SRS

Multi-Role Shadow Robotic System for Independent Living

Small or medium scale focused research project (STREP)

DELIVERABLE D 5.1

Name of the Deliverable – SRS UI_PRI, UI_LOC and core system implemented

Contract number :	247772
Project acronym :	SRS
Project title :	Multi-Role Shadow Robotic System for Independent Living

Deliverable number :	**D5.1**
Nature :	P – Prototype
Dissemination level :	PU – PUBLIC
Delivery date :	31-01-2012 (month 24th)

Author(s) :	Rafa López Tarazón, Nayden Chivarov, Gernot Kronreif, Renxi Qiu
Partners contributed :	ROB, BAS, IMA, CU
Contact :	rlopez@robotnik.es

The SRS project was funded by the European Commission under the 7th Framework Programme (FP7) – Challenges 7: Independent living, inclusion and Governance

1 INTRODUCTORY SECTION: PURPOSE OF DOCUMENT AND CONTENTS

The purpose of this document is to present the current status of the User Interface for Private Remote Operator “UI_PRI” and User Interface for Local User “UI_LOC”.

Both interfaces are systems that have been developed throughout the SRS project in order to control the Care-O-bot 3 from Fraunhofer IPA and provide practical support for local elderly users of the robot as well as to ensure a proper interaction between private care-giver and the robot.

At “UI_LOC” the user interface focuses on usability of elderly users, privacy and safety assurance. It allows the local elderly user to start a simple command to the robot.

At “UI_PRI” the user interface focuses on issuing high level control commands to control the behavior of the robot. This interface allows real time visualization from the cameras of the robot and some manual intervention when the autonomous mode fails.

The contents of both user interfaces and corresponding interaction patterns and technologies have been designed in WP2

2 UI_PRI

2.1 PREREQUISITES

The UI_PRI has been developed to be run over an iPad or iPad2.

The software requirement is Operative System iOS 4.2 or later

The following packages must be installed and running either on Care-O-Bot or in simulation

Name	Status	Description
Srs_decision_making	Mandatory	Decision making service
srs_mixed_reality_server	Mandatory	Mixed reality server
srs_human_sensing	Optional	Human sensing using robot sensors
rosbridge	Mandatory	Communication between iPad user interface and Care-O-Bot

2.2 CONFIGURATION

Next figure show an overview of the SRS application run on an iPad:

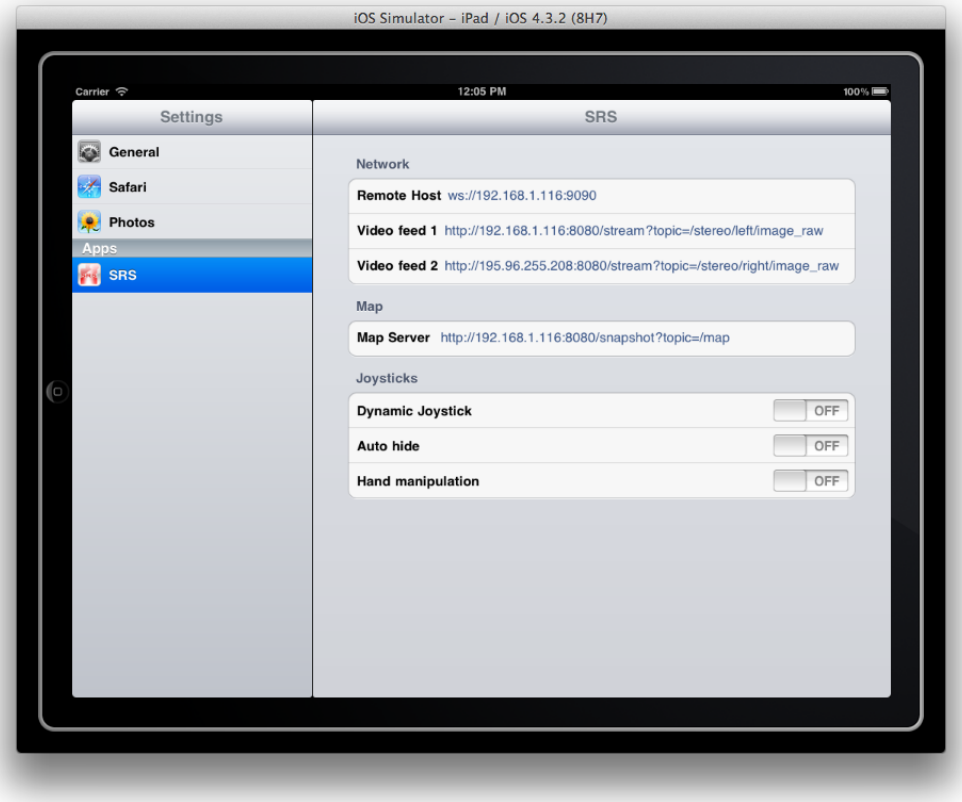


Figure 1: SRS application

From this initial screen we have access to the network setup:

Remote Host	Remote Host specifies the address of the rosbridge package ws://IP:9090 where the IP is the address of the remote machine
Video feeds	Video feeds specifies the address of video stream provided by Mixed Reality Server http://IP:8080/stram?topic=VIDEO_STREAM where IP is the address of the remote machine and VIDEO_STREAM is one of the provided video streams (for example /stereo/left/image_raw or /stereo/right/image_raw
Map server	Map Server specifies the address of the map server provided by Mixed reality server http://IP:8080/snapshot?topic=/map where IP is the address of the remote machine running Mixed reality server

and configuration of joysticks:

