ACM/IEEE international conference on Human-robot interaction. ACM, pp 173–180
8. López J, Pérez D, Santos M, Cacho M (2013) Guidebot. A tour guide system based on mobile robots. Int J Adv Rob Syst 10:381
9. López J, Pérez D, Zalama E, Gomez-Garcia-Bermejo J (2013) Bellbot—a hotel assistant system using mobile robots. Int J Adv Rob Syst 10:40
10. Marder-Eppstein E, Berger E, Foote T, Gerkey B, Konolige K (2010) The office marathon: robust navigation in an indoor office

environment. In: International conference on robotics and automation

11. Mukai T, Hirano S, Nakashima H, Kato Y, Sakaida Y, Guo S, Hosoe S (2010) Development of a nursing-care assistant robot RIBA that can lift a human in its arms. In: 2010 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 5996–6001
12. Neobotix. Neobotix—robotics and automation. http://www.neobotix-robots.com. Accessed 24 Jan 2017
13. Omron Adept MobileRobots. Omron adept Mobile robot systems for research and development. http://www.mobilerobots.com/Mobile_Robots.aspx. Accessed 24 Jan 2017
14. Pieska S, Luimula M, Jauhiainen J, Spiz V (2013) Social service robots in wellness and restaurant applications. J Commun Comput 10(1):116–123
15. Rodriguez I, Jauregi E, Astigarraga A, Ruiz T, Lazkano E (2016) Robot operating system (ROS). The complete reference, chapter. In: Standardization of a heterogeneous robots society based on ROS. Number 625 in Studies in computational intelligence. Springer, pp 289–313
16. Rosenthal S, Biswas J, Veloso M (2010) An effective personal mobile robot agent through symbiotic human-robot interaction. In: Proceedings of the 9th international conference on autonomous agents and multiagent systems: vol 1–vol 1. International Foundation for Autonomous Agents and Multiagent Systems, pp 915–922
17. Shiomi M, Kanda T, Ishiguro H, Hagita N (2007) Interactive humanoid robots for a science museum. IEEE Intell Syst 22(2):25–32
18. Susperregi L, Fernandez I, Fernandez A, Fernandez S, Maurtua I, de Vallejo IL (2012) Interacting with a robot: a guide robot understanding natural language instructions. In: Ubiquitous computing and ambient intelligence. Springer, pp 185–192
19. Thrun S, Bennewitz M, Burgard W, Cremers AB, Dellaert F, Fox D, Hahnel D, Rosenberg C, Roy N, Schulte J et al (1999) Minerva: a second-generation museum tour-guide robot. In: 1999 IEEE international conference on robotics and automation, 1999. Proceedings, vol 3. IEEE
20. Thrun S, Burgard W, Fox D (2005) Probabilistic robotics. MIT Press, Cambridge
21. Trahanias P, Burgard W, Argyros A, Hähnel D, Baltzakis H, Pfaff P, Stachniss C (2010) TOURBOT and WebFAIR: web-operated mobile robots for tele-presence in populated exhibitions. IEEE Rob Autom Mag 12(2):77–89
22. Troniak D, Sattar J, Gupta A, Little JJ, Chan W, Calisgan E, Croft E, Van der Loos M (2013) Charlie rides the elevator–integrating vision, navigation and manipulation towards multi-floor robot locomotion. In: 2013 international conference on computer and robot vision (CRV). IEEE, pp 1–8
23. Veloso M, Biswas J, Coltin B, Rosenthal S (2015) Cobots: robust symbiotic autonomous mobile service robots. In: 24th international joint conference on artificial intelligence, pp 4423–4428
24. Veloso M, Biswas J, Coltin B, Rosenthal S, Brandão S, Mericli T, Ventura R (2012) Symbiotic autonomous service robots for user-requested tasks in a multi-floor building. In: IEEE/RSJ international conference on intelligent robots and systems
25. Wang C, Savkin AV, Clout R, Nguyen HT (2015) An intelligent robotic hospital bed for safe transportation of critical neurosurgery patients along crowded hospital corridors. IEEE Trans Neural Syst Rehabil Eng 23(5):744–754
26. Wurman PR, D'Andrea R, Mountz M (2008) Coordinating hundreds of cooperative, autonomous vehicles in warehouses. AI Mag 29(1):9–20