Ref. Ares(2020)5139414 - 30/09/2020

Deliverable 2.2

ERL Consumer Rulebook Update (b)

Project acronym:	**SCIROC**
Project number:	780086
Project title	European Robotics League plus Smart Cities Robot Competitions
WP number and title:	WP2 – ERL Consumer Service Robots
WP leader:	IST-ID

Organisation responsible for deliverable:	Associação do Instituto Superior Técnico para a Investigação e Desenvolvimento (IST-ID)
Deliverable author(s):	Meysam Basiri (IST-ID) Pedro Lima (IST-ID)
Deliverable version number:	2.0
Actual delivery date:	30/09/2020
Dissemination Type	Report
Dissemination level:	Public

Change log			
Version	**Date**	**Author**	**Reason for change**
2.0	30/09/2020	M. Basiri and P. Lima	Updated the rules of TBM1, 2, and 4, to increase the scientific and technical challenges, given the level achieved by most teams in the 2018/19 Season and during the 2019 Smart Cities Tournament. Introduced a new section in Part I on the possible adaptation of the current TBMs to handle the COVID-19 pandemic in domestic environments.

Release approval			
Version	**Date**	**Name and organisation**	**Role**

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement n° 780086.

This project is funded by the European Union

EUROPEAN ROBOTICS LEAGUE

Brought to you by SPARC

European Robotics League plus
Smart Cities Robot Competitions
Horizon 2020

SciRoc

ERL Consumer

European Robotic League for Consumer Service Robots

Latest Update: September 29, 2020

Editors: Meysam Basiri, Pedro Lima

Contributors: Aamir Ahmad, Francesco Amigoni, Iman Awaad, Jakob Berghofer, Rainer Bischoff, Andrea Bonarini, Rhama Dwiputra, Giulio Fontana, Frederik Hegger, Nico Hochgeschwender, Luca Iocchi, Gerhard Kraetzschmar, Pedro U. Lima, Matteo Matteucci, Daniele Nardi, Pedro Miraldo, Viola Schiaffonati, Sven Schneider, Meysam Basiri, Emanuele Bastianelli, Pedro Resende, João Mendes

IST-ID Associação do Instituto Superior Técnico para a Investigação e Desenvolvimento

SAPIENZA UNIVERSITÀ DI ROMA

Hochschule Bonn-Rhein-Sieg

POLITECNICO DI MILANO

Introduction

The European Robotic League (ERL) is an innovative concept for robot competitions. The ERL is composed of multiple across-the-year Local Tournaments, held in different certified test beds located across Europe, and Major Tournaments, where the competition takes place as part of a major conference or event such as RoboCup. From 2019 on, Smart City tournaments will also take place every other year in European cities. Smart City tournaments are a single annual event. During each across-the-year ERL Season, teams participate in a minimum of two tournaments (Local and/or Major) and get scores based on their performances on Task Benchmarks (TBM) and Functionality Benchmarks (FBM). A final Season score is computed for each team, using the team best two participations in the Season Tournaments, and teams are ranked based on their final score. Prizes for the top teams are awarded during the next year's European Robotics Forum (ERF). For more information, please visit:

`www.robotics-league.eu`

The **ERL Consumer Service Robots** Challenge (**ERL Consumer** for short) stems from its predecessors, the **RoCKIn@Home** Challenge, and the **ERL Service Robots** Challenge, and covers the domain of consumer service robotics, with current focus on domestic applications. This document describes the rules of the ERL Consumer Challenge for the season 2018/2019.

Note: The top performing teams from all Challenges (ERL Consumer, ERL Professional and ERL Emergency) of the ERL 2018/19 Season will qualify to participate in the Smart City tournament that will take place in Milton Keynes, United Kingdom in the Fall of 2019.

Note: This version of the ERL Consumer rule book is valid until 28.02.2019.

Contents

I: ERL Consumer in a Nutshell

1 Introduction

ERL Consumer is a competition that aims at bringing together the benefits of scientific benchmarking with the attraction of scientific competitions in the realm of consumer service robotics. The objectives are to bolster research in consumer service robotics for home applications and to raise public awareness of the current and future capabilities of such robot systems to meet societal challenges like healthy ageing and longer independent living.

Currently, ERL Consumer raises challenges in domestic environments that resemble similar challenges to be posed in smart city environments (e.g., shopping malls), such as recognising and picking objects from shelves, moving outside the home to populated areas, and/or taking elevators between floors, so as to prepare teams for the Smart City Tournaments. But ERL Consumer objectives can be adapted to other challenges, still within domestic environments or in other environments, e.g., hospitals. The COVID-19 pandemic brought new challenges to Robotics during SciRoc lifetime, and in Section 4 we explain how the current user story of Section 2 could be adapted for a COVID-19 patient quarantined at home.

2 The ERL Consumer User Story

Granny Annie is an elderly person, who lives in an in ordinary apartment. Granny Annie is suffering from typical problems of ageing people: She has some mobility constraints. She tires fast. She needs to have some physical exercise, though. She needs to take her medicine regularly. She must drink enough. She must obey her diet. She needs to observe her blood pressure and blood sugar regularly. She needs to take care of her pets. She wants to have a vivid social life and welcome friends in her apartment occasionally, but regularly. Sometimes she has days not feeling so well and needs to stay in bed. She still enjoys intellectual challenges and reads books, solves puzzles, and socializes a lot with friends.

For all these activities, ERL Consumer is looking into ways to support Granny Annie in mastering her life. The context for performing such activities by technical systems is set in the subsequent scenario description.

3 ERL Consumer Scenario

The ERL Consumer scenario description is structured into three sections: environment, tasks and their functionalities, and robots:

- The environment section specifies the environment in which the tasks have to be performed. This information is also relevant for building test beds and simulators.
- The tasks and functionalities section provides some more detail on the tasks the participating teams are expected to solve through the use of one or more robots and possibly additional equipment. Since task benchmarks are seen in ERL Consumer as a composition of functionality benchmarks, several functionalities are also introduced. This information tells teams what to prepare for.
- The robot section specifies some constraints and requirements for participating robots, which mainly arise for practical reasons (size and weight limitations, for example) and/or due to the need to observe safety regulations.

3.1 ERL Consumer Environment

The current setup of the ERL Consumer environment reflects an ordinary European apartment with all its environmental aspects, like walls, windows, doors or blinds as well as common household items, furniture, decoration and so on. The apartment depicted in Figure 1.1 serves as a guideline. More detailed specifications are given in the rule book.

Figure 1.1: Granny Annie's apartment.

The following embedded devices may be installed and are accessible within the apartment's WLAN:

- A networkable, camera-based intercom at the front door. It allows to see who is in front of the door.
- The lamps in the bedroom (e.g. on the bed stand) are accessible and controllable via network.
- The shutters on the bedroom or living room window are accessible and controllable via network.

Currently, ERL Consumer is not using these networked devices except for possible demos during Local Tournaments in the test beds equipped with them.

3.2 ERL Consumer Benchmarks

Two types of Benchmarks are defined and used in across-the-year ERL Consumer Local/Major Tournaments: Task Benchmarks and Functionality Benchmarks.

3.2.1 Task Benchmarks

The following Task Benchmarks will be available in all test beds and Local/Major Tournaments:

1. ***Getting to know my home:*** The robot must detect new changes in the environment, and update the semantic map of the apartment, within a limited time frame. The task shall be performed by the robot autonomously, though it may include moments of symbiotic interaction with a user in the apartment, e.g., to learn more about an object or a location.

At the end of the knowledge acquisition phase, the robot must show the understanding of the new environment, namely by handling one of the changed objects.

2. ***Welcoming visitors:*** The robot needs to handle a set of known and unknown visitors, who will arrive individually at the home entrance in random sequence. The robot must correctly recognize the known visitors and interact with the unknown visitors to identify them and understand their purpose of visit. The robot must then perform a set of visitor-specific behaviors that could range from manipulating and delivering of objects to guiding and following the visitors.

3. ***Catering for Granny Annie's comfort:*** This task aims at providing general purpose requests of Granny Annie inside the apartment. It focuses on the integration of different robot abilities such as human-robot interaction, navigation and robot-object interaction. After waking up in the morning, Granny Annie calls the attention of the robot by touching a button on a tablet computer. When the robot arrives, she gives a single speech command composing of three actions from the following three categories:

   (a) Finding an item or a person in the apartment
   (b) Manipulating and delivering an object
   (c) Accompanying a person inside or outside the apartment

   These three actions are randomly generated into a single command using a command generator in no predefined order. The robot is required to understand the actions and execute them accordingly.

4. ***Visiting my home:*** This TBM focuses on safe navigation in dynamic environments, people perception, obstacle avoidance and tracking and following a human. In this task, the robot should visit a set of predefined rooms, to perceive and count the number of people in each room, while avoiding and/or interacting with different obstacles based on the nature of the obstacle. Furthermore, the robot must interact and follow a previously unknown person outside the arena through a small crowd and then guide that person back to the arena.

3.2.2 Functionality Benchmarks

The following Functionality Benchmarks may be available in Local/Major Tournaments (number and type of Functionalities may vary across test beds).

1. **Object Perception:** Evaluates the ability of a robot to recognize and localize a wide range of objects. A set of objects, selected from the list of ERL Consumer items, will be positioned, one at the time, on a table located directly in front of the robot. For each object presented, the robot has to perform the following activities: i) Object detection: perception of the presence of an object on the table and association between the perceived object and one of the object classes. ii) Object recognition: association between the perceived object and one of the object instances belonging to the selected class. iii) Object localization: estimation of the 3D pose of the perceived object with respect to the surface of the table.

2. **Navigation:** Evaluates the ability of a robot to correctly, safely, and autonomously navigate in a ordinary apartment, including: the navigation in a apartment-like environment with furniture, walls, and doors, i.e. in a previously mapped area; avoiding collisions with different type of unknown obstacles, in unknown positions (not previously mapped); and navigating in the presence of people in the arena.

3. **Speech Understanding:** Evaluates the ability of a robot to understand speech commands that a user gives to a robot in a consumer environment, including all the related issues, such as background noise caused by other ongoing activities and by the robot motion. A set of spoken sentences, recorded in different environments, will be broadcast through a speaker. The robot needs to interpret the commands and produce an output according to a defined representation.

4. **People Perception:** Evaluates the ability of a robot to locate and recognize humans. Similarly to object perception, the robot must recognize a person who is standing inside a target area and to estimate the location of this person.

5. **Person Following:** Evaluates the ability of a robot to effectively *follow* a human target around and through obstacles and crowd of walking people. The benchmark requires that the robot accompanies a human target and always maintain a desired distance with this person.

6. **Grasping and Manipulation:** Evaluates the ability of a robot to correctly grasp and manipulate objects. In particular, it assesses the object picking and placing capabilities of robots suitable for many consumer applications such as setting up a dining table in domestic environments.

3.3 ERL Consumer Robots

Participating teams can use one or two robots to solve the tasks. The robots must fit through a door of 80cm width and weigh no more than 250kg. They must be fully autonomous, i.e. neither power supply via cable nor any kind of tele-operation is permitted.

Each robot must be safe to operate in the environment. Robots polluting or damaging the environment or presenting a threat to humans in the environment are not allowed to participate. A mechanism to stop the robots in case of emergencies must exist.

Robots must be properly equipped to be able to solve the tasks at least in principle. For example, it is not permitted to substitute for lack of speech understanding by entering commands on the keyboard.

Teams are not allowed to modify the environment, or to install their own embedded devices in the environment, e.g. additional sensors or actuators.

4 The ERL Consumer User Story and COVID-19

The COVID-19 pandemic brought several challenges to robot systems that act autonomously or tele-operated in situations where particular care must be taken to avoid infection of humans by contact with other infected humans and/or surfaces where the SARS-COV-2 virus may be deposited. Since this extraordinary situation, which has caused many damages worldwide, has arisen during the SciRoc project lifetime, an attempt to show how the ERL Consumer TBMs can be adapted to be useful under a COVID-19 mitigation scenario is in order.

The ERL Consumer User Story fits fairly well many such COVID-19 mitigation activities to be performed by robots. Here we suggest subtle modifications for each of the TBMs that would make them target explicitly such activities.

- **TBM1 - Getting to know my home**: the semantic information stored regarding furniture and objects inside the home should be augmented with information about their type of material (e.g., plastic, metal, wood), so as to guide the disinfection operations to be performed during TBM4.

- **TBM2 - Welcoming visitors**: the robot should take precautionary measures regarding the visitors allowed to enter the home, i.e., Dr. Kimble, the Deli Man and the Plumber, such as taking their temperature, asking a few questions (e.g., ”are you coughing or sneezing?”, ”have you been in contact with people potentially having COVID-19?”) and modifying its regular line of action accordingly, e.g., temperatures higher than 37.0° Celsius, and/or a positive answer to the above questions should prevent the visitor from entering the home. The robot could also handle situations of more than one simultaneous visitor, possibly disallowing more visitors in in case the maximum capacity defined is exceeded, and making sure the social distance rules are met inside the home, always checking distances and warning the involved visitors in case of their violation.
- **TBM3 - Catering for Granny Annie's comfort**: this task is a case of medicine and food service to a patient by a robot, without the need of another human presence, typical of hospital scenarios where medical personnel should avoid contact with infected patients. It could be augmented with the ability to provide remote conversations between Granny Annie and her family, friends or visitors. It should enforce social interaction rules, so as to compensate, implementing as natural human-robot interaction as possible, the absence of face-to-face interaction between the involved humans.
- **TBM4 - Visiting my home**: this task can be transformed to a disinfection task, where the robot, equipped with adequate disinfection devices (e.g., UV light emitting systems) goes around the home and switches on its disinfection device whenever the current room does not have any people. The robot must have way points that enable it performing a full home tour, avoid obstacles and detect people before disinfecting a room, much like in the current version of the TBM. The duration of the disinfection, as well as the incidence of the UV emission (e.g., using the robot arm), can be guided by the type of material expected in the current location, as mapped in TBM1.

II: ERL Consumer Rulebook

1 Introduction

This part of the document describes in detail the actual competition rules for ERL Consumer Benchmarks and it assumes that the reader has already read Part I of the document, with the ERL Consumer nutshell information. The audience for the current part are teams who want to participate in the competition, the organizers of ERL Consumer Tournaments, and the developers of simulation software, who want to provide their customers and users with ready-to-use models of the environment. They all need to know more details on the competition than the nutshell document provides.

2 How to Read and Outline of the Document

This remainder of this part is structured as follows: Section 3 describes in detail the ERL Consumer Task Benchmarks, their goals, specific procedures, scoring methods and details. Section 4 does the same for ERL Consumer Functionality Benchmarks.

This is the most relevant part of the rulebook, that an experienced ERL Consumer participant should read carefully.

Sections of former versions of this rulebook providing general information on awards, test beds, robots and teams, as well as the general procedures, scoring and benchmarking methods for Task Benchmarks, were now turned into appendices.

Appendix A surveys the number and kind of ***awards*** that will be awarded and when, as well as how the ranking of the award categories is determined based on individual benchmark results. The certification requirements for a ***test bed*** to be used in ERL Consumer Tournaments are described in some detail in Appendix B). Subsections of this appendix are devoted to the specification of the structure of the environment and its properties (B.1), to the objects in the environment relevant to the tasks on hand (B.2), to other objects not directly related to tasks but possible affecting the robot's behavior in other ways (e.g. need to avoid them in navigation, distractions and perceptual noise caused by them in robot vision) (B.3), to the networked devices embedded in the environment and accessible to the robot (B.5), and to the benchmarking equipment which may impose additional constraints to the robot's behavior (equipment presenting obstacles to avoid) or add further perceptual noise (visible equipment) (B.6). In Appendix C), we provide some specifications and constraints applying to the ***robots and teams*** permitted to participate in ERL Consumer, including the requirements for data logging. The ***general procedure*** to run Task Benchmarks, as well as the scoring methods used to rank the teams, are introduced in Appendix D. Appendix E depicts one instance of a ERL Consumer ***certified testbed***.

Any ERL Consumer participant should read these appendices at least once. Experienced ERL Consumer participants should use them primarily for consultation of details.