Dynamic Joysticks	Joysticks position can be dynamically adjusted on screen. To move joystick to a new position turn this ON, then hold on a new position for 2 seconds
Auto hide	Dim joystick controls when not used
Hand manipulation	Hand manipulation joystick control

2.3 ROBOT STATUS

Robot status screen provides the following information

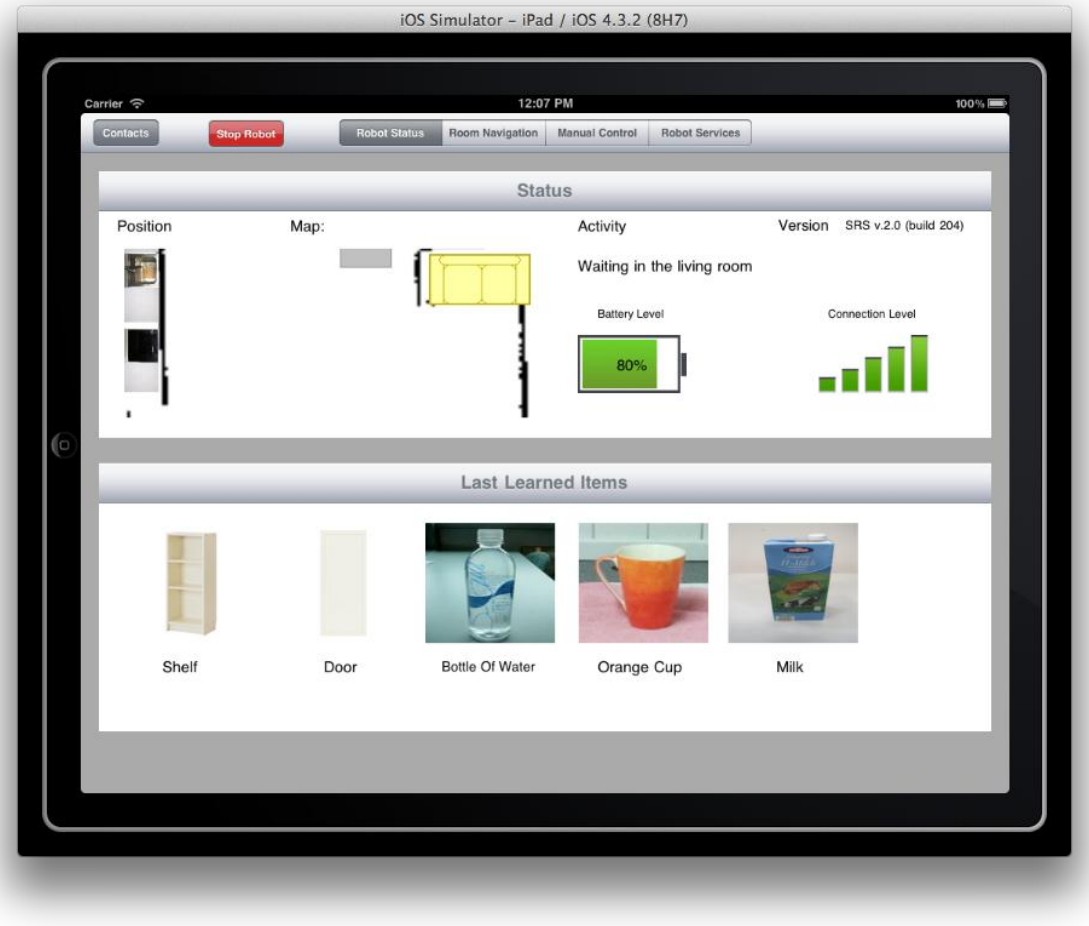


Figure 2: main robot status screen

The screen provides overview of the current active map as well as robot position

Figure 3: robot position in the current map

There is also an activity log that shows the latest activity or notification message from the robot (e.g. Activity: “Waiting in the living room”)

Next table shows the power meter that provides information about the robot battery level:

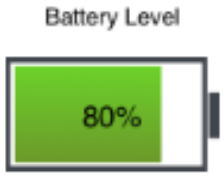
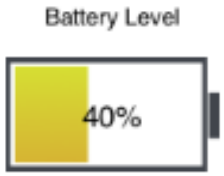
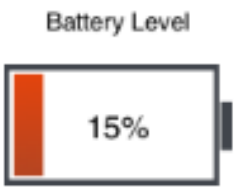
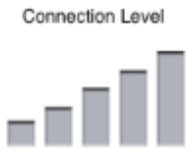
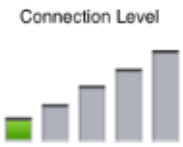
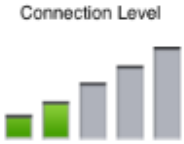
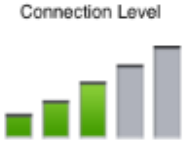
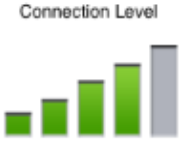
Value	Description
	Battery level above 50% is shown in green
	Battery level between 30% and 50% is shown in yellow
	Battery level below 30% is shown in red

Table 1: Battery level description

As for the power meter, the application provides information about the connection level with the robot:

Level	Description
	0 bars - no connection to the robot
	1 bar - very weak connection to the robot, video streams, db updates will not function
	2 bars - weak connection to the robot, video streams might work with big delay
	3 bars - average connection, video streams, other functionality might work with delay
	4 bars - good connection, video streams, other functionality might work with minor delay