3 Task Benchmarks

This chapter describes in detail each of the Task and Functionality Benchmarks that will be available during the current ERL Consumer Season.

3.1 Task *Getting to know my home*

This task is focused on acquiring knowledge about the environment and on its explicit representation. The robot is required to identify some changes in the environment fully autonomously without any assistance from human users.

3.1.1 Task Description

Before each task run, some random changes in the environment are made with respect to the nominal configuration given to the teams during the set-up days. These changes involve:

- Closing a door connecting two rooms
- Moving two pieces of furniture from one room to another
- Moving two known objects from one furniture to another
- Placing an unknown object (trash) on the floor

The robot has to detect all the changes automatically and to provide an explicit representation of these changes referred to the map of the environment (that can be either acquired on-line during the task run, off-line before the run, or merge off-line and on-line acquisition).

3.1.2 Input Provided

The teams must create a semantic map of the apartment, during the set-up days, and label it with pieces of furniture and objects. The names of rooms, furniture and objects, along with their default locations, will be distributed to the teams before the competition. The objects and furniture will also be available to the teams during the setup days for training purposes.

The following information about the object locations is provided in advance to facilitate the robot's searching process:

- A list of four candidate locations (four pieces of furniture) where the changed objects could be located will be given in advance. These four furniture are not among the moved furniture and are always at their default locations.
- Three objects are placed on every candidate location. Two locations will contain a changed object that is to be found while the other two locations will contain the default objects for that location.
- An unknown manipulable object, such as an empty water bottle or an empty drink can, acting as the trash, will be placed on the floor of one of the rooms in a noticeable location (away from walls and furniture);

3.1.3 Expected Robot Behavior or Output

Phase 1: knowledge acquisition.

The robot is expected to detect the changes and represent them in an explicit format (see Expected Output below) fully autonomously.

At any time the teams can decide to move to Phase 2, even if not all changes have been detected. However, the task in Phase 2 can refer only to objects acquired during Phase 1.

Phase 2: knowledge use.

The robot has to show the use of the new acquired knowledge. This will be accomplished by moving one of the changed objects recognized in Phase 1 back to its default location. The default location of the trash is considered as the trash bin.

Expected output. The output provided by the teams is a set of files that must be saved in a USB stick given to the teams before the test. The USB stick will be formatted with FAT32 file system and all the files should be saved in a folder with the name of the team.

The following files are expected:

- semantic map file
- pictures of changed objects and furniture
- metric map files

1. **Semantic map file:** this must be a text file named `semantic_map.txt` containing a set of Prolog-like statements (or facts) in the following form:

```
predicate(arg_1, ..., arg_n).
```

The following predicates will be considered for evaluation:

(a) Predicates about doors and their status

- **Definition:**

```
type(door_ID, door).
connect(door_ID, room_name, room_name).
isOpen(door_ID, true|false ).
```

- **Example:**

```
type(door36,door)
connects(door36, kitchen, office).
isOpen(door36, true).
```

(b) Predicates about location of pieces of furniture

- **Definition:**

```
type(furniture_ID, furniture_name)
in(furniture_ID, room_name)
picture(furniture_ID, image_filename).
```

- **Example:**

```
type(ch1, kitchen_chair).
in(ch1, living_room).
picture(ch1, image_ch1.jpg).
```

Note: Only one piece of furniture for each type will be involved in the test and described in the semantic map file.

(c) Predicates about position, location and properties of objects)

- **Definition:**

```
type(object_ID, object_category).
in(object_ID, room_name).
on(object_ID, furniture_name).
picture(object_ID, image_filename).
```

- **Example:**

```
type(obj33,apple).
in(obj33,kitchen).
on(obj33,kitchen_table).
picture(obj33,image_obj33.jpg).
```


Note: Only the information about the door/furniture/objects that are involved in the change during the test will be evaluated. Additional information about other doors/furniture/objects can be included in the file as well, but they will not be used for determining the score.

2. **Pictures of objects:** actual images in standard image format (JPEG, PNG, BMP, PPM) named in the picture predicate of the semantic map. These images will be evaluated by a referee through visual inspection. The object named in the semantic map file must be visible in the foreground.

3. **Metric map files:** metric map (possibly acquired before the test) should be included, preferably in ROS format (i.e. bitmap (PNG/PPM) + YAML file), using the global reference system provided during the setup days. Note: this map will not be evaluated, but it is useful for benchmarking and statistics.

Complete Example: A complete example is illustrated below, where the following changes where executed:

- the door connecting the kitchen and the hallway is closed,
- a kitchen chair is moved to the living room,
- a plant is moved to the hallway,
- a can of coke is placed on the kitchen table in the kitchen,
- a box of biscuits is placed on the coffee table in the living room, and
- an empty water bottle is placed on the floor of the dining room.

The generated semantic map file which refers to the changes in the environment is given below:

```
type(door_1, door).
connects(door_1, kitchen, hallway).
isOpen(door_1, false).

type(kitchen_chair_1, chair).
in(kitchen_chair_1, living_room).
picture(kitchen_chair_1, kitchen_chair_1.jpg).

type(plant_1, plant).
in(plant_1, hallway).
picture(plant_1, plant_1.jpg).

type(object_1, coke).
in(object_1, kitchen).
on(object_1, kitchen_table).
picture(object_1, object_1.jpg).

type(object_2, biscuits).
in(object_2, living_room).
on(object_2, coffee_table).
picture(object_2, object_2.jpg).

type(object_3, trash).
```

```
in(object_3, dining_room).
on(object_3, floor).
picture(object_3, object_3.jpg).
```

3.1.4 Procedures and Rules

Before the robot enters the apartment, any of the doors in the environment will be closed and two randomly-selected pieces of furniture will be moved from their original position. In addition, several randomly-selected objects will be placed in the four candidate locations (above some piece of furniture) where only two of the objects are not in their default locations and need to be identified as the changed objects. Note that the closed door could be also the entrance door. In this case, this door will be closed immediately after the robot has entered the apartment.

Phase 1 The robot can move around in the environment for up to the maximum time limit of this task. The robot has to detect the changes specified above, and then represent them in an explicit format as described in sub-subsection 3.1.3.

Phase 2 (Assuming that the robot has either finished the task requested in the previous steps, or is incapable to do so) the robot can automatically, or via a speech command, proceed to move one of the objects recognized in Phase 1 back to its default location.

The knowledge acquired during Phase 1 must be reported in an explicit way by producing files in the given output format and by automatically saving them on a USB stick given to the team by the Technical Committee before the start of the run, as described in the Expected Output section. No manual intervention on the robot is allowed to save files on the USB stick.

3.1.5 Acquisition of Benchmarking Data

During this Task Benchmarks, beside the information related to the semantic map of the environment to be evaluated, the Internal Data described in Section C.3.2 will be collected.

3.1.6 Scoring and Ranking

The set A of eight possible *achievements* for this task are:

- The robot detects the door with changed state. (1)
- The robot detects each piece of the moved furniture. (2)
- The robot detects each of the changed objects. (2)
- The robot locates the trash. (1)
- The robot grasps an object in Phase 2. (1)
- The robot delivers the object to the default location. (1)

The set PB of *penalized behaviors* for this task are:

- The robot bumps into the furniture.
- The robot stops working.
- The robot drops the object when delivering.

Additional penalized behaviors may be identified and added to this list if deemed necessary. The set DB of *disqualifying behaviors* for this task are:

- The robot hits a human in the environment.
- The robot damages the test bed.

Additional disqualifying behaviors may be identified and added to this list if deemed necessary. These sets will be completed in later rule revisions.

3.2 Task *Welcoming visitors*

This task assesses the robot's capability to interact effectively with humans and to demonstrate different human-robot interaction behaviors when dealing with known and unknown people.

3.2.1 Task Description

The task involves handling several known and unknown visitors arriving at the home entrance in a random sequence, but separately from each other. The robot must recognize the known visitors and interact with the unknown visitors to identify them. The robot must then perform a set of visitor-specific behaviors such as manipulation, guiding and following. If a visitor has been admitted, the robot should guide him/her out after the visit is over.

3.2.2 Input Provided

The following information about the people acting as visitors is provided below.

- Dr. Kimble is a known person from the Organizing or Technical Committee running all the runs. Pictures of Dr. Kimble will be provided before the competition.
- The Postman is a known person from the Organizing or Technical Committee running all the runs. He/she will also wear a known uniform. Images of the uniform and the Postman will be given before the competition.
- The Deli Man will be unknown beforehand (the actual person will change in every run). He/she will be carrying a shopping bag with the breakfast inside.
- The Plumber will be unknown beforehand (the actual person will change in every run). He will carry a toolbox in his hand.

Some privileges are given to the four different kinds of people, as reported in the following table, and the robot has to act accordingly.

Privileges	Dr. Kimble	Postman	Deli Man	Plumber
Allow to enter the house	Yes	No	Yes	Yes
Allow to deliver objects	No	Yes	Yes	Yes
Allow access to bathroom	Yes	No	No	Yes
Allow access to kitchen	Yes	No	Yes	Yes
Allow access to bedroom	Yes	No	No	No
Allow access to living room	Yes	No	No	No

Pictures of the parcel used by the postman will be provided before the competition. The dimensions of the parcel are 9.5 inches / 24.13 cm (width) x 6.5 inches / 16.51 cm (length) x 1.75 inches / 4.45 cm (height) and weights about 30 grams.

3.2.3 Expected Robot Behavior or Output

For each visiting person, the following two phases are expected.

Phase 1: detection and recognition of the visitor.

Doorbell: Whenever a person rings the door bell, the robot can use its own on-board audio system to detect the bell ring(s) or the signal from the RSBB.
Opening the door: Naturally, the front door will be closed. The robot can either use its own manipulators to open the door or request the visitor to open it (e.g. using speech). An extra

achievement is assigned for the former case. Fig. 2.2 shows an example of the type of handle that will be mounted on the entrance door.
Identifying the visitor: Once the door is open, the robot must then use its on-board sensing to correctly identify the person first as known or unknown visitor. If the visitor is known, the robot should be able to recognize him/her. If the visitor is unknown the robot must interrogate the person to learn who is visiting and what is the purpose of the visit.
Greeting the visitor: For each detected visitor, the robot must greet the visitor with a spoken sentence, demonstrating that it has recognized the visitor. For example, "Hi postman, I am coming to get the post mail" or "Hello Plumber, where do you want to visit in the house?".

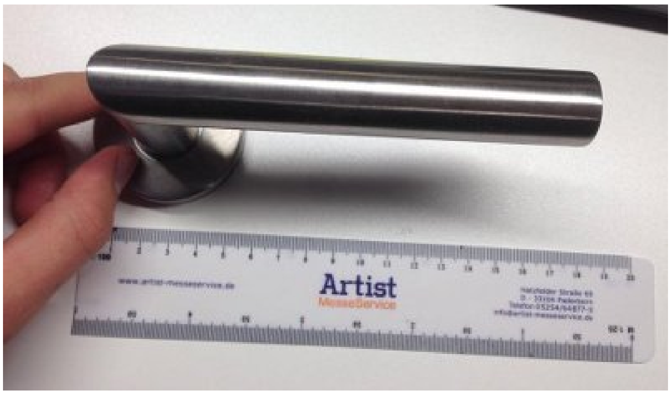


Figure 2.2: Door handle at the main entrance

Phase 2: executing the visitor-specific behavior.

The following behaviors are expected depending on the visitor:
Dr. Kimble: The robot *guides* the Doctor to Annie's bedroom; then it waits until the Doctor exits the bedroom, *follows* him/her to the entrance door, and allows the Doctor to exit. The human-following behavior must be observed.
PostMan: The robot receives the postal mail using its manipulator(s), allows the Postman to exit and delivers the mail to Granny Annie in the bedroom. A natural robot-human object handover is preferred, however, the robot could also advise the postman on how to hold or hand over the object.
Deli Man: The robot *guides* the Deli Man to the kitchen, asking him/her to deliver the breakfast box on the table; then it *guides* the Deli Man back to the entrance door, and allows him/her to exit. The human-guiding behavior must be observed and any violation of the privileges by the visitor must be identified and reported.
Plumber: The robot asks the Plumber where he/she would like to visit, *guides* the Plumber to the desired location, it waits for the plumber to finish his job; then it *follows* him/her back to the entrance door. The robot must always keep a close eye on the plumber by maintaining a close distance with him/her. The human guiding and following behaviors must be observed and any violation of the privileges by the visitor must be identified and reported.

After the execution of the visitor-specific behavior, the robot must return to the initial position and wait for the next visitor.

3.2.4 Procedures and Rules

The test starts with the robot being in one of the available rooms. The exact room will be announced at the beginning of the competition and will be same for all the teams. The visitors will be from all the four categories mentioned previously. The following steps will be repeated for each visitor (in a random order).

Step 1 The door-bell will be rung by the visitor. At this point in time, the robot must recognize the sound or use the RSBB, to see if someone rung the door bell.

Step 2 The robot must navigate to the entrance and open the entrance door either using its manipulator(s) or by asking the visitor to open the door.

Step 3 The robot must correctly identify the visiting person, greet him/her, and proceed to do the behavior-specific actions as instructed in Section 3.2.3.

Step 4 Once the visit is over, the robot will return to the initial position.

Note: To ensure that a closed-loop human-guiding and human-following is implemented on the robots, the visitors should slow down or stop during their way and the robot is expected to be able to maintain its distance with the visitor or be able to instruct the visitor to resume walking. Furthermore, the unknown visitors entering the house will try to break their privileges at least once during their visit that must be correctly identified and reported by the robot.

3.2.5 Acquisition of Benchmarking Data

During the execution of the benchmark, the Internal Data defined in Section C.3.2 will be collected together with the additional information described in the following table

Topic	Type	Frame Id	Notes
/ERLSCR/command[1]	std_msgs/String	–	–
/ERLSCR/visitor[2]	std_msgs/String	–	–
/ERLSCR/audio[3]	audio_common_msgs/AudioData	–	–
/ERLSCR/notification[4]	std_msgs/String	–	–

NOTE: the images and pointclouds in the Internal Data should contain the sensorial data used to recognize the visitor.

3.2.6 Scoring and Ranking

The set A of *achievements* for this task are:

- The robot correctly recognizes Dr. Kimble.
- The robot correctly recognizes the Postman.
- The robot correctly identifies the Deli Man.
- The robot correctly identifies the plumber.
- The robot exhibits the expected behavior for interacting with Dr. Kimble.
- The robot exhibits the expected behavior for interacting with the Postman.
- The robot exhibits the expected behavior for interacting with the Deli Man.

[1]The event or command causing the activation of the robot.
[2]The result of any attempt by the robot to detect and classify a visitor
[3]The audio signals of the conversation with the visitors.
[4]Any notifications from the robot (e.g., alarm if a visitor shows anomalous behavior)

- The robot exhibits the expected behavior for interacting with the Plumber.
- The robot opens the entrance door using its manipulator at least one time during the task.
- The robot is able to identify the violation of the privileges at least once from the two occurrences.

The set PB of *penalized behaviors* for this task are:

- The robot stops working.
- The robot fails in maintaining the original state of the environment.
- The robot requires extra repetitions of speech.
- The robot bumps into the furniture.

Additional penalized behaviors may be identified and added to this list if deemed necessary. The set DB of *disqualifying behaviors* for this task are:

- The robot hits Annie or one of the visitors.
- The robot damages the test bed.

Additional disqualifying behaviors may be identified and added to this list if deemed necessary. These sets will be completed in later rule revisions.

3.3 Task *Catering for Granny Annie's Comfort*

This benchmark aims at assessing the robot's performance for executing general purpose requests of users in the apartment. It focuses on the integration of different robot abilities such as human-robot interaction, navigation and robot-object interaction. The task does not have a predefined story or a predefined set of actions requiring more than the traditional state machine-like behaviors. Note: This task is an updated and merged version of the former ERL-SR *Catering for Granny Annie's Comfort* and *General Purpose Service Robots* tasks.