Level	Description
	5 bars - excellent connection, video streams, other functionality will work without delay

Table 2: Connection level description

The application is able to show the list of the latest learned objects or the latest objects downloaded from the objects database

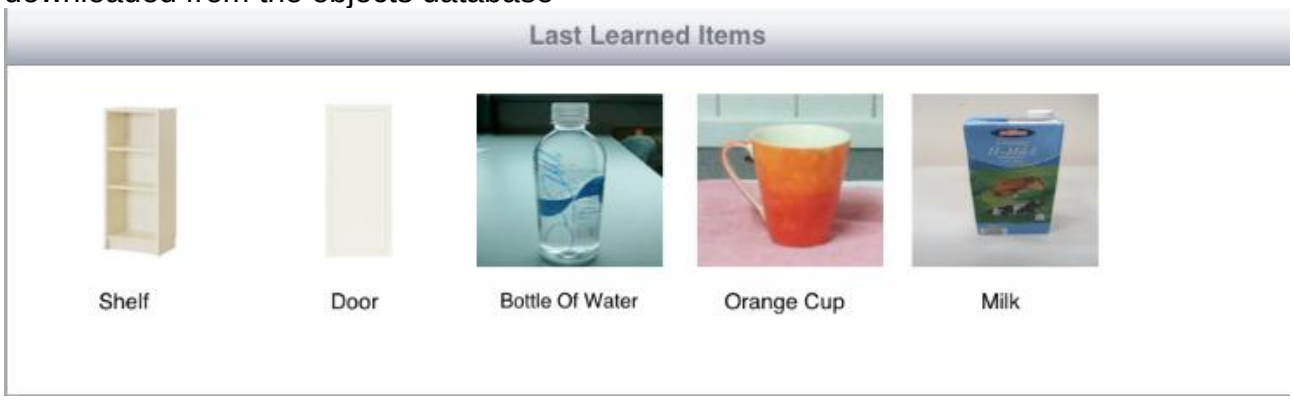


Figure 4: Last learned objects list

2.4 ROOM NAVIGATION

Room navigation screen provides information about current location of the robot, surrounding objects, human detected on the map as well as video stream from the robot camera.

All robot tasks can be executed from this window, including sending to location, detecting and grasping an object, etc. The map is real time augmented with objects detected and stored into objects database.

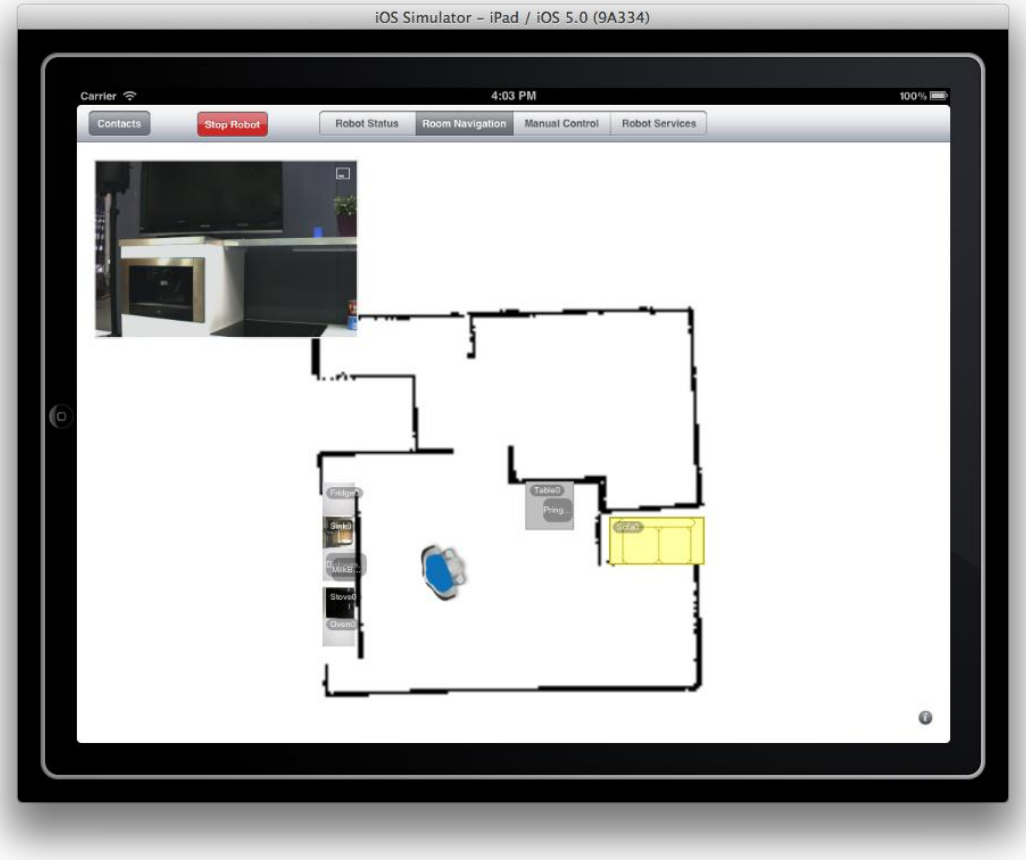


Figure 5: navigation main screen

The Video stream box shows the current video feed from the robot.
To move the video stream window around the screen hold one finger over it for about 3 seconds until it pops up then move the finger around, dragging the window to the desired location.
To change the size of the window click on the size button in the top right corner, the resize it using the anchor points.

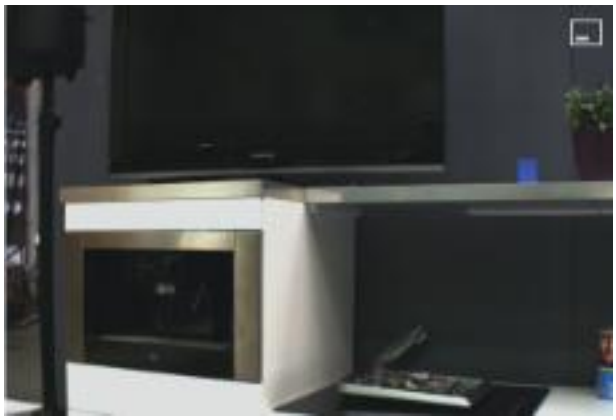


Figure 6: video stream screen box

To send the robot to a desired location click on the map and then confirm the location. In case of accidentally clicking on map, cancel request.

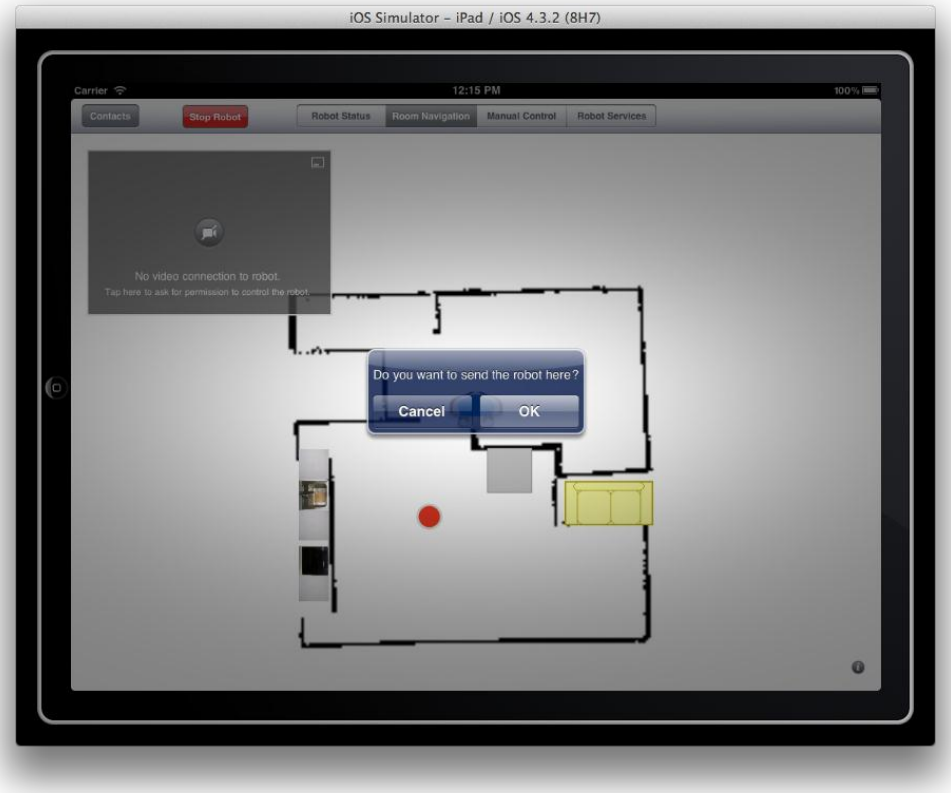


Figure 7: send robot to desired location screen

2.5 MANUAL CONTROL

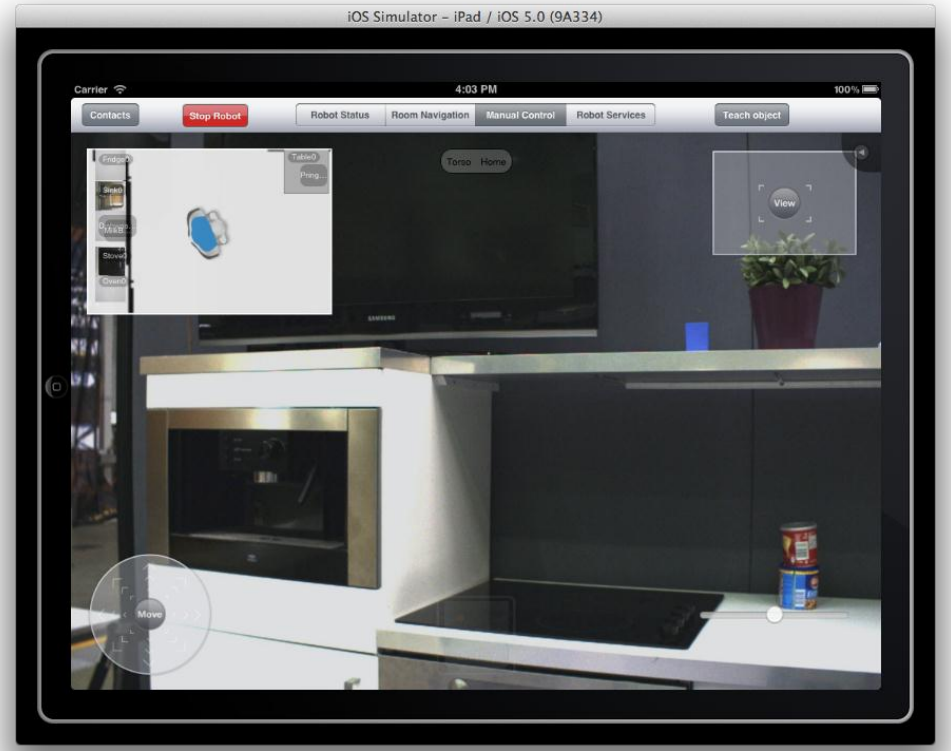


Figure 8: Manual control example screen

The manual control features of the UI_PRI allow the user to move the robot manually by using the next control buttons:

Button	Description
	Navigation joystick: Allows driving the robot in all directions
	Turn control: Allows turning the robot around itself in clockwise and counter clockwise directions

Button	Description
	Torso control: Setting the robot torso in one of the predefined positions - Home, Front and Back
	Camera control: Controlling camera position
	Tray control: Outlines if there are objects on the tray

Table 3: Manual control buttons

2.6 ROBOT SERVICES

Robot services screen allows the execution of tasks, editing available tasks as well as creating new ones, and executing actions on available objects.

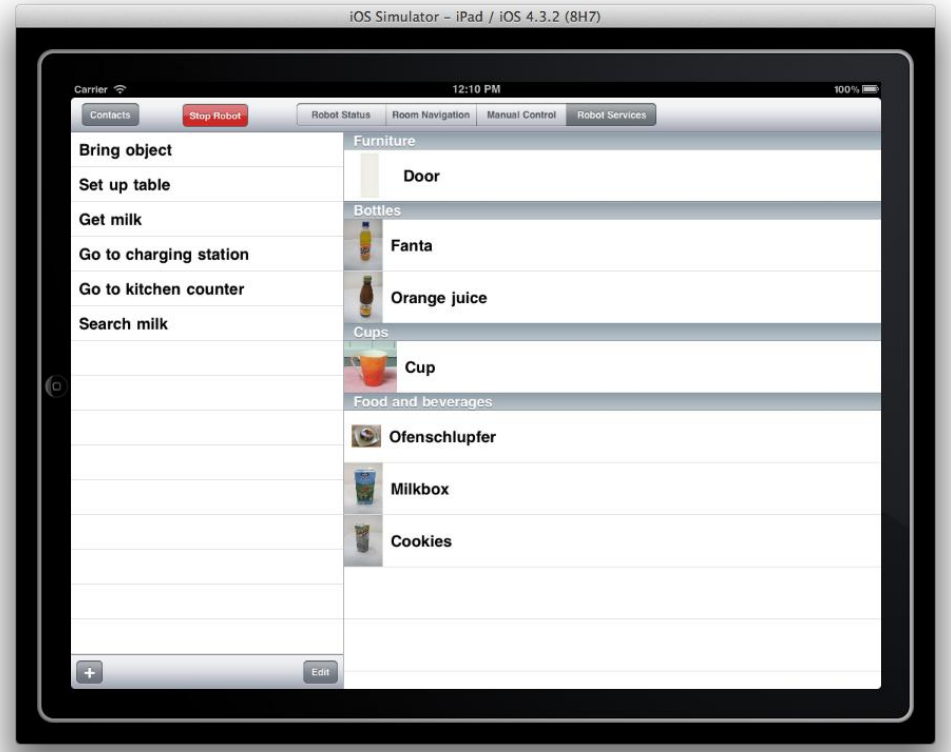


Figure 9: Robot services screen

In order to execute a task, we just select one of the available tasks on the left panel (e.g Bring object, Set up table, Get milk…) and execute it.

To execute an action on object, select the object and then select one of the available actions - Get, Deliver, Move, Detect, Grasp.

Note that available actions menu is context sensitive and not all actions are available on a given object.

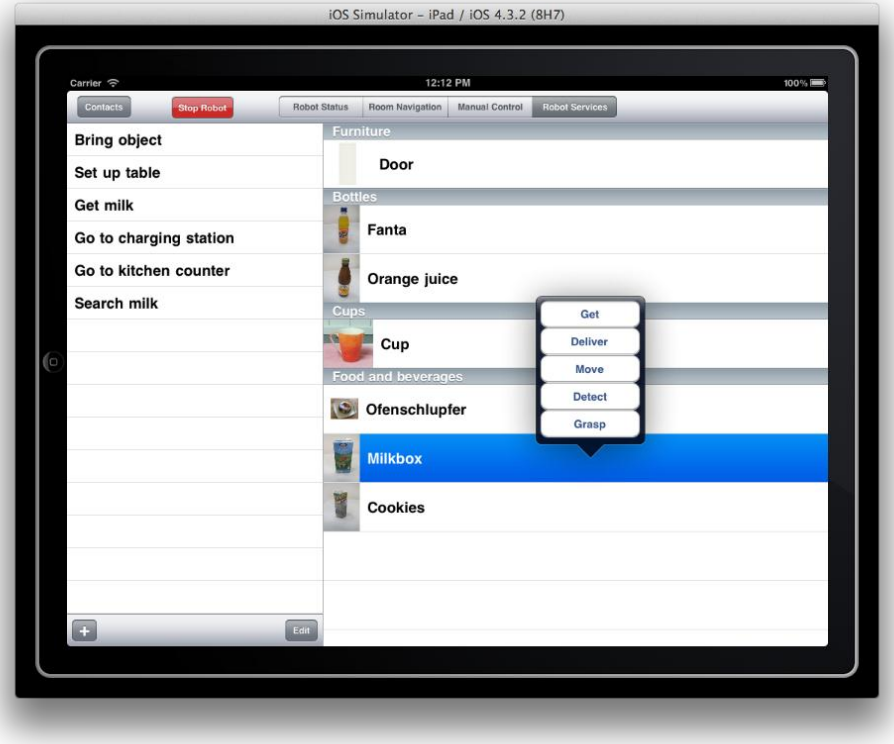


Figure 10: Task execution screen

To create a new task click “+” button on the task panel and enter the following:
Task name-->Action to be performed --> Object to be used -->Location of the task

Figure 11: New Task creation screen

3 UI_LOC

The main task for UI_LOC is to start robot services. According to the requirements (cf deliverable D2.2 for details) the device for UI_LOC should be small and mobile and thus the decision for the platform to be used was a standard internet tablet. Due to the current market situation we further have decided to choose an Android-based device for the UI. For the present prototype, selection finally was about the following system:

Hardware: ARCHOS 43 internet tablet
Operating system: Gingerbread: Android 2.3.x (API Level 9)

Implementation of the GUI was done using Eclipse Indigo, ADT Plugin, Android SDK.

3.1 General Settings

The implementation uses ROS-services and ROS-actionclients. The ROS-actionclients are being used to send a high-level command (e.g. “Bring me an object”) and to receive status messages during the execution. The ROS-services are being used to connect to the Household Object Database in order to update the trained objects and additional information (e.g. icons).

Configuration of UI_LOC is basically done via two XML-files – one containing all screen commands and one other containing the content (e.g. available objects from Household Object Database, pre-defined room locations, etc). This allows an easy change of the UI_LOC language (will be changed accordingly by changing the language on the tablet itself) as well as defines an appropriate interface between UI_LOC and any other software tool for system setup. At the moment UI_LOC is available in three languages – English, German and Italian.

```
<?xml version="1.0" encoding="UTF-8"?>
<srs>
  <rooms>
    <room name="Livingroom">
      <place name="Couch" id="0"></place>
      <place name="Armchair" id="1"></place>
      <place name="Small table" id="2"></place>
    </room>
    <room name="Kitchen">
      <place name="Table" id="3"></place>
      <place name="Chair" id="4"></place>
      <place name="Workspace" id="5"></place>
```

```
    </room>
    <room name="Bathroom">
      <place name="Basin" id="6"></place>
      <place name="Tub" id="7"></place>
      <place name="Shower" id="8"></place>
    </room>
    <room name="Bedroom">
      <place name="Bedside table" id="9"></place>
      <place name="Dresser" id="10"></place>
      <place name="Dressing table" id="11"></place>
    </room>
  </rooms>
  ...
```

Figure 12: Configuration of UI_LOC via XML file

3.2 ROBOT SERVICES

As mentioned above main purpose of UI_LOC is to call robot services. The main screen of the interface thus allows selecting this task (see figure 13).

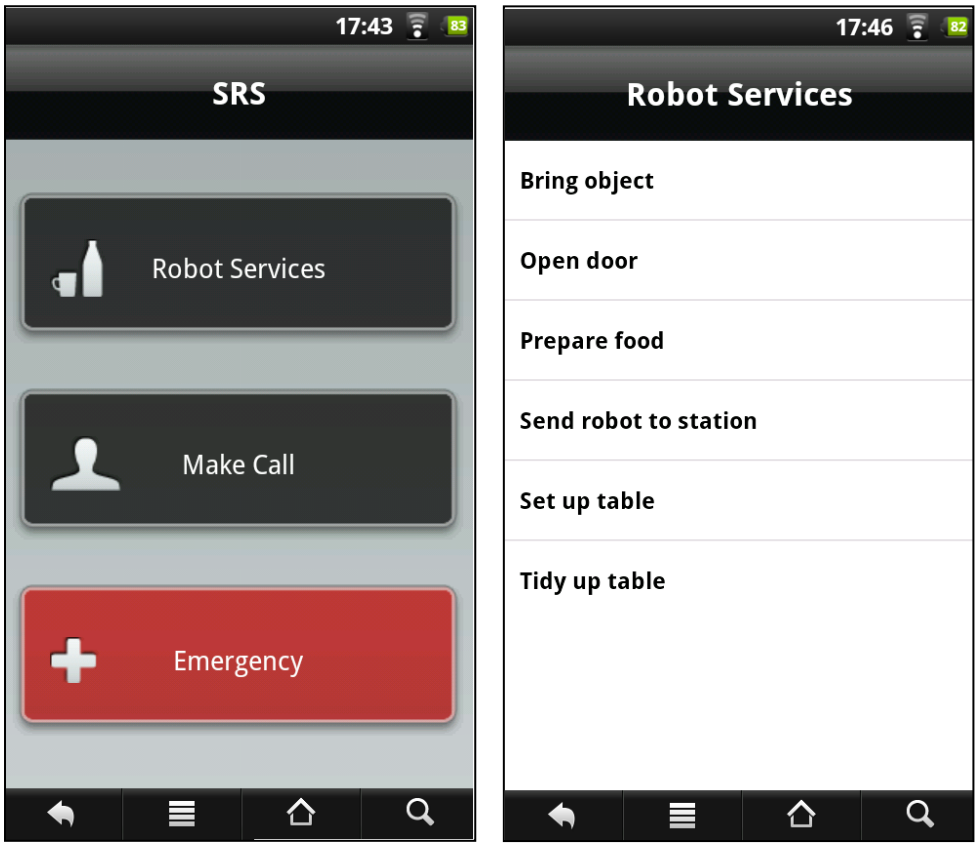


Figure 13: UI_LOC starting screen; Selection of desired robot service

For the main robot service -- “Bring Object” – there are three windows for definition of the service related parameters. The “Bring Object” main screen asks the user to select both the desired object as well as the target place for this fetch&carry task (see pictures below). The screens for object selection as well as the screen for selection of target place are structured into groups (e.g. Books, Bottles, etc for objects) in order to allow easy selection. After definition of the parameters the “Bring Object” screen is showing the final command in “natural form”: `Robot should bring "Water Bottle" to place "Couch in Living Room".`

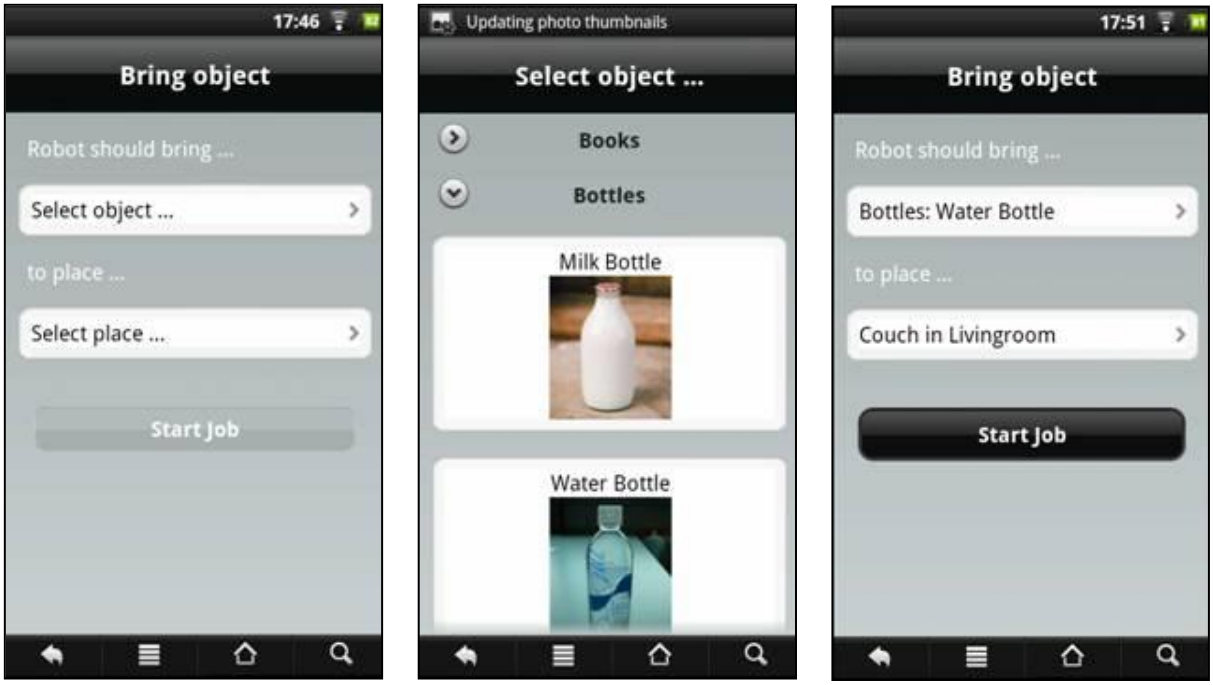


Figure 14: UI_LOC screens for fetch & carry task: (1) start screen, (2) selection of desired object, (3) selection of target place

After start of the service the robot is trying to execute the task. As main parts of task execution are out of sight for the local user there are intermediate progress messages in order to inform the user about the current state. This screen also shows a “Stop”-Button in order to bring the robot system to an immediate halt. If there is any problem during operation, the interface shows a screen with different options for solving this exceptional case or to cancel the current operation.

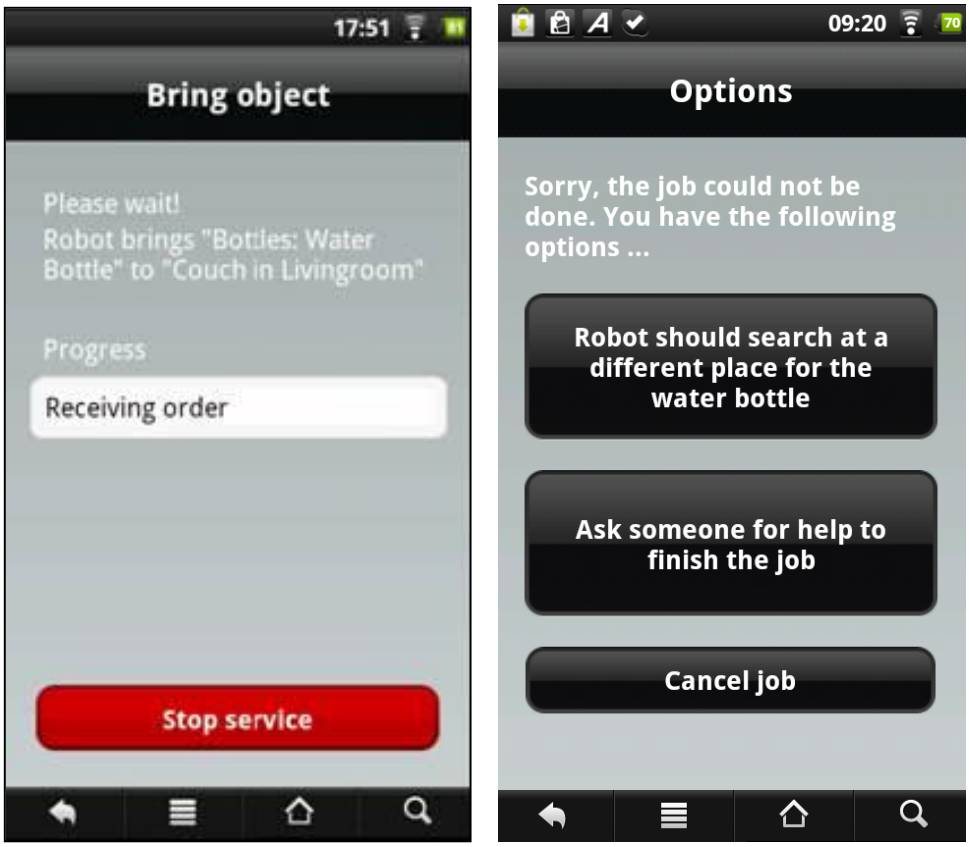


Figure 15: Information screens during operation; (1) display of progress; (2) selection between different options for exceptional cases

After successful finishing of the service the GUI again shows the selection screen for robot services. The GUI also uses the standard Android-buttons for navigation. The “Home”-button is programmed to jump back to the SRS main menu – the “Back”-button shows the previous screen. For some of the screens the “Back”-button is disabled in order to follow the defined flow of actions (e.g. during display of the “Bring Object” progress screen).

3.3 OTHER SERVICES

There are two other services available in SRS – “Emergency Call” and “Make a Call”.

The idea behind the “Emergency Call” is that UI_LOC is starting a video-communication with a pre-defined/preferred person. At the moment the system automatically proposes to call the “24hrs-Service Center” (i.e. the professional remote operator). In addition to the call the robot should move to the calling person. As the current version of SRS / UI_LOC does not allow self-localization of UI_LOC in the household environment, there is an additional selection box in order to define the target location (see pictures below). This screen might be removed from the GUI once such a self-localization function I available.

Figure 16: Screens for emergency call

Another functionality of the UI_LOC is to connect to a video-call. Based on the address book of the device there is a selection list shown by the GUI after selection of the “Make a Call” service. After selection of a contact the GUI automatically opens a SKYPE connection. After the call, the SKYPE windows closes and the system switches back to the SRS main screen.

Figure 17: Communication with UI_PRI and UI_PRO via SKYPE

3.4 IMPLEMENTATION VS REQUIREMENTS

The following table shows the comparison between the realized functionality of the present version of UI_LOC and the requirements defined in deliverable D2.2.

UI_LOC1	Wireless communication with robot: the used device for UI_LOC is connected via WLAN (and via a base station) to the robot system
UI_LOC2	Small, lightweight and mobile: UI_LOC is installed at a small internet tablet (ARCHOS 43)
UI_LOC3	Intuitive interface – preferably without additional devices: user interaction is via touch-screen only; GUI has been successfully evaluated in user trials
UI_LOC4	Wireless communication with PRI or PRO: communication with UI_PRI and UI_PRO is via SKYPE – running as an external task but started via UI_LOC.

UI_LOC5	Accept/deny remote operation; Information in case of PRI or PRO desiring to log in: will be included for next UI update
UI_LOC6	Selection and start of task: GUI basically consists of an intuitive menu structure – selection is supported by lists with icons.
UI_LOC7	Selection of “parameters” for a task is supported by lists with icons.
UI_LOC8	Stop action: Running task can be stopped by the GUI. Risk analysis takes into account, that this stop function is not having the same safety level than a wired emergency stop button.
UI_LOC9	“Localization” of LOC position: as the modification of the environment is not in the scope of the SRS project, position of LOC currently is being defined by an appropriate UI function.
UI_LOC10	Selection of desired object for grasping tasks: will be included for next UI update
UI_LOC11	Display of basic robot status messages: Connection status, battery status, feedback about task status, error messages are implemented
UI_LOC12	Display of control status: will be included for next UI update
UI_LOC13	Display of next robot activity: will be included for next UI update

Table 4: Functionality vs requirements UI_LOC

3.5 NEXT STEPS

The following additional features will be included in the next version(s) of UI_LOC:

- Additional screen for handing over command to remote interface (UI_PRI or UI_PRO). Such a request can be a result of an exceptional case occurring during task execution (see figure 15) or on request by one of the remote interfaces. If the local user accepts such a change of operation mode, UI_LOC will be disabled and a dedicated screen shows that the robot is in remote mode

- Screen for remote mode: See above. If the control is handed over to one of the remote interfaces, this screen signals that the robot cannot be controlled via UI_LOC and that unexpected movement of the robot and/or one of its components can occur. There will be a STOP-button always accessible in order to immediately stop the remote mode and switch back to local control.

- Screen for object selection: if the autonomous behavior cannot be continued because of problems in object recognition – e.g. there are two identical (target) objects in the search area and there must be a selection of one of the two – such

a situation can be resolved by user input at UI_LOC. In such a situation there will be a picture displayed to the GUI and the user needs to select the desired object by simply pointing to it. After resolving of the problem the operation is being continued in autonomous mode.

- Modifications due to input from user trials: increasing size of fonts, increasing size of buttons, changing “jumping address” for some navigation buttons,

4 LIST OF FIGURES AND TABLES