3.3.1 Task Description

The robot is going to help a user with his/her daily tasks. After waking up in the morning, the user calls the attention of the service robot by touching a button on a tablet computer. When the robot arrives, the user gives a single speech command composing of three actions from the following three categories:

1. Searching for an item or a person
2. Manipulating and delivering an object
3. Accompanying a person inside or outside the house

These actions are randomly generated into a single sentence in no predefined order. The robot is required to understand the actions and execute them accordingly.

3.3.2 Input Provided

Objects and names The list of objects and people's names used in the task will be provided before the competition by the local organizers. For every object/person to be found by the robot a set of possible locations with an associated likelihood will be provided:

Object/Person	Location	Likelihood %
Object i	Furniture 1	60
	Furniture 2	30
	Furniture 3	10
Person i	Room 1	60
	Room 2	30
	Room 3	10

All of the objects will be made available to the teams for calibration during the setup days.
Note: In the case of finding a person, only one person will be present at one of the target locations.

Commands: A command will be generated randomly before the task run and is composed of three actions as described in Section 3.3.1. The lexicon used in the commands will consist of the names of people/objects/locations (provided to the teams before the competition), along with the following verbs divided based on the action categories:

Searching
find, locate, look for, search for, pinpoint, spot

Accompanying
guide, take, lead, accompany, follow, escort

Manipulating
put, place, leave, set, get, grasp, take, retrieve, pick up, bring, deliver, give, hand

The following are some examples selected from a large set of possible commands:

- Locate Tracy, lead her to the bedroom, and bring me an apple from the kitchen cabinet.
- Find the bottle, place it in the trash bin, and take John from the bedroom to the exit.
- Guide me to the bedroom, find my glasses, and bring me a pear from the kitchen table.
- Give me the cup on the coffee table, find John, and follow him.

Calling the Robot for Attention: The initial location of the user will be announced in advance. The user has a tablet computer which runs an application provided by the consortium. It allows the user to call the robot's attention. The communication with the tablet is realized through the RSBB. Further details are specified in the ERL Wiki [1]. Alternatively, if the tablet is unavailable in a testbed, the start signal of the task, sent from the RSBB, will be considered as the calling signal.

3.3.3 Expected Robot Behavior or Output

After the robot is called, the robot is expected to reach the room where the user is located, approaching in such a way that spoken communication is possible. The robot should then state its readiness to receive the command. When the command is given, the robot should show that it has understood all the actions in an appropriate way (e.g., by repeating it back to the user and asking if it was correctly understood).

The robot is then expected to successfully execute all of the actions in the order that was given in the command. If the robot is unable to perform any of the actions it should announce it and proceed with the other actions.
Notes:

- When executing the action "Searching for an item or a person", an object/ person is considered as found if the robot gets to the location of that object/person and while facing it, it announces that it has found the object. The robot can announce the location of an object/person without actually visiting its location if and only if it has already visited the other two possible locations where it could not detect the object.
- For the action of "Accompanying a person", the robot must initially instruct the user on how to communicate the stop following command. For example, the robot can ask the user to speak a command or wave in order to issue the stop signal. The user will behave as instructed by the robot.

3.3.4 Procedures and Rules

A successful execution of this task would be the robot going to the user (played by a member of the OC/TC) and executing all or some of the actions in the command. The task involves the following activities:

1. User initiates the task by calling the robot's attention.
2. The robot approaches the user in such a way that spoken communication is possible, stating its readiness to receive commands.
3. The user provides the sentence from the predefined list of possible commands.
4. The robot confirms that it understood the command and executes the corresponding actions in the command.
5. After completing the task, the robot returns to the user.

Teams are allowed to give instructions as to how the user should speak to the robot. If the robot fails to understand the original command after three times a team member can replace the user to repeat the same command.

Alternatively, if the robot is incapable of processing and understanding the original command in full, a team member can decompose or interpret the command into a desired format that is understandable by the robot. However, the achievements for "correctly understanding the original command" and "executing the full command without any interruption" will not be obtained.

3.3.5 Acquisition of Benchmarking Data

During the execution of the benchmark, the Internal Data defined in SectionC.3.2 will be collected together with the additional information described in the following table

Topic	Type	Frame Id	Notes
/erlc/command[5]	std_msgs/String	–	–
/erlc/audio[6]	audio_common_msgs/AudioData	–	–

NOTE: the images and pointclouds in the Internal Data should contain the object to be operated.

3.3.6 Scoring and Ranking

The set A of *achievements* for this task are:

- The robot enters the room where the user is waiting.
- The robot understands the user's original command without any interpretations.
- The robot understands at least one of the actions to be executed.
- The robot manages to find the right object/person.
- The robot correctly executes the accompanying action.
- The robot grasps and picks up the object to be delivered.
- The robot delivers the object to the desired location
- The robot completes all three actions continuously without any interruptions.

The set PB of *penalized behaviors* for this task are:

- The robot requires multiple repetition of the command
- The robot bumps into the furniture.
- The robot drops an object.
- The robot stops working.

[5] The command produced by the natural language analysis process.

[6] The audio of the conversation between user and the robot. Speech files from all teams and all benchmarks (both Task benchmarks and Functionality benchmarks) will be collected and used to build a public dataset. The audio files in the dataset will therefore include all the defects of real-world audio capture using robot hardware (e.g., electrical and mechanical noise, limited bandwidth, harmonic distortion). Such files will be usable to test speech recognition software, or (possibly) to act as input during the execution of speech recognition benchmarks.

- The robot fails to return to user after task completion.

Additional penalized behaviors may be identified and added to this list if deemed necessary. The set DB of *disqualifying behaviors* for this task are:

- The robot hits the user or another person in the environment.
- The robot damages or destroys the objects requested to manipulate.
- The robot damages the test bed.

Additional disqualifying behaviors may be identified and added to this list if deemed necessary. These sets will be completed in later rule revisions.

3.4 Task *Visit my home*

This task focuses on safe navigation in dynamic environments, obstacle avoidance, obstacle interaction, people perception and tracking and recognizing a previously unknown person.

3.4.1 Task Description

In this task, the robot should visit three predefined rooms in the apartment and to identify the number of persons in each of the rooms. The robot needs to avoid obstacles on its path and to interact with them based on the nature of the obstacle. Furthermore, the robot must interact and follow a person to an unknown location outside the arena and to find its way back to the arena.

3.4.2 Input Provided

Three rooms from the apartment will be selected and announced to the teams during the setup days. Each room will contain between zero to five persons that can be either sitting or standing anywhere inside the room. The path to one of the rooms will be blocked by an obstacle of unknown nature. This obstacle is selected randomly and can be one of the following:

- Small object: A small Kraft paper shopping bag
- Wheeled object: A rolling chair, cart, wheeled luggage or some other object with wheels.
- Human obstacle: A person who is unknown beforehand.

(Objects will be available to the teams during the setup days)
The entrance door to the apartment will be initially closed while all other doors are left open.

3.4.3 Expected Robot Behavior or Output

Phase 1: Entering the arena

The task starts with the robot waiting outside the arena and when the entrance door is opened by the referee. The robot should detect the door-opening event and enter the arena autonomously. Alternatively, the team may start the robot using a start button without getting the respective achievement.

Phase 2: Navigation, People perception and Obstacle interaction

After entering the arena, the robot must navigate to all three defined rooms to identify and report the number of persons inside each room.

The path to one of the rooms will be blocked by one of the obstacles described in Section 3.4.2. The robot is expected to identify, state the nature of the obstacle and interact with the obstacle accordingly. Possible actions include:

Figure 2.3: The type of shopping bag used as the small obstacle to be removed using the robot's manipulators. (The bag dimensions will be between 10 and 30 cm per edge with weight of less than 300 grams)

- Small object: Remove the small object from the path using the robot's manipulators, or overcome the obstacle in case of bipedal robots.
- Wheeled object: Gently push away the object to unblock the path using either the manipulators or the robot's body.
- Human obstacle: Kindly ask the human to move away.

If the robot is unable to remove the obstacle it can ask the referee to remove it to continue with the task. Otherwise, the obstacle will clear or be cleared after roughly 45 seconds.

Phase 3: Human following

After completing phase two, the robot must ask one of the identified persons to guide the robot outside the apartment. The robot must memorize this person and follow him/her outside the arena to a previously unknown waypoint. This phase can include the following states:

- Training: The robot has to memorize the operator. The robot may instruct the operator to follow a certain setup procedure and instruct the operator on what to do when the robot needs to stop following.
- Following: When the robot signals that it is ready to start following, the operator starts walking in a natural way through a designated path outside the arena. The robot needs to follow the operator until the operator asks the robot to stop.
- Resuming: If the robot loses the operator, it can ask him/her to come back or give a signal by waving to resume following, but will be penalized for doing so.
- Stop following: Upon reaching the target, the operator will command the robot to stop following him as instructed in the training phase.

Phase 5: Returning Home

After receiving the stop command from the operator the robot must then go back and enter the apartment through the entrance door. The robot has to open the door of the apartment, that was closed by the referee after the robot left the arena, in order to enter the apartment. The robot is allowed to use its base to gently push the door open, but only after it has used its manipulator to initially open the door by the handle

3.4.4 Procedures and Rules

The referee starts the test by opening the entrance door. The robot must navigate to the rooms of the apartment and follow the expected behaviors described in Section 3.4.3. If a robot is unable to reach a room it must announce it and proceed to the next waypoint.

3.4.5 Acquisition of Benchmarking Data

During this Task Benchmark, the Internal Data described in Section C.3.2 will be collected.

3.4.6 Scoring and Ranking

The set A of *achievements* for this task are:

- The robot correctly identifies the number of persons in room 1.
- The robot correctly identifies the number of persons in room 2.
- The robot correctly identifies the number of persons in room 3.
- The robot recognizes properly the nature of the obstacle.
- The robot interacts with obstacle according to their nature.
- The robot follows the operator and reach the target location.
- The robot is able to find its way back to the entrance.
- The robot correctly opens the door and passes through the open door.

The set PB of *penalized behaviors* for this task are:

- The robot does not enter the arena autonomously when the door opens.
- The robot bumps into a non-human obstacle.
- If the robot loses the operator, it requires that explicit actions are performed by the operator (e.g., waving) to retrieve them.
- The robot stops working.
- Any behavior towards attendees and/or operator that is deemed as threatening by the referees (e.g., rapidly moving towards a person, then stopping at moderate distance)

Additional penalized behaviors may be identified and added to this list if deemed necessary. The set DB of *disqualifying behaviors* for this task are:

- The robot hits the operator or any other human.
- The robot damages an object.

Additional disqualifying behaviors may be identified and added to this list if deemed necessary. These sets will be completed in later rule revisions.

4 Functionality Benchmarks

4.1 Object Perception Functionality

This functionality benchmark has the objective of assessing the capabilities of a robot in processing sensor data in order to extract information about observed objects.

4.1.1 Functionality Description

Teams are provided with a list of individual objects (instances), subdivided into classes. The benchmark requires that the robot, when presented with objects from such list, detects their presence and estimates their class, instance, and location. For example, when presented with a bottle of milk, the robot should detect a bottle (class) of milk (instance) and estimate its pose w.r.t. a known reference frame.

4.1.2 Input Provided

The set of individual objects, denoted as "object instances", that will actually be presented to the robot during the execution of the functionality benchmark will be provided to the teams before the competition. All objects are commonplace items that can be found in domestic or retail environments. Object instances are subdivided into classes of objects that have one or more properties in common, here denoted as "object classes". Objects of the same class share one or more properties, not necessarily related to their geometry (for instance, a class may include objects that share their application domain). Each object instance and each object class is assigned a unique ID.

More precisely, teams will be provided with the following information:

- Descriptions of all the object instances;
- Subdivision of the object instances into object classes (for instance: boxes, mugs, cutlery);
- Reference systems associated to the table surface and to each object instance (to be used to express object poses).

4.1.3 Expected Robot Behavior or Output

The objects that the robot is required to perceive are positioned, one at the time, on a table (benchmark setup area) located directly in front of the robot. The actual pose of the objects presented to the robot is unknown before they are set on the table. For each presented object, the robot must perform all of the following:

- Object detection (i.e., class recognition): perception of the presence of an object on the table and association between the perceived object and one of the object classes (see "Information provided to the team").
- Object recognition (i.e., instance recognition): association between the perceived object and one of the object instances belonging to the selected class (see sub-subsection 4.1.2).
- Object localization (i.e., pose estimation): estimation of the 3D pose of the perceived object with respect to the benchmark setup reference frame.

4.1.4 Procedures and Rules

Every run in all the stages of this functionality benchmark will be preceded by a safety-check similar to that described for the task benchmark procedures.

Each team is required to perform this functionality benchmark according to the steps mentioned below. The maximum time allowed for one functionality run is 10 minutes.

Step 1 An object of unknown class and unknown instance will be placed on a table in front of the robot.

Step 2 The robot must determine the object's class, its instance within that class as well as the 2D pose of the object w.r.t. the reference system specified on the table (see Fig. 2.4).

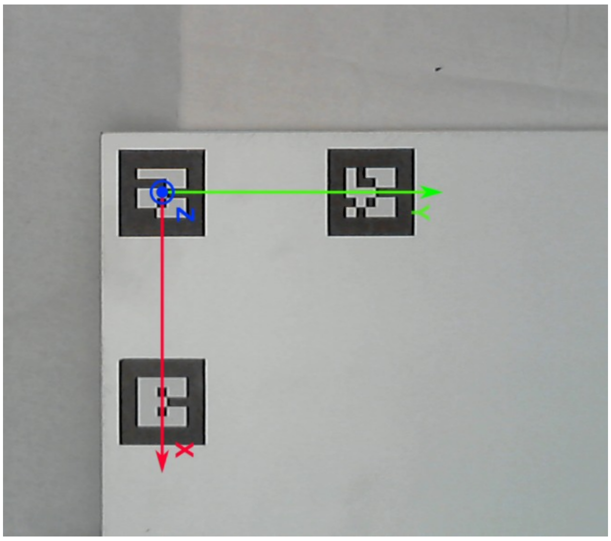


Figure 2.4: Reference frame on the FBM table.

The preceding steps are repeated until the time runs out or the maximum number of objects have been processed.

4.1.5 Acquisition of Benchmarking Data

During the execution of the benchmark, the Internal Data defined in Section C.3.2 will be collected together with the additional information described in the following table

Topic	Type	Frame Id	Notes
/erlc/notification[7]	std_msgs/String	–	–

NOTE: the images and pointclouds in the Internal Data should contain the sensorial data used to recognize each of the presented object, thus we expect to have 10 images (if recognition uses a camera), 10 pointclouds (if recognition uses a depth sensor), and 10 notification strings.

[7]The string with the notification of the perceived object should be in a tab separated string: CLASS OBJECT_ID X Y THETA

4.1.6 Scoring and Ranking

Evaluation of the performance of a robot according to this functionality benchmark is based on:

1. The number and percentage of correctly recognized objects. (i.e.,instance recognition)

2. Pose error for all correctly identified objects.

3. The number and percentage of correctly classified objects. (i.e.,class recognition)

4. Execution time (if less than the maximum allowed for the benchmark).

The previous criteria are in order of importance: the first criterion is applied first and teams will be scored according to the common accuracy metrics; the ties are broken by second criterion which is the position error. Since the position error is affected by the precision of the ground truth system we will use a set of accuracy classes in this as well and in case of ties we will resort to the third criterion and finally the execution time.

4.2 Navigation Functionality

4.2.1 Functionality Description

This functionality benchmark aims at assessing the capabilities of a robot to correctly and autonomously navigate in a typical apartment, containing furniture and objects spread through the apartment's rooms. The benchmark will use the ERL Consumer test bed (see Sec. B). From a predefined starting position, the robot will receive a list of waypoints that it must visit to reach a goal position.

4.2.2 Input Provided

The following information will be provided in advance:

- The starting position.

- The number of waypoints that the robot must visit.

- The maximum time allowed to the robot to go from each waypoint to the next waypoint, without penalization.

Furthermore, the coordinates of the waypoints that the robot must visit will be communicated sequentially to the robot during the run time. A waypoint encapsulates a X, Y and θ. The coordinates are given in the test bed reference frame (which will be defined on site and provided during the first day to the teams).

4.2.3 Expected Robot Behavior or Output

Teams are required to set their robot on a specific starting position. Then, the robot receives, through the RSBB, the start signal, as well as the first waypoints that it must reach. The robot must navigate to this waypoint and send back a signal to the RSBB, indicating that it has reached the waypoint. At this time the RSBB will issue a new waypoint. This process is repeated until the robot has reached the final waypoint and a stop signal is issued by the RSBB.

4.2.4 Procedures and Rules

All teams are required to perform this functionality benchmark according to the steps mentioned below. The task must be performed exclusively in autonomous mode. No teleoperation is allowed. Teams will have up to ten minutes to complete the functionality benchmark.

Step 1 The team is required to start their robot on a pre-defined starting position. This starting position will be given to the teams before each run.

Step 2 When the procedure starts, the robot receives a sequence of waypoints, one by one, that it must follow in the respective order, sending back a signal through the RSBB, each time it reaches a waypoint. The robot must avoid hitting any obstacle it encounters in its path.

Step 3 The procedure stops when the robot notifies it has reached the last waypoint, when the time given to complete the test expires, or when the robot hard-hits an obstacle.

It is considered a hard hit when:

- A collision damages any of the obstacles;
- The robot pushes or continually touches an obstacle for more than 3 seconds;
- The robot forces its path through an obstacle;

The obstacles can be of three types:

- **Static and previously mapped**: Hardware already present in the house such as furniture, doors, walls, defined in B.2.1. The teams should already have this obstacles mapped from set-up days. These items will not change during this functionality benchmark.
- **Static**: Items Granny Annie left lying on the ground. The obstacles may be of different shapes and sizes, are not previously known by the teams, and may be different in between runs.
- **Dynamic**: Granny Annie's visitors. People moving inside the house.

4.2.5 Acquisition of Benchmarking Data

During the execution of the benchmark, the Internal Data defined in SectionC.3.2 will be collected together with the additional information described in the following table

Topic	Type	Frame Id	Notes
/erlc/robot_pose_waypoint[8]	geometry_msgs/PoseStamped	/map	when reached
/erlc/marker_pose_waypoint[9]	geometry_msgsPoseStamped	/map	when reached

This functionality benchmark will be fully automated (no human operation will be allowed) and, for that, the robot has to communicate with the RSBB. It will receive the list of target waypoints from the RSBB and it must send back a signal, each time it reaches a waypoint.

4.2.6 Scoring and Ranking

At each run and for each team, three metrics will be used to score the performance: Accuracy, number of obstacle hits, execution time.

[8]The 2D robot pose, once each waypoint is reached, at the floor level, i.e., $z = 0$ and only yaw rotation.
[9]The 3D pose of the marker in 6 degrees of freedom once each waypoint is reached.

Topic	Type	Notes
/roah_rsbb/goal	geometry_msgs/Pose2D	List of waypoints, sent by the RSBB to the robot, when starting the task.
/roah_rsbb/reached_waypoint	roah_rsbb_comm_ros/UInt8	Message sent by the robot to the RSBB, when reaching a point. It must include the number of the respective waypoint in the sequence (starting from zero).

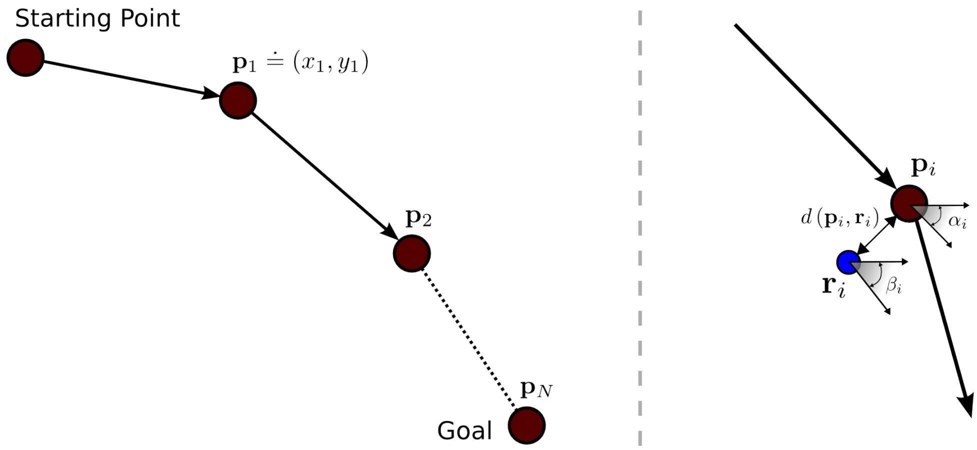


Figure 2.5: Depiction of the sequence of waypoints that the robot must follow, during the navigation functionality benchmark.

- Accuracy scoring will be based on the the mathematical means of the distance from the waypoints and and the corresponding orientation error. The distance mean from the robot to the target waypoint follows the equation

$$A = \frac{1}{N} \sum_{i}^{N} d(\mathbf{p}_i, \mathbf{r}_i), \tag{2.1}$$

where $d(\mathbf{p}_i, \mathbf{r}_i)$ is the Euclidean distance from the robot's position $\mathbf{r}_i$ to the target waypoint $\mathbf{p}_i$ (as shown in Figure 2.5) and N the total number of points.

Considering the orientation, the mean is

$$B = \frac{1}{N} \sum_{i}^{N} \delta(\alpha_i, \beta_i), \tag{2.2}$$

where $\delta(\alpha, \beta)$ is the absolute value of the difference between the desired waypoint orientation α and the robot's orientation β, such that $\delta(\alpha, \beta) = |\alpha - \beta|$

After the computation of these accuracy scorings, they will be discretized and fitted in one of the following groups:

  - 1: $A < 10cm$ AND $B < 20°$;
  - 2: $A < 30cm$ AND $B < 45°$;
  - 3: $A < 50cm$ AND $B < 90°$;
  - 4: $A < 80cm$ AND $B > 90°$;

A lower group number corresponds to the better performance. Therefore, teams will be ranked starting from group 1. Note that for a team to be placed in any of the groups, it **must** respect the limits for A and B. If a team has a score that does not fit any of groups defined above (e.g. mean of the error above 80cm), it will not receive scoring in the respective Functionality Benchmark run;

- If more than one team is found inside each of the previously defined group, the number of obstacle hits will be used as a tie breaker, where the team with less hits will be ranked first and so on. Note that hits will only be considered as a tie breaker, *i.e.*, a team in group 2 will never be ranked first than any team in 1, despite of the number of hits (note that a class of hard hits was defined which, if they happen, the procedure stops);
- If teams are still tied, time will be the decisive tie breaker.

Note that throughout the competition and according to the teams performance, the thresholds for the classes of A and B can be changed.

4.3 Speech Understanding Functionality

4.3.1 Functionality Description

This functionality benchmark aims at evaluating the ability of a robot to understand speech commands from users. A set of spoken sentences, recorded in different consumer environments, will be broadcast through a speaker. All robots should be able to capture audio from an on-board microphone, to record the captured audio in a file and to interpret the corresponding commands. The system should produce an output according to a final representation defined in the following sections. Such a representation will have to respect a command/arguments structure, where each argument is instantiated according to the arguments of the command evoking verb. It is referred to as *Command Frame Representation* (CFR) (e.g., "go to the living room" will correspond to `MOTION(goal:`*"living room"*`)`).

4.3.2 Input Provided

Some information about the lexicon (verbs and nouns of objects) used in the benchmark and a sample audio file can be found in the ERL Wiki [1].

In order to evaluate the correct understanding of a command expressed in natural language (e.g. through a sentence), a semantic representation formalism based on *semantic frame* has been selected. Each frame corresponds to an action, namely a robot command. A set of arguments is associated to each frame, specifying part of the command playing a particular role with respect to the action expressed by the frame. For example, in the command "*go to the dining room*" the *Motion* frame is expressed by the verb *go*, while the part of the sentence "*to the dining room*" corresponds to the GOAL argument, indicating the destination of the *Motion* action. The set of frames defined and selected for this Benchmark is reported in the following list, together with the set of associated arguments (current arguments are reported in bold italic inside the example commands):

- *Motion*: the action performed by the robot itself of moving from one position to another, occasionally specifying a specific path followed during the motion. The starting point is always taken as the current position of the robot.
  - GOAL: the final position in the space to be occupied at the end of the motion action, e.g. "*Go* ***to the kitchen***"; "*Go* ***to the right of the sofa***".
  - PATH: the trajectory followed while performing the motion towards the GOAL, e.g. "*Move* ***along the wall***".

- *Searching*: the action of inspecting an environment or a general location, with the aim of finding a specific entity.
  - THEME: the entity (most of the time an object) to be searched during the searching action, e.g. "*Search* ***for the glass***".
  - GROUND: the environment or the general location in the space where to search for the THEME, e.g. "*Find the glass* ***in the living room***".
- *Taking*: the action of removing an entity from one place, so that the entity is in robot possession.
  - THEME: the entity (typically an object) taken through the action, e.g. "*Take* ***the cereal box***".
  - SOURCE: the location occupied by the THEME before the action is performed and from which the THEME is removed, e.g. "*Grab the mayo* ***on the table***"; "*Remove the sheets* ***from the bed***".
- *Placing*: the action of placing an entity that the robot already posses in a place or position in the space.
  - THEME: the entity (typically an object) placed through the action, e.g. "*Drop* ***the jar***".
  - GOAL: the location that should be occupied by the THEME after the action is performed, e.g. "*Put the can* ***on the counter***".
- *Bringing*: the action of changing the position of an entity in the space from a location to another.
  - THEME: the entity (typically an object), being carried during the bringing action, e.g. "*Bring* ***the garbage*** *to the kitchen*".
  - GOAL: the endpoint of the path along which the carrier (e.g. the robot - and thus the THEME) travels, e.g. "*Bring the garbage* ***to the kitchen***".
  - SOURCE: the beginning of the path along which the carrier (e.g. the robot - and thus the THEME) travels, e.g. "*Bring the garbage* ***from the dining room*** *to the kitchen*".
  - BENEFICIARY: the person to whom the THEME must be brought, e.g. "*Bring* ***me*** *the mobile phone*".

Composition of actions is also possible in the CFR, corresponding to more complex action as the *Pick_and_place* action, represented by a sequence of *Taking* frame followed by a *Bringing* frame (e.g. for the command "*take the box and bring it to the kitchen*").

The grammar specifying the correct syntax for a CFR will be also provided, and is reported in the following. In order to represent the semantic frames extracted from the commands in a compact way, a specific syntax has been defined, called *Command Frame Representation* (CFR). Such a representation will have to respect a command/arguments structure, resembling the common syntax for a programming language method: the frame represents the method name, while the arguments represent the method arguments. For example, for the "*go to the dining room*" command, where a *Motion* frame is expressed and the GOAL argument is instantiated with "*to the dining room*", the corresponding CFR will be:

```
MOTION(goal:"to the dining room").
```

It is worth underlying that more than one argument can be expressed in a command.

Results of the Speech Understanding Functionality must be presented according to the CFR formalism. The grammar specifying the correct syntax for a CFR will be also provided, and is reported in the following.

```
Command → Single_command | Composed_command

Composed_command → Single_command#Command

Single_command → Action(Arguments)

Action → MOTION | TAKING | BRINGING | SEARCHING | PLACING

Arguments → Argument | Argument, Arguments

Argument → Argument_name:“Role_filler”

Argument_name → theme | goal | source | path | ground | beneficiary

Role_filler → Defined_lexicon
```

where `Defined_lexicon` is the lexicon that will be released to the team before the competition, including names of rooms (e.g. *hallway*, *living room*, etc.) and objects (e.g. *cereal box*, *jar*, etc.).

Composition of actions is also possible in the CFR, corresponding to more complex action as the *Pick_ and_ place* action, represented by a sequence of *Taking* frame followed by a *Placing* frame (e.g. for the command "*take the box and put it on the table*"). The corresponding CFR will be:

```
TAKING(theme:“the box”)#PLACING(theme:“it”,goal:“on the table”)
```

A parser is available in the ERL Wiki [1] in order to check the format compliance of the produced output in terms of interpretation.

Here we provide some annotation examples. Considering the command "*go to the kitchen and take the coffee mug*", where two frames are evoked (e.g. *Motion*, the two associated annotation will be:

$$[go]_{Motion}\ [to\ the\ kitchen]_{\text{GOAL}}\ and\ take\ the\ tea\ cup$$

$$go\ to\ the\ kitchen\ and\ [take]_{Taking}\ [the\ tea\ cup]_{\text{THEME}}.$$

The frame are associated to the verbs as they evoke the action expressed by the frames. This double tagging of the semantic information will correspond to the conjunction of the CFR of the two frames, that is:

```
MOTION(goal:“to the kitchen”)#TAKING(theme:“the tea cup”).
```

Frames can have more than one single arguments, as for the command "*take a towel from the bathroom*", corresponding to the following tagging and CFR:

$$[take]_{Taking}\ [a\ towel]_{\text{THEME}}\ [from\ the\ bathroom]_{\text{SOURCE}}$$

```
TAKING(theme:“a towel”,source:“from the bathroom”).
```

In a sentence, it will be possible to find some arguments that have not been tagged. These will be only arguments not defined for this task (and thus not reported in the list in Section 4.3.2) as, for example, the MANNER argument, representing the manner in which the action take place. For example, in the command "*Search carefully the bedroom for my wristwatch*", the adverb *carefully*, representing the MANNER, is not tagged:

$$[search]_{Searching}\ carefully\ [the\ bedroom]_{\text{GROUND}}\ [for\ my\ wristwatch]_{\text{THEME}}$$

`SEARCHING(ground:"the bedroom",theme:"for my wristwatch").`

Finally, some commands have been enriched with colloquial forms, as the use of modal verbs, e.g. "*could you please find my jacket?*". These particles are not considered in the tagging, as they represent only inflections that, in this case, don't affect the general meaning of the command:

$$could\ you\ please\ [find]_{Searching}\ [my\ jacket]_{\text{THEME}}\ ?$$

`SEARCHING(theme:"my jacket").`

More detailed information about the available audio datasets, the parser and the lexicon can be found in the ERL Wiki [1].

4.3.3 Expected Robot Behavior or Output

The robot should be able to understand a command starting from the speech input. The robot should correctly transcribe the command and recognize the action to perform, resulting in the correct command frame (e.g. `MOTION` for a *motion* command) and the arguments involved (e.g. the `goal` of a *motion* command). The output of the robot should provide the CFR format for each command, as reported in Section 4.3.1 and defined in Section 4.1.2.

For each command, the system should generate the corresponding transcription and the interpretation in the CFR format. This information have to be saved in an output text file called `results.txt`. In this file, a line has to be added for each command, following the format

$$\texttt{command\_number}_i|\texttt{command\_transcription}_i|\texttt{CFR}_i$$

where `Command_number`, `command_transcription` and `CFR` represent respectively the number, the transcription and the interpretation of the i-th audio file, separated by a pipe (|). The `results.txt` file must be encoded using standard UTF-8 character encoding: files with a different encoding will be automatically rejected.

NOTE: the correspondence between the data in the results file and the audio command numbers must be consistent.

If the Automatic Speech Recognition fails in transcribing a sentence, the label `BAD_RECOGNITION` must be used. Similarly, if the semantic frame extraction does not produce any correct result, the `NO_INTERPRETATION` label must be reported. An example of `results.txt` file is reported in the following:

```
command_1|move to the living room|MOTION(goal:"living room")
command_2|BAD_RECOGNITION|NO_INTERPRETATION
...
```

NOTE: every line not respecting the format required and described above will be skipped during the evaluation phase.

In addition to saving the output on the text file, which will be used for scoring, the organizing committee might ask the teams to also send the output to the RSBB, after every command, in order to visualize the results in real-time for the audience.

4.3.4 Procedures and Rules

The robots will be disposed in circle around a 360° speaker that will broadcast a set of spoken sentences. These sentences are pre-recorded commands in different consumer environments and are played continuously while containing a small pause between the commands. For every command the command number is also announced in advance. An example can be downloaded from: https://sites.google.com/site/erlsrlisbon/resources/test.mp3
NOTE: An autonomous detection of the command number is highly preferred, however, teams can also indicate the start of a new command by pressing a single button (either a button in a GUI or a key on the keyboard).

The benchmarking procedure is performed for all the teams in parallel. Each team receives a USB stick before the benchmark begins to connect to their robots. Once the benchmark is over, only one button can be pressed to store the output file and all teams must then deliver their USB sticks.

4.3.5 Acquisition of Benchmarking Data

During the execution of the benchmark, the following data will be collected:

- Sensor data (in the form of audio files) used by the robot to perform speech recognition[10];
- The set of all possible transcription for each user utterance;
- The final command produced during the natural language analysis process;
- Intermediate information produced or used by the natural language understanding system during the analysis as, for example, syntactic information.

4.3.6 Scoring and Ranking

During the functionality benchmark, different aspects of the speech understanding process will be assessed:

1. The Word Error Rate on the transcription of the user utterances, in order to evaluate the performance of the speech recognition process.
2. For the generated CFR, the performance of the system will be evaluated against the provided *gold standard* version of the CFR, that is conveniently paired with the analyzed audio file and transcription. Two different performances will be evaluated at this step. One measuring the ability of the system in recognizing the main action, called *Action Classification* (AcC), and one related to the classification of the action arguments, called *Argument Classification* (AgC). In both cases the evaluations will be carried out in term of Precision, Recall and F-Measure. This process is inspired to the *Semantic Role Labeling* evaluation scheme proposed in [2]. For the AcC this measures will be defined as follow:
   - Precision: the percentage of correctly tagged frames among all the frames tagged by the system;
   - Recall: the percentage of correctly tagged frames with respect to all the *gold standard* frames;

[10] Speech files from all teams and all benchmarks (both Task benchmarks and Functionality benchmarks) will be collected and used to build a public dataset. The audio files in the dataset will therefore include all the defects of real-world audio capture using robot hardware (e.g., electrical and mechanical noise, limited bandwidth, harmonic distortion). Such files will be usable to test speech recognition software, or (possibly) to act as input during the execution of speech recognition benchmarks.

- F-Measure: the harmonic mean between Precision and Recall.

Similarly, for the *AgC*, Precision, Recall and F-Measure will be evaluated, given an action f, as:

- Precision: the percentage of correctly tagged arguments of f with respect to all the arguments tagged by the system for f.
- Recall: the percentage of correctly tagged arguments of f with respect to all the *gold standard* arguments for f.
- F-Measure: the harmonic mean between Precision and Recall.

3. Time utilized (if less than the maximum allowed for the benchmark).

The final score will be evaluated considering both the *AcC* and the *AgC*. Only the F-Measure will be considered for both measures, each one contributing for 50% of the score. The *AgC* F-Measure will be evaluated for each argument, and the final F-Measure for the *AgC* will be the sum of the single F-Measure of the single arguments divided by the number of arguments. This final score has to be considered as an equivalence class. If this score will be the same for two or more teams, the *WER* will be used as penalty to evaluate the final ranking. This means that a team belonging to an equivalence class can not be ranked lower than one belonging to a lower one, even though the final score, considering the *WER* of the first is lower than the score of the second.

4.4 People Perception Functionality

This functionality benchmark aims at assessing the capabilities of a robot to correctly detect, locate and recognize humans. Human perception is one of the most important capabilities required for human-robot interactions and essential for a successful deployment of robots in many consumer environments.

4.4.1 Functionality Description

The benchmark requires that the robots correctly detects the presence of a human inside a predefined target area, recognizes the person and accurately estimates his/her position.

4.4.2 Input Provided

A bounded rectangular target area, of approximately 3m×3m, will be defined on the floor that will accommodate humans during the benchmark. The coordinates of the target area along with the reference frame will be communicated to the teams during the setup days. This area will only accommodate one person at a time.

The test area will be surrounded by motion tracking cameras, such as an Optitrack system, to locate the precise position of the person and automatically evaluate the performance of the robots in locating people.

Up to 10 persons will be chosen from the technical and organizing committee as well as from the team members of the competing teams during the setup days. Each team will be given the opportunity to meet the people and to collect training data before the beginning of the benchmark.

4.4.3 Expected Robot Behavior or Output

The participating team is required to position their robot on any desired initial position, that could be anywhere outside the specified target area, from which it can best perceive this area.

The human targets that the robot is required to perceive will move into the target area one at the time. The identity and location of the person presented to the robot is unknown before he/she moves and stands inside the area. For each presented person, the robot must perform the following:

- Person localization: Locate the 2D position of the person with respect to the benchmark setup reference frame.
- Person recognition: Recognize the person inside the target area.

For each person, the robot must communicate the results to the RSBB, or announce that it is incapable of perceiving the person. In addition, for the interest of the audience, the robot must announce the name of the recognized person after recognition.

4.4.4 Procedures and Rules

All teams are required to perform this functionality benchmark according to the steps mentioned below:

Step 1 The participating team is required to set their robot on any desired initial position outside the target area.

Step 2 Once the start signal is sent out from the RSBB, the robot is required to detect and localize the person standing inside the target area.

Step 3 The robot sends the results to the RSBB and announces its readiness for the next person.

Step 4 The RSBB requests the person to move out of the area and the randomly selects the next person to move into the area.

This process is repeated until all people are presented to the robot. Teams will have up to ten minutes to complete the functionality benchmark.

4.4.5 Acquisition of Benchmarking Data

During the execution of the benchmark, the Internal Data defined in Section C.3.2 will be collected.

4.4.6 Scoring and Ranking

Evaluation of the performance of a robot according to this functionality benchmark is based on:

1. The number and percentage of correctly recognized persons.
2. Location error for all detected persons.
3. Execution time (if less than the maximum allowed for the benchmark).

The previous criteria are in order of importance: the first criterion is applied first and teams will be scored according to the common accuracy metrics; the ties are broken by second criterion which is the position error. Since the position error is slightly affected by the precision of the ground truth system we will use a set of accuracy classes, and in case of ties we will resort to the third criterion.

4.5 Person Following Functionality

Human accompanying is an important capability required by many service robots for human-robot interactions. This functionality benchmark aims at assessing the capabilities of mobile robots in effectively following humans. The human following capability is expected to be one of the essential capabilities required for the robots planning to take part in the upcoming Smart City Major Competitions.

4.5.1 Functionality Description

The benchmark requires that the robot accompanies a human and always maintain a desired distance with this person. In this benchmark, a person will initially stand in front of the robot and then will start to walk around the arena. The robot must always follow this person while avoiding obstacles or other humans who are passing by.

4.5.2 Input Provided

The test is executed inside an apartment-like arena that will include stationary and dynamic obstacles such as furniture or people. The initial pose of the robot and the person will be communicated to the teams before the competition.

Since the main aim of this benchmark is to evaluate the following capabilities of robots, and the human-perception is not the main focus of the test, teams are open to exploit any preferred perceptual technique for their robots to display an efficient following behavior. This can range from anything such as laser range scanners, ultrasound sensors to vision-based perception systems. Furthermore, teams are allowed to instruct the person to wear tags, markers, colored shirt or anything that will minimize the perceptual errors.

4.5.3 Expected Robot Behavior or Output

Initially, a human will stand in front of the robot that is placed at a previously defined location in the arena. Once the benchmark starts, the person will start to walk and visit different regions of the arena. The robot is expected to follow and maintain its original distance with this person at all times while avoiding obstacles and other people that will interrupt the motion of the robot.

If the robot loses or cant keep up with the target, it can request the person to come back, walk slower, wait or stand in front of the robot to resume the walking behavior. However, such interruptions will have a direct impact on the performance score that is computed automatically by the system.

4.5.4 Procedures and Rules

All teams are required to perform this functionality benchmark according to the steps described below. The task must be done exclusively in autonomous mode and no teleoperation is permitted.

Step 1 The robot is placed on the starting position facing a standing human.

Step 2 The benchmark starts after the start signal is communicated by the RSBB.

Step 3 The person starts to walks in normal walking speed and visits different areas of the apartment, occasionally stopping and then resuming the walking. The robot must exhibit the following behaviour as described in Section 4.5.3 .

Step 4 The benchmark will stop automatically after the timeout. An End signal will be issued by the RSBB at timeout.

This benchmark will last 5 minutes.

4.5.5 Acquisition of Benchmarking Data

During the execution of the benchmark, the Internal Data defined in Section C.3.2 will be collected.

4.5.6 Scoring and Ranking

The evaluation of the performance of a robot is performed autonomously with the aid of a motion capture system that precisely measures the true pose of the robot and the person at all times. The performance score is computed based on:

1. The accuracy in keeping the desired relative distance to the target.

2. The total distance covered by the robot when it is following the target. The robot is automatically considered to be following when it is within a tolerance distance from the desired relative distance.

The previous criteria are considered to be of equal importance, encouraging both accurate and fast solutions with minimum interruptions.

4.6 Object Grasping and Manipulation Functionality

This functionality benchmark evaluates the capabilities of a robot to correctly grasp and manipulate objects. In particular, it assesses the object picking and placing capabilities of service robots that is an essential requirement for many consumer applications, for example to set up a dining table.

4.6.1 Functionality Description

In this benchmark the robots are required to correctly grasp, lift and move a set of objects over a tabletop. Objects will be placed one by one on top of a table located in front of the robot. The initial pose of the object is not known beforehand. The robot must grasp the object and deliver it to a target position over the table that is specified and communicated by the RSBB. Figure 2.6 describes an illustration of the benchmarking procedure.

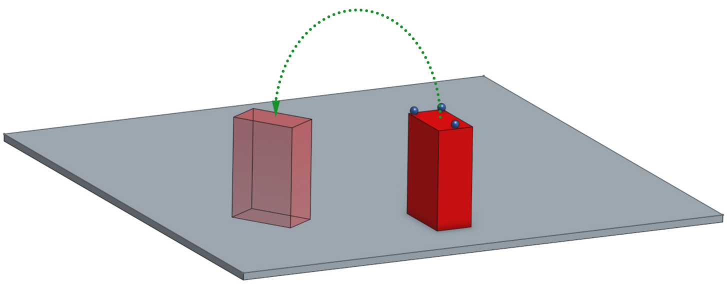


Figure 2.6: Illustration of the Grasping and manipulation functionality benchmark where the robot is required to grasp, pick up and place the object on a defined target position. Motion capture markers are attached on the object and are used for measuring the object pose and automatic evaluation

4.6.2 Input Provided

A set of 3D-printed objects of different shapes will be used for this benchmark. The 3D models of all objects, in digital format, will be available to the teams before the competition allowing them to train their robots in simulation or in real-world by 3D-printing the objects. Furthermore, objects are also provided to the teams during the setup days.

As the focus of this benchmark is on evaluation of the grasping and manipulation capabilities of robots and object, object perception is simplified by using 3D printed single-color objects over a white table. Furthermore, the nature of objects will be communicated to the robot through the RSBB to further assist the robot with the perception.

The reference frame will be marked on the table as in the object perception functionality, see Figure 2.4, and the test area will be surrounded by motion tracking cameras, such as an Optitrack system, to precisely locate the objects and evaluate the performance of robots.

4.6.3 Expected Robot Behavior or Output

The participating team is required to set their robot in front of the working table. After an object is placed on the table and the benchmark starts, the robot must grasp and pick up the object from the table surface and then deliver and place it on the target position specified by the RSBB. If required, the robot can further manipulate the object to improve the pose of the object. Once the robot has completed the task, or it is incapable of completing it, it must send the end command to the RSBB for the next object to be placed on the table.

4.6.4 Procedures and Rules

All teams are required to perform this functionality benchmark according to the steps mentioned below:

Step 1 The participating team is required to set their robot on any desired initial position in front of the working table.

Step 2 The start signal will be communicated to the robot by the RSBB to indicate the start of the benchmark.

Step 3 Once the robot announces its readiness to the RSBB by sending the Prepared signal, an object is randomly selected by the RSBB, from the list of possible objects, and is manually placed over the table by the referee.

Step 4 The execution command along with the target position of the object will be issued from the RSBB.

Step 5 The robot must grasp, pick up and place the object on the specified target location.

Step 6 The robot sends the end signal to the RSBB and announces its readiness for the next object.

This process is repeated until all objects are presented to the robot. Teams will have up to ten minutes to complete the functionality benchmark.

4.6.5 Acquisition of Benchmarking Data

During the execution of the benchmark, the Internal Data defined in Section C.3.2 will be collected together with the additional information described in the following table

[11] Any notification from the robot

[12] Internal robot data referring to end effector 3D position

Topic	Type	Frame Id	Notes
/erlc/notification[11]	std_msgs/String	–	–
/erlc/gripper_pose[12]	geometry_msgs/Pose	–	–

4.6.6 Scoring and Ranking

Evaluation of the performance of a robot for this functionality benchmark is performed autonomously and is based on:

1. The number and percentage of correctly grasped objects. An object is considered to be grasped by the RSBB system once it is lifted completely from the table.
2. Final pose error of the object with respect to the specified target location.
3. Execution time (if less than the maximum allowed for the benchmark).

The first criterion is applied first and teams will be scored according to the common accuracy metrics; the ties are broken by second criterion which is the position error. Since the position error can be slightly affected by the precision of the ground truth system, a set of accuracy classes is defined and used, in case of ties we will resort to the the execution time.

A ERL Consumer Award Categories

Individual tournament awards:
Awards will be given to the best teams for each of the Task Benchmarks (TBMs) and Functionality Benchmarks (FBMs) per tournament by the local organizers.

In every Local/Major Tournament, each team is requested to perform several *trials* of the available TBMs and FBMs. The team score for a given TBM/FBM in a Local/Major Tournament is computed as follows:

1. Select the best five trials of the TBM/FBM by the team (all the trials if less than five trial attempts are offered by the local organizers)
2. The team score is the median of the scores of the trials selected in 1.

The team with the highest score in each of the task benchmarks will be awarded the title ("ERL Consumer Best-in-class Task Benchmark <*task benchmark* title>"). The teams with the highest score ranking for each of the functionality benchmarks will be awarded the title ("ERL Consumer Best-in-Class Functionality Benchmark <*functionality benchmark* title>" and 'ERL Consumer Second-Best-in-Class Functionality Benchmark <*functionality benchmark* title>").

Notes:

- The number of TBM/FBM trial attempts offered in a tournament is recommended to be a minimum of 3 and a maximum of 7 trials.
- When a single team participates in a given benchmark, the corresponding benchmark award will only be given to that team if the Executive and Technical Committees consider the team performance of exceptional level.
- When less than three teams participate in a given *functionality benchmark*, only the "ERL Consumer Best-in-class Functionality Benchmark <*functionality benchmark* title>" award will be given to a team, and only if the Executive and Technical Committees consider that team's performance as excellent

Best ERL Consumer team award:
Only one final "Best ERL Consumer team" award will be awarded to the best performing team at the end of the Season. The Season ranking is computed as follows:

- To be eligible for an award, a team must have executed valid (i.e., scored) trials of at least one TBM in minimum two Local/Major Tournaments of the Season;
- Only TBMs are considered for computing the final score;
- A score is computed for every TBM as the median of the pooled trials used for scoring the best two performances of a team in the Local/Major Tournaments of the Season (see above);
- The team final score is then calculated by summing up all the TBM scores;
- A minimum score value will be required in order for the award to be given.

B The ERL Consumer Test bed

The test bed for ERL Consumer consists of the environment in which the competition will happen, including all the objects and artefacts in the environment, and the equipment brought into the environment for benchmarking purposes.

An environment fitting quite well the user story is depicted by Figure 7.

Figure 7: Granny Annie's apartment.

Note: There is considerable discussion in the community, as to how specific and precise the description of the environment should be, especially in the light of benchmarking. In order to require teams to develop robots which can be easily and flexibly adapted to a wide range of different environments, can deal with a wide variety of objects, etc., and to avoid overengineering of solutions, it would be highly advisable to keep the description as flexible as possible. We reflect this in the specifications below by providing comparatively generous *boundary conditions*, e.g. on room sizes, and by complementing them by a *recommendation*, which should be understood as a default size and the size we target to use at least for the first iteration of the competition. Competition organizers should plan with these recommended sizes and objects, while the developers of simulation software should provide means to easily modify models, e.g. by resizing rooms or by changing the properties of the environment or replacing objects, etc.

The *recommended* environment for the ERL Consumer Challenge is illustrated in Figure 8. Participating teams should assume the competition environment to be as illustrated; deviations should only occur if on-site constraints (space available, safety regulations) enforce them. An example of a certified ERL Consumer test bed and its corresponding environment is provided in Appendix E.

B.1 Environment Structure and Properties

The following set of scenario specifications must be met by the ERL Consumer environment.

Environment Specification B.1 (*Structured Environment*)

The environment consist of an ensemble of five spatial areas.

Figure 8: The test bed for the ERL Consumer Competition (Patio and Bathroom will not be accessible to the robots in this edition).

Two additional areas are foreseen for future extensions; they may be provided as part of a test bed, but they are not foreseen and not accessible to the robots, yet. We do not further consider them in the following specifications.

Environment Specification B.2 (*Flat Environment*)

All spatial areas all located on the same level, except where specified otherwise. There are no stairs in the environment.

Environment Specification B.3 (*Spatial Areas and Rooms*)

Spatial areas completely enclosed by walls are referred to as rooms. The apartment follows an open plan architecture, i.e. several spatial areas are connected to each other by open space ("openly connected") and are only jointly surrounded by walls. In human-robot interaction, such spatial areas may still referred to as rooms.

Environment Specification B.4 (*List of Rooms*)

The environment features the following five spatial areas: hallway, living room, dining room, kitchen, and bedroom.

Environment Specification B.5 (*Sizes of Spatial Areas*)

Robots are expected to cope with rooms and spatial areas of different sizes.

The minimum sizes of the spatial areas are as follows: hallway $120cm \times 200cm$*, bedroom* $400cm \times 300cm$*, kitchen* $200cm \times 240cm$*, dining room* $300cm \times 300cm$*, living room* $400cm \times 400cm$*.*

The whole apartment should fit into bounding rectangular box having a minimum area of $50m^2$ *and a maximum area of* $200m^2$*.*

The ***recommended sizes*** *of the spatial areas are as follows: hallway* $120cm \times 200cm$*, bedroom* $400cm \times 300cm$*, kitchen* $300cm \times 260cm$*, dining room* $300cm \times 300cm$*, living room* $400cm \times 580cm$*.*

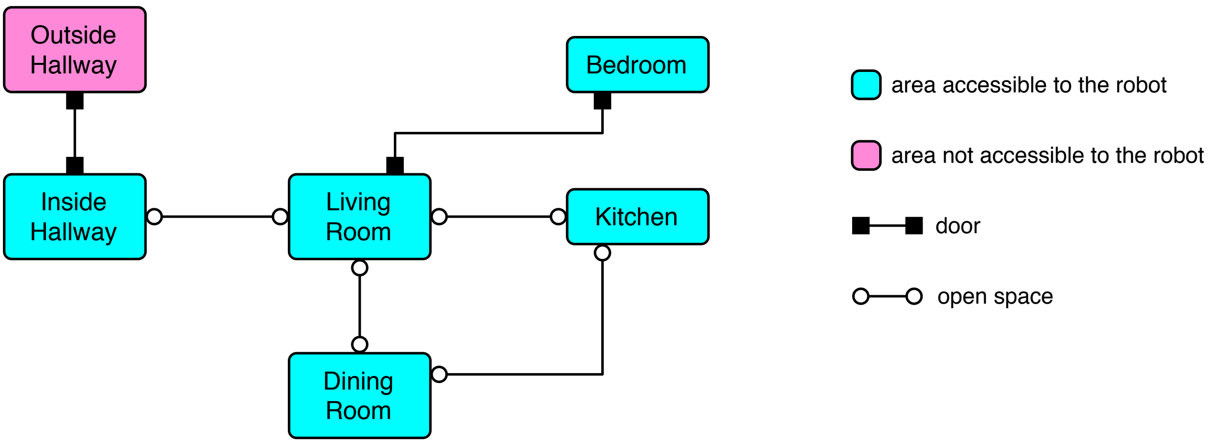


Figure 9: Graph showing the topological structure of environment.

Environment Specification B.6 (*Connectivity of Spatial Areas*)

The environment is accessible from outside through a front door to the hall way. The hallway is openly connected to the living room by a portal. The living room is connected to the bedroom by a door. The living room is openly connected to the kitchen. The living *room is openly connected to the dining room. The dining room is openly connected to the kitchen.*

The connectivity of the spatial areas is illustrated by the topological graph depicted in Figure 9.

Environment Specification B.7 (*Floors*)

The floor of each spatial area must be such that safe operation of robots meeting the specifications laid down in Section C is possible. The following criteria must be met:

Material features: *The floor is either carpet or parquet floor. No constraints exist with respect to the colors or patterns used.*

Slope: *The floor should be well-leveled, but slopes of up to* $2°$ *and unevenness of up to* $5mm$ *are acceptable.*

Uniqueness: *The floor may be unique or not, i.e. a floor in a room be the same as in other room, or it may be different.*

Environment Specification B.8 (*Walls*)

The walls of the environment must meet the following criteria:

Material features: *The bedroom walls have to be made of some stiff material, such as wood, wood-based materials (chipboards), stone, concrete, or metal. For competition arenas, walls will usually be made of chipboards in combination with wood or metallic frames. The color of the walls will usually be some light color (such as white, ivory, yellow, light green, to name a few examples). One or two walls per room may be painted with some darker color (such dark red, aubergine, mocca, dark grey, dark brown, to name a few examples) for decorative purposes. Patterned wallpaper may be used for finishing the walls. The walls are not translucent. If some kind of translucent material, such as glass, Plexiglass®, Perspec®, or Lucite®is used, then these wall areas are defined as windows.*

Shape/form: *The walls are upright. No slanted walls will be used. Offsets up to* $5cm$ *are allowed to ease construction of test beds.*

Size: *The minimum height of the walls is* $80cm$*. The recommended wall height is* $240cm$*. Exceptions may be made for up to two connecting walls of each room in order to allow better visibility for the audience at competitions. The length of the walls is defined by the connecting floor shapes. The width (thickness) of the walls must be large enough to ensure sufficient stability. The drawings assume a wall thickness of* $20cm$*. If the construction of an environment foresees inside walls with less thickness, then the space of the connecting spatial areas will grow accordingly.*

Uniqueness: *The walls may be unique or not, i.e. the wall colors and patterns may be the same as in other spatial areas of the apartment or they may be different.*

Environment Specification B.9 (*Ceilings*)

The rooms may or may not be covered by a ceiling. If it is not covered by a ceiling, then special constructions may be foreseen for fixing lamps, sensors, or other objects. If it is covered by a ceiling, the following specifications apply:

Material features: *The ceiling can be of any material.*

Slope: *The ceiling may be (partially or completely) sloped.*

Uniqueness: *The ceiling is not unique.*

Environment Specification B.10 (*Bedroom*)

The furniture includes a double bed, two bedside tables, a large wardrobe, a large mirror, a dressing table, and a carpet. The lighting includes two bedside lamps.

An example configuration of the bedroom is depicted in Figure 10.

Figure 10: An example bedroom created with the IKEA bedroom planner.

Environment Specification B.11 (*Living Room*)

The living room has a large window front side to the patio. These windows cannot be opened. These windows have inside blinds that are operated electrically. The furniture includes a large carpet, a coffee table, a couch, two armchairs, a low-height sideboard, and a bookshelf. A TV set is located on the sideboard. The lighting includes central ceiling lights, a band of dimmable ceiling spots, and a large floor light.

Environment Specification B.12 (*Dining Room*)

The dining room has a window and the furniture includes a dining table seating four, and four chairs.

Environment Specification B.13 (*Kitchen*)

The kitchen has no window and no door. The furniture includes several cupboards with drawers and doors. The installation include a fridge/freezer combination, a sink, a dish-washer, a stove, a baking oven, and an exhaust hood. Two rows of ceiling lights are available for the lighting. (Details of the kitchen will be specified in more detail as needed.)

Environment Specification B.14 (*Hallway*)

The hallway has no windows. The furniture consists of a coat rack. The lighting consists of lamps mounted on the walls.

B.2 Task-Relevant Objects in the Environment

The test bed environment will contain numerous objects, some of which are explicitly relevant for one or more of the *task benchmarks* or *functionality benchmarks* described in Section 3 and 4, respectively. We distinguish three major categories of task relevance:

Navigation-Relevant Objects: This class of objects comprises of all objects which have extent in physical space and do (or may) intersect (in 3D) with the robot's navigation space. All such objects must be avoided during navigation, i.e. whenever the robot moves, it may not bump into these objects or touch them, unless otherwise specified by a task. Navigation-relevant objects may be known by name or not. If these objects have a unique name, then the object may occur as a destination, e.g. for a navigation or manipulation operation.

Manipulation-Relevant Objects: This class contains all objects that the robot may have manipulative interactions with, which may include touching (a switch), grasping (a glass), lifting (a book), holding (a cup), placing (a parcel), dropping (waste), carrying (a glass), pushing (a drawer), pulling (a drawer), turning (a book), filling (a glass), pouring (from a cup), etc. For these objects, the most comprehensive information will be provided.

Perception-Relevant Objects: These are objects that the robot must "only" be able to perceive. By "perceive" we mean that the robot should be able to recognize if such an object is in its view, that it should be able to identify the object if it is unique or to classify it if not (e.g. an instance of a cup, if several non-unique instances exist), and that it should be able to localize the object. Objects that are only perception-relevant usually occur in tasks where the robot is supposed to find and localize these objects, but is not required to manipulate them.

Subsequently, we describe a complete collection of all objects relevant for the task benchmarks and functionality benchmarks.

B.2.1 Navigation-Relevant Objects

Environment Specification B.15 (*Navigation-Relevant Object Types*)

The navigation-relevant objects that may be present in the environment include the following types of objects:

- *Rugs, which may be placed on top of floors, covering the floor usually only partially.*
- *Furniture, which is placed in the environment.*
- *Doors, which connect rooms and may be in various different states.*
- *Any other kind of object, task-relevant or not, and including networked embedded devices and benchmarking equipment, if placed in the environment such that the object occupies space in the robot's workspace.*

Object Specification B.1 (*Rugs*)

The ERL Consumer Competition does not foresee any rugs in the environment, yet.

Object Specification B.2 (*Furniture*)

The furniture placed in each room or spatial area is listed in the environment specifications B.10 to B.14. Further details on the furniture are provided in the ERL wiki.

Object Specification B.3 (*Doors*)

The doors used in the environment have a door handle on both sides. Doors are dynamic objects that can be in different states at different times. At any time, a door may be in one of *the following four states: open, ajar, closed, locked. A door is considered open if its opening angle is 80° or more. A door is considered closed, if its opening angle is 0° and the door is latched but not locked. In case the door is locked as well, the door state obviously is locked. In all other cases, the door is considered to be ajar.*

B.2.2 Manipulation-Relevant Objects

Environment Specification B.16 (*Manipulation-Relevant Object Types*)

The manipulation-relevant objects that may be present in the environment include the following types of objects:

- *Personal items, like keys connected by key rings, mobile phones, tablet computers, MP3 players, eyeglasses/spectacles and their cases, wallets/purses and billfolds, watches, bracelets and rings.*
- *Mail items, like letters and parcels, and print material, like newspapers, magazines, journals, paperbacks and books.*
- *Household items, such as glasses, cups, plates, knifeware, and such.*
- *Groceries in containers of various forms, like cartons, cans, bottles, tubes, bags.*
- *Switches for lighting and electronic appliances.*
- *Handles of doors, including handles of wardrobe doors and drawers.*
- *Any other kind of object provided that it meets all of the object constraints B.1 to B.4.*

Object Specification B.4 (*Personal Items*)

The personal items to be used include:

- *keys with a key ring and lanyard.*
- *mobile phones.*
- *tablet computers (e.g. iPad).*
- *eyeglasses with cases, e.g. rimless, horn-rimmed or sunglasses.*
- *purses with different colors.*
- *wristwatches.*

Object Specification B.5 (*Mail Items and Print Material*)

The mail items to be used include:

- *A parcel as specified in Section 3.2.2.*

Object Specification B.6 (*Household Items*)

The household items to be used include:

- *coffee mugs, in a different color or pattern*
- *coffee cups with saucers, all identical.*
- *dessert plates*
- *cake plates*
- *small and large glasses*
- *water jugs*

- *coffee machine*
- *electric kettle*

Object Specification B.7 (*Groceries*)

The grocery items to be used include:

- *cartons of different size and coloring, containing food items such as cornflakes, cereal, pasta, salt, cornstarch,*
- *tin cans in at least three different sizes, containing tinned food such as tomato paste, sauerkraut, tuna, and fruits such as pears, peaches, and pineapples*
- *tin cans, all of the same size, but differently colored, containing soft drinks.*
- *glass jars in at least two different sizes, containing food such as pickled vegetables like cucumber, onions, corn, and beetroot, or pasta sauces, mustard, mayonnaise, or jams and jellies.*
- *PET bottles, containing water, soft drinks, or juices.*
- *No tubes or bags are foreseen yet.*

The next two object specifications concern objects relevant for manipulation, which themselves are embedded either directly into the environment or into objects placed into the environment, like furniture.

Object Specification B.8 (*Switches*)

Switches are buttons and may be latching (e.g. power button of a washing machine) or non-latching (e.g. power button of a PC). Digital switches are connected to either lamps or shutters. Switches are used as follows:

- *Switches embedded into walls for operating the lighting. These may be simple on/off switches, changeover switches, or intermediate switches.*
- *Switches embedded into walls for operating blinds or shutters. These switches come as a pair of pushbuttons, one for each direction of operation, and require to remain pushed for the duration of the operation.*

Object Specification B.9 (*Door Handles*)

The door handles to be used include:

- *Each of the doors in the environment has a door handle on each side of the door.*
- *Furniture features some handles, but they will not be relevant for manipulation.*

Fig. 2.2 illustrates the look and feel of one door handle in the environment.

Object Constraint B.1 (*Object Weight*)

The objects foreseen for manipulation can have a maximum weight of $1kg$*.*

Object Constraint B.2 (*Object Size*)

The default minimum width/length/depth/diameter/thickness (henceforth: size) of an object foreseen for manipulation is $2cm$*, and the default sum of the length, width, and height of the smallest bounding box around the object (henceforth: box sum) is* $6cm$*. An object may have a lower size than* $2cm$*, down to* $5mm$*, in up to two dimensions, if the other dimensions compensate for it, i.e. if the box sum is still at least* $6cm$*.*

Object Constraint B.3 (*Object Consistency, Rigidity, Stiffness*)

Any objects foreseen for manipulation tasks must be sufficiently rigid such that grasping by a robot is possible. There may be constraints on where objects can or may be grasped. Some objects may be foreseen which can appear in different shapes, e.g. glasses or a bunch of keys.

Object Constraint B.4 (*Object Content*)

Objects may not consist of or contain any kind of hazardous material. The content of objects may be solid matter (e.g. paper, nutrition), fluids (e.g. water or juices), or gases (e.g. air). If the object contains fluids or solid matter in the form of loose material (e.g. cornflakes or chips), the object must have a lid or other kind of fixture which ensures that the content is properly contained in the object and not spilled.

B.2.3 Perception-Relevant Objects

This section provides some clarifications with respect to perception.

Environment Specification B.17 (*Perception-Relevant Object Types*)

The perception-relevant objects in the environment include the following types of objects:

- *The basic environment structure including floors, walls, and ceilings.*
- *All navigation-relevant objects, including rugs, furniture, and any other physical object in the workspace.*

- *All dynamic navigation-relevant objects, i.e. objects with changeable state, like doors, windows, and some furniture.*
- *All manipulation-relevant objects, some of which may be uniquely identifiable while others are not (identical copies present)*
- *Target locations for navigation and manipulation may require the capability to identify objects (such as furniture items) that are not manipulation-relevant. Examples include objects the robot is supposed to move nearby (e.g. the bedside table) or objects the robot is supposed to grasp or place objects (e.g. the kitchen counter)*
- *Non-task-relevant objects (see Section B.3) will not have to be perceived as part of task benchmarks or functionality benchmarks, but due to their presence in the environment, they will present* ***perceptual noise****.*

The perception-relevant objects are already specified in various other sections. Further details on the target locations (and respective objects) are provided in the ERL wiki.

B.3 Non-Task-Relevant Objects in the Environment

The test bed environment for ERL Consumer is supposed to resemble a realistic apartment inhabited by an elderly person.Aside of the already listed and specified objects, which make up the environment itself, and objects relevant for navigation and manipulation, as well as networked embedded devices (see Section B.5) and benchmarking equipment (see Section B.6), the environment will contain numerous other objects, mostly for decoration and providing the required realism.

Environment Specification B.18 (*Non-Task-Relevant Object Types*)

The non-task-relevant objects that may be present in the environment include the following types of objects:

- *Textile objects, like curtains, tablecloth, placesets, napkins, and pillows, either affixed to walls/ceilings or sitting on top of furniture.*
- *Mirrors, usually fixed to walls or furniture.*
- *Lamps, on the floor or on top of furniture.*
- *Floristic objects, like flowers and plants, and associated objects like flower pots and vases, which may be on the floor, on window sills, or on top of furniture.*
- *Pictorial objects, like posters, photographs, drawings, and paintings, either affixed to walls or sitting on furniture.*
- *Other decorative objects, like plates and bowls, candles, and miniatures, usually sitting on top of furniture items.*

Object Specification B.10 (*Textile Objects*)

The textile objects to be used include:

- *curtains for the bedroom windows.*
- *tablecloths for the dining table.*
- *differently colored sets of textile placesets, each set consisting of four items.*
- *differently colored sets of textile napkins, each set consisting of four items.*
- *pillows for couch and chairs in the living room.*
- *linens, pillows, and coverlets for the bed in the bedroom.*

Object Specification B.11 (*Mirrors*)

The mirror objects to be used include:

- *a tall mirror, e.g. in the hallway.*
- *a large mirror, e.g. in the bedroom*

Object Specification B.12 (*Lamps*)

The lamps to be used include:

- *small lamp(s), e.g. on top of the bedside tables.*
- *floor lamp(s), e.g. in the living room.*
- *floor uplighters, e.g. in the living room.*

Note that ceiling lamps are already specified with the environment.

Object Specification B.13 (*Floristic Objects*)

The floristic objects to be used include:

- *small plants in pots, e.g. on the bedroom window sill.*
- *herbal plants in pots, e.g. on the kitchen window sill.*
- *large plants in pots, e.g in the living or dining room.*
- *small vases with flowers, e.g. on the kitchen counter.*
- *large vases with xerophytes, e.g. on the floor of the living room.*

Object Specification B.14 (*Pictorial Objects*)

The pictorial objects to be used include:

- *small framed pieces of photographs or drawings, each sized less than A4.*
- *medium-sized pieces of posters or prints, each sized about A3.*
- *large pieces of paintings or posters, each sized about A1.*

Object Specification B.15 (*Decoration Objects*)

The decoration objects to be used include:

- *plates*
- *bowls*
- *triplets of candles.*
- *miniatures or other decorative objects.*

B.4 Referee, Scoring and Benchmarking Box

The Referee, Scoring and Benchmarking Box (RSBB) software is available at the following repository:

`https://github.com/rockin-robot-challenge/rsbb`

During the benchmarks, a human referee enforces the rules. This referee must have a way to transmit his decisions to the robot, and receive some progress information. To achieve this in a practical way, an assistant referee is seated at a computer communicating verbally with the main referee. The assistant referee uses the Referee, Scoring and Benchmarking Box (RSBB). Besides basic starting and stopping functionality, the RSBB is also designed to receive scoring input and provide fine grained benchmark control for functionality benchmarks that require so.

The Referee, Scoring and Benchmarking Box was designed to support the following features:

Benchmark starting and stopping: Benchmark can only start if robot clock skew is below 100 milliseconds. Stop can be issued manually by the referee, by the robot if it completed the benchmark or automatically by the RSBB if the time for the benchmark is over or if the robot does not declare that it is saving offline data.

Devices communication: the ERL Consumer competition area includes automated home devices such as lights and window blinds. The RSBB provides an interface to control these devices, enabled only in certain benchmarks, so that the robot does not command the devices directly. The assistant referee can control the devices from his graphical interface.

Tablet communication: the ERL Consumer competition area includes a tablet device that can be used to communicate with the robot. Tablet communication passes through the RSBB and is enabled only for certain benchmarks.

Schedule: the full schedule of the competition is stored in the RSBB, allowing for automated progression with no setup time before each benchmark.

Online data: data produced by the robot during benchmarks falls in two categories: online and offline. Offline data is saved in a USB stick for latter analysis. Online data is transmitted to the RSBB. The RSBB displays and saves the data.

Logging: the RSBB saves a full log for each benchmark.

Referee interface: the RSBB includes a fully featured graphical interface to be used by the assistant referee.

Single client communication interface: the RSBB includes all features in a single communication interface. This way, participating teams only have to implement one communication mechanism.

State information: the RSBB continuously displays what state the benchmark is in.

Client libraries are available and should be integrated in the teams software `https://github.com/rockin-robot-challenge/at_home_rsbb_comm_ros`. This repository also includes the protocol to access the home automation devices and as well as the tablet application.

Passwords: a password will be given to each team at the beginning of the competition. These can be used to set up the private communication channel with the RSBB and to get access to the home automation devices.

Security note: the password security mechanism is only designed to prevent unintentional honest mistakes from the teams, like accessing the camera over WLAN, while another team is executing a benchmark. Any team caught trying to hack, circumvent or change the behaviour of any component described here for any purpose will be punished.

B.4.1 Communication between Benchmarking Equipment and Robots

For some types of internal benchmarking data (i.e. provided by the robot), logging is done on board the robot, and data are collected after the benchmark (for instance, via USB stick). Other types of internal benchmarking data, instead, are communicated by the robot to the test bed during the benchmark. In such cases, communication is done by interfacing the robot with standard wireless network devices (IEEE 802.11n) that are part of the test bed, and which therefore become a part of the benchmarking equipment of the test bed. However, it must be noted that network equipment is not strictly dedicated to benchmarking: for some benchmarks, in fact, the WLAN may be also (or exclusively) used to perform interaction between the robot and the test bed.

Due to the need to communicate with the test bed via the WLAN, all robots participating to the ERL Consumer Competition are required to:

1. possess a fully functional IEEE 802.11n network interface[13];
2. be able to keep the wireless network interface permanently connected to the test bed WLAN for the whole duration of the benchmarks

B.5 Networked Devices in the Environment

Networked sensors and actuators may be provided as a part of the environment. These devices are as enumerated and described below.

(Note: Currently, ERL consumer is not using these networked devices except for possible demos during Local Tournaments in the test beds equipped with them. Furthermore, the use of the fixed IP camera at the entrance is no longer permitted for any of the tasks)

Home automation controller: This device will run as a server on the local area network within the test bed. It will be accessible from all 'permitted' devices (wifi-enabled laptops/single board computers, etc. on the robots) on the same network. Using this controller, devices such as motorized window blinds/shutters, some of the room lights, motorized tiltable windows, etc. can be controlled. The controller will be able to receive messages (in a specific format provided to the teams) from the 'permitted' devices in order to control all the aforementioned devices. An example of such a controller can be found here[14].

Ethernet Camera: There will be a standard IP camera mounted a the front door, e.g. to retrieve images from the ringing visitors. The camera can have its parameters (frame rate, resolution, color gains) changed over Ethernet and it is not motor-controlled (no pan-tilt).

One way to access the camera is by receiving the MJPEG stream published by the camera[15]. During the setup days, the camera can be configured directly by each team to fit the needs (with admin but not root access). The OC/TC will save the configuration of each team and load them before each benchmark. Teams are free to use any protocol or configuration which the camera supports.

A ROS package that can be used to acquire images from the camera is available[16].

The diagram in Fig. 11 illustrates the network infrastructure which will be used throughout the competition. The network consists of the following devices:

- **Server**: computer used to manage the network.

[13]It must be stressed that full functionality also requires that the network interface must not be hampered by electromagnetic obstacles, for instance by mounting it within a metal structure and/or by employing inadequate antenna arrangements. Network spectrum in the Competition area is typically very crowded, and network equipment with impaired radio capabilities may not be capable of accessing

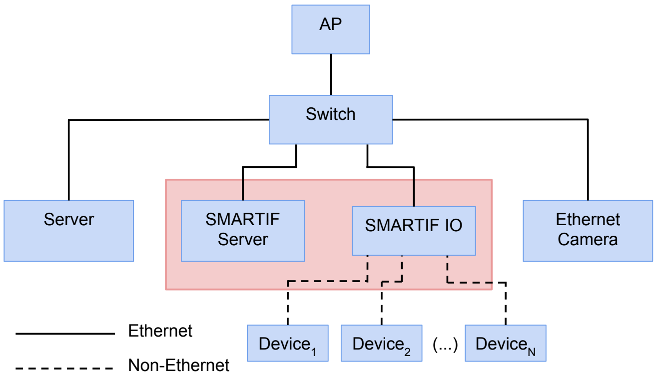


Figure 11: Network infrastructure of the ERL Consumer Competition

- **Switch:** Ethernet switch used to connect all the devices.
- **AP:** access point where the robot is supposed to connect to. This is the only connection between the robot and the network.
- **Ethernet Camera:** perspective camera outside at the main entrance door.
- **Devices:** different devices may exist in the environment, such as a motor to control the window blinds, controlled power plugs, light dimmers or a door bell button.
- **SMARTIF IO:** module to control the different devices/sensors existing in the house. They can only be accessed via the SMARTIF server. Teams do not have to interact with each device separately.
- **SMARTIF Server:** device which is responsible for the communication between the SMARTIF IO and the network. It can only be accessed from the server. Teams do not have to interact directly with it. Technical details regarding SMARTIF products can be found at the official site[17].

B.6 Benchmarking Equipment

ERL benchmarking is based on the processing of data collected in two ways:

- **Internal benchmarking data**, collected by the robot system under test (see Section C);
- **External benchmarking data**, collected by the equipment embedded into the test bed.

External benchmarking data is generated by the ERL test bed in different ways depending on their nature. One of the types of external benchmarking data used by ERL are pose data about robots and/or their constituent parts. To acquire these, ERL uses a camera-based commercial motion capture system composed of dedicated hardware and software. Benchmarking data has the form of a time series of poses of rigid elements of the robot (such as the base or a marker mounted in a known position). Pose data are acquired and logged by a customized external software system based on ROS (Robot Operating System): more precisely, logged data is saved as *bagfiles* created with the *rosbag* utility provided by ROS.

Other types of external benchmarking data are usually collected using devices that are specific to the benchmark and they are described in the context of the associated benchmark. Equipment

the test bed WLAN, even if correctly working in less critical conditions.

[14] http://rollertrol.com/store/en/vera-home-automation-control/87-vera-v3.html

[15] For example by using http://team_name:PASSWORD@10.0.0.2/mjpg/video.mjpg

[16] https://github.com/rockin-robot-challenge/at_home_ipcam

[17] http://www.smartif.com

to collect external benchmarking data includes any *server* which is part of the test bed and that the robot subjected to a benchmark has to access as part of the benchmark. Communication between servers and robot is performed via the test bed's own wireless network (see Section C.3).

C Robots and Teams

The purpose of this section is threefold:

1. It specifies information about various robot features that can be derived from the environment and the targeted tasks. These features are to be considered at least as desirable, if not required for a proper solution of the task. Nevertheless, we will try to leave the design space for solutions as large as possible and to avoid premature and unjustified constraints.

2. The robot features specified here should be supplied in detail for any robot participating in the competition. This is necessary in order to allow better assessment of competition and benchmark results later on.

3. It specifies the benchmarking equipment and data logging facilities required to perform benchmarking.

C.1 General Specifications and Constraints on Robots and Teams

Robot Specification C.1 (*Type/Class*)

A competition entry may use a single robot or multiple robots acting as a team.

Robot Specification C.2 (*Mobility Subsystems*)

At least one of the robots entered by a team must be mobile and able to visit different task-relevant locations by autonomous navigation. Teleoperation (using touch screens, tablets, mouse, keyboard, etc.) of robots for navigation is not permitted (except when otherwise specified, e.g., in particular instances of task and functionality benchmarks). The robot mobility must work in the kind of environments specified for ERL Consumer and on the kind of floors defined in the ERL Consumer environment specifications.

Robot Specification C.3 (*Sensor Subsystems*)

Any robot used by a team may use any kind of ***onboard*** *sensor subsystem, provided that the sensor system is admitted for use in the general public, its operation is safe at all times, and it does not interfere with other teams or the environment infrastructure.*

A team may use any kind of sensor system ***provided as part of the environment****, e.g. the networked camera specified in Section B.5, by correctly using a wireless communication protocol specified for such purpose and provided as part of the scenario. Sensor systems used for benchmarking and any other systems intended for exclusive use of the organisers are not accessible by the robot system.*

Robot Specification C.4 (*Communication Subsystems*)

Any robot used by a team may ***internally*** *use any kind of communication subsystem, provided that the communication system is admitted for use in the general public, its operation is safe at all times, and it does not interfere with other teams or the environment infrastructure.*

A robot team must be able to use the communication system provided ***as part of the environment*** *by correctly using a protocol specified for such purpose and provided as part of the scenario.*

Robot Specification C.5 (*Power Supply*)

Any mobile device (esp. robots) must be designed to be usable with an onboard power supply (e.g. a battery). The power supply should be sufficient to guarantee electrical autonomy for a duration exceeding the periods foreseen in the various benchmarks, before recharging of *batteries is necessary.*

Charging of robot batteries must be done outside of the competition environment. The team members are responsible for safe recharging of batteries. If a team plans to use inductive power transmission devices for charging the robots, they need to request permission from the event organizers in advance and at least 3 months before the competition. Detailed specifications about the inductive device need to be supplied with the request for permission.

Robot Constraint C.1 (*Computational Subsystems*)

Any robot or device used by a team as part of their solution approach must be suitably equipped with computational devices (such as onboard PCs, microcontrollers, or similar) with sufficient computational power to ensure safe autonomous operation. Robots and other devices may use external computational facilities, including Internet services and cloud computing to provide richer functionalities, but the safe operation of robots and devices may not depend on the availability of communication bandwidth and the status of external services.

Robot Constraint C.2 (*Safety and Security Aspects*)

For any device a team brings into the environment and/or the team area, and which features at least one actuator of any kind (mobility subsystems, robot manipulators, grasping devices, actuated sensors, signal-emitting devices, etc.), a mechanisms must be provided to immediately stop its operation in case of an emergency (emergency stop). For any device a team brings into the environment and/or the team area, it must guarantee safe and secure operation at all times. Event officials must be instructed about the means to stop such devices operating and how to switch them off in case of emergency situations.

Robot Constraint C.3 (*Environmental Aspects*)

Robots, devices, and apparatus causing pollution of air, such as combustion engines, or other mechanisms using chemical processes impacting the air, are not allowed.

Robots, devices, and any apparatus used should minimize noise pollution. In particular, very loud noise as well as well-audible constant noises (humming, etc.) should be avoided. The regulations of the country in which a competition or benchmark is taking place must be obeyed at all times. The event organizers will provide specific information in advance, if applicable.

Robots, devices, and any apparatus used should not be the cause of effects that are perceived as a nuisance to the humans in the environment. Examples of such effects include causing wind and drafts, strong heat sources or sinks, stenches, or sources for allergic reactions.

C.2 Safety Check and Robot Inspection

During the set-up days, all robots will be checked by the TC/OC for compliance with the specifications and constraints described in Section C.1. Teams will be asked to show the safety mechanisms of their robots and to demonstrate their use. A live demonstration is necessary: for example, pushing an emergency stop button while the robot is moving and verifying that the robot immediately stops. If the robot has other mechanical devices (e.g. a manipulator), their safety must be demonstrated as well.

This inspection can be done at any time during the set-up days. When teams are ready for an inspection, they can request one of the TC/OC members. The inspection can be repeated at any time during the competition days, upon request of the TC/OC. Referees, TC/OC members, team members and any other user who is interacting with the robot are always allowed to operate the safety mechanisms when there is a clear risk for the safety of any person or for the damage of any part of the environment.

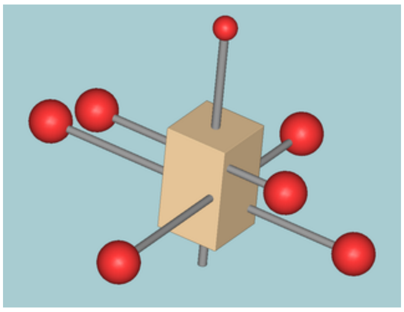
Robots that are not considered safe by the TC/OC are not allowed to participate in the competition!

C.3 Benchmarking Equipment and Data Logging on the Robot

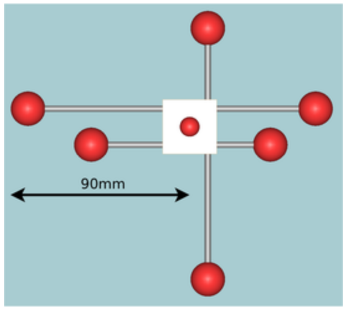
Whenever teams are required to install some element provided by ERL on (or in) their robots, such element will be carefully chosen in order to minimize the work required from teams and the impact on robot performance.

C.3.1 Setup for the Motion Capture System

During all task benchmarks, the pose of the robot will be captured by a motion capture system. For this, a *Marker Set*, similar to the one depicted in Fig. 12, which was used during the RoCKIn 2014 Competition, will be provided by ERL to be fitted on the robot.



(a) Perspective view



(b) Top view

Figure 12: RoCKIn Marker Set used during the RoCKIn 2014 Competition (the motion capture markers are illustrated as red spheres)

As part of robot internal data (see Section C.3.2) Teams must provide the position of the *Marker Set* in the test bed reference frame together with the robot estimated pose in the same reference frame. To do so, the static transformation between the origin of the reference frame of the base of the robot and the *Marker Set*, acquired through the motion capture system, will be provided to the Teams during the setup days.

The Marker Set will be provided to the Teams in advance with respect to the competition and the Teams require to mount it on the robot so to be above the robot and completely visible from the motion capture system. The robot must avoid collisions between the Marker Set and other objects (when assessing collisions for scoring, the Marker Set is considered as a part of the robot).

C.3.2 Internal Data Logging

During all task benchmarks, robots are required to log Internal data according to the following specifications. This data must be expressed in the reference frame of the test bed which will be clearly marked on it. It will be possible for teams to define such frame in their robot before the start of the competition. Fig. 13 illustrates one possible position of the test bed reference frame.

Only relevant data is expected to be logged (i.e. pointcloud used to recognize an object, more than one if an algorithm requiring multiple pointclouds is used). There are no restriction about the framerate: data can be saved, for the relevant parts of the benchmark, at the rate they are acquired or produced. The log may be a rosbag or the corresponding YAML representation, as

Figure 13: Example of a test bed reference frame for ERL Consumer. The z-axis points towards the reader.

specified in Section C.4, here we refer to the rosbag version, the corresponding YAML translation should be direct.

The list of topics to be logged (i.e., for all tasks and functionality benchmarks) is reported in the following table

Topic	Type	Frame Id	Notes
/erlc/robot_pose[18]	geometry_msgs/PoseStamped	/map	10 Hz
/erlc/marker_pose[19]	geometry_msgsPoseStamped	/map	10 Hz
/erlc/trajectory[20]	nav_msgs/Path	/map	Each (re)plan
/erlc/<device>/image[21]	sensor_msgs/Image	/<device>_frame	–
/erlc/<device>/camera_info[22]	sensor_msgs/CameraInfo	–	–
/erlc/depth_<id>/pointcloud[23]	sensor_msgs/PointCloud2	/depth_<id>_frame	–
/erlc/scan_<id>[24]	sensor_msgs/LaserScan	/laser_<id>_frame	10-40Hz
tf[25]	tf	–	–

The format for the name of the bag file to be saved by the Teams on their robot is the following:

[18] The 2D robot pose at the floor level, i.e., $z = 0$ and only yaw rotation.

[19] The 3D pose of the marker in 6 degrees of freedom.

[20] Trajectories planned by the robot, referred to the robot base, including when replanning.

[21] Image processed for object perception; <device> must be any of stereo_left, stereo_right, rgb; if multiple devices of type <device> are available on your robot, you can append "_0", "_1", and so on to the device name: e.g., "rgb_0", "stereo_left_2", and so on.

[22] Calibration info for /erlc/<device>/image.

[23] Point cloud processed for object perception; <id> is a counter starting from 0 to take into account the fact that multiple depth camera could be present on the robot: e.g., "depth_0", "depth_1", and so on.

[24] Laser scans, <id> is a counter starting from 0 to take into account the fact that multiple laser range finders could be present on the robot: e.g., "scan_0", "scan_1", and so on.

[25] The tf topic on the robot; the tf tree needs to contain the frames described in this table properly connected through the /base_frame which is the odometric center of the robot.

`{F|T}BM{H|W}{1|2|3}_YYYYMMDDhhmm_{teamname}.bag`

e.g., `FBMH1_201503041356_myteam.bag`, `TBMH3_201503041156_myteam.bag`, etc.

What data must be saved? Beside the data in the table, additional data the robot must save is specified in the particular benchmark subsection. Please note that some data streams (those with the highest bitrate) must be logged only in the time intervals when they are actually used by the robot to perform the activities required by the benchmark. In this way, system load and data bulk are minimized. For instance, whenever a benchmark includes object recognition activities, video and point cloud data must be logged by the robot only in the time intervals when it is actually performing object recognition.

What we do with the data? This data is not used during the competition. In particular, they are not used for scoring. The data are processed by ERL after the end of the competition for in-depth analyses and/or to produce datasets to be published for the benefit of the robotics community.

Where and when the robot must save the data? Robots must save the data, as specified in the particular benchmark subsection, on a USB stick provided by ERL. The USB stick is given to the team immediately before the start of the benchmark, and must be returned (with the required data on it) at the end of the benchmark.

NOTE: while the content of the data files saved by the robot is not used for scoring, **the existence of such files and their compliance to the specifications does influence the score of the robot**. Teams have the responsibility of ensuring that the required data files are saved, and of delivering them to the referee at the end of the benchmark. These aspects will be noted on the score sheet and considered for team ranking.

C.4 YAML Data File Specification

The subsequent paragraphs specify the YAML file format that can be converted to ROS bag files. This closely follows the data items described in D-2.1.7 [3]. The YAML format was chosen because it is a simple format, easy to produce without using any special library. Furthermore, the ROS messages format is already defined: as produced by the `rostopic echo` command.

C.4.1 File Format

The YAML file should be composed of a single list of messages. Each message should have four items:

- `topic` - The topic name.
- `secs` - Timestamp of the message, in number of seconds since 1970.
- `nsecs` - Nanoseconds component of the timestamp.
- `message` - The message, according to the topic type.

The message should be formatted in YAML, according to its structure. This is the same as the output of `rostopic echo`. However, binary fields may be specified in base 64 encoding for much smaller files. You can copy the file `src/base64.hpp` to your project, it depends only on boost to encode base 64.

And example for a file generated according to above specification could look as follows:


```
- topic: pose2d
  secs: 1397024209
  nsecs: 156423000
  message:
   x: 5.5
   y: 6
   theta: 6.4
- topic: image
  secs: 1397024210
  nsecs: 53585000
  message:
   header:
     seq: 306
     stamp:
       secs: 1397024210
       nsecs: 53585000
     frame_id: ''
   height: 4
   width: 4
   encoding: bgr8
   is_bigendian: 0
   step: 12
   data:
     !!binary JaU8JY0kGXUIAZ0UDWzgAXjgAb0kIglwbkGsnkWwoiWUfiGUhi2olhmUgc1YRaUw
```


C.4.2 YAML-to-ROSbag Conversion Tool

A tool to convert ERL Consumer YAML files into ROS bag files is available at the following Github repository:

`https://github.com/rockin-robot-challenge/benchmark_and_scoring_converter`

D General Procedure and Scoring for Task Benchmarks

General procedures, as well as scoring methods, are common to all Task Benchmarks. Those are detailed in this Appendix.

D.1 Safety check

Every run of each of the task benchmark will be preceded by a safety-check, outlined as follows:

1. The team members must ensure and inform at least one of the Organizing Committee (OC) or Technical Committee (TC) member, present during the execution of the task, that they have an emergency stop button on the robot which is fully functional. Any member of the OC/TC can ask the team to stop their robot at any time which must be done immediately.

2. A member of the OC/TC present during the execution of the task will make sure if the robot complies with the other safety-related rules and robot specifications presented in C.

D.2 General Procedures

This section specifies the procedures that will be followed for the start, restart or exit of each TBM. All teams are required to perform each task according to the steps mentioned in the rules and procedures sections for the tasks.

D.2.1 Start Procedure

The robots must be prepared outside of the apartment, in particular in a preparation area outside at one of the doors that has been designed to start the test. This preparation area is reserved for the next team in the schedule and can be accessed about 5 minutes before the start of the test. Any other preparation must be done at the own team area or in any other location that does not interfere with the competition.

The referee will inform the two participating team 2 minutes before the start of the test. After 2 minutes, the referee will start the test, i.e. s/he starts the timer (no delays for any reason). From this moment on, the robot is allowed to enter the apartment. If the robot is not ready and team members are still working on it after the test is started, there will be no penalty. But the time will run on. Whenever the robot enters the apartment, the team is not allowed to operate the robot in any way (e.g., touching any device, using a mouse, keyboard or touch screen, also remotely). In some task benchmarks it may be required that a person, e.g. guides the robot through the environment. In such a case the description of the respective task benchmark will relax the previous rule. Only those actions described in the particular test are allowed inside the apartment.

For each test, a desired location that the robot has to reach inside the apartment will be communicated to the teams during the set-up days. Entering the apartment must be done with a natural behavior (no joystick, keyboard, remote control, etc.). Autonomous navigation is the preferred solution, but following a person (e.g. a team member) is also an alternative way to guide the robot into the apartment. Using an easy-to-use interface may be considered as a natural behavior, however this must be approved beforehand by the by TC. In case of non-fully autonomous behaviors, the teams *must* verify with the TC in advance that their solution is suitable.

If a team prepares a behavior which is not within the scope for a particular test, it should contact the TC by e-mail *at least one week before the competition.*

D.2.2 Restart Procedure

Within the first 2 minutes after the robot enters the apartment and within the first 5 minutes from the start a test, the team can request for a restart. In this case the team is allowed to enter the apartment, bring the robot outside and perform any operation on the robot. It is not allowed to work on the robot inside the apartment (even if it is a quick and simple operation). Whenever the robot is ready, it can re-enter the arena and restart the test. The restart can be done only once for each run of the test. No penalties will be given for a restart. But any score achieved before the restart will be canceled and the time will not be stopped during the restart procedure.

D.2.3 Exit Procedure

After the end of the test, as communicated by the referee(s), the robot must quickly exit the apartment from the door designated for the test (which is usually different from the entrance door). The team members are allowed to manually drive, push or lift the robot. A penalty (in terms of an absolute negative score) will be given to the team if the robot is not outside the arena 2 minutes after the end of the test.

D.3 Scoring and Ranking

Evaluation of the performance of a robot (a.k.a. *scoring*) in Task Benchmarks is based on performance equivalence classes.

The criterion defining the performance equivalence class of robots is based on the concept of *tasks required achievements*, while the ranking of the robot within each equivalence class is obtained by looking at the performance criteria. In particular:

- The performance of any robot belonging to performance class N is considered as better than the performance of any robot belonging to performance class M whenever $M < N$
- Considering two robots belonging to the same class, then a penalization criterion (penalties are defined according to task performance criteria) is used and the performance of the one which received less penalization is considered as better
- If the two robots received the same amount of penalization, the performance of the one which finished the task more quickly is considered as better (unless not being able to reach a given achievement within a given time is explicitly considered as a penalty).

Performance equivalence classes and in-class ranking of the robots are determined according to three sets:

- A set A of **achievements**, i.e. things that should happen (what the robot is expected to do).
- A set PB of **penalized behaviors**, i.e. robot behaviors that are penalized, if they happen, (e.g., hitting furniture).
- A set DB of **disqualifying behaviors**, i.e. robot behaviors that absolutely must not happen (e.g. hitting people).

Scoring is implemented with the following 3-step sorting algorithm:

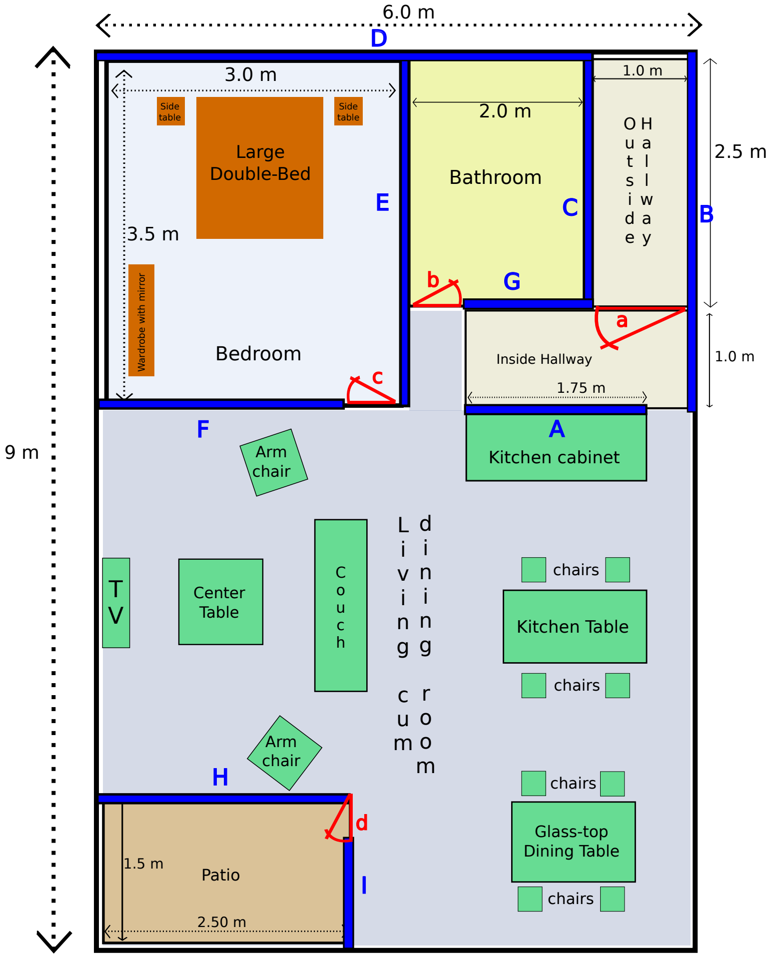
1. If one or more of the elements of set DB occur during task execution, the robot gets disqualified (i.e. assigned to the lowest possible performance class, called class 0), and no further scoring procedures are performed.

2. Performance equivalence class X is assigned to the robot, where X corresponds to the number of achievements in set A that have been accomplished.

3. Whenever an element of set PB occurs, a penalization is assigned to the robot (without changing its performance class).

One key property of this scoring system is that a robot that executes the required task completely will always be placed into a higher performance class than a robot that executes the task partially. Moreover the penalties do not make a robot change class (also in the case of incomplete task).

E The ISRoboNet@Home Test Bed at IST



	Width	Height
A	1.75 m	1.50 m
B	3.50 m	1.00 m
C	2.50 m	
D	5.00 m	
E	3.50 m	
F + c	3.00 m	
G + b	2.00 m	
H	2.50 m	
I + d	1.50 m	
a	1.00 m	

* Only wall A is the high wall
* A - I in blue are wall segments
* a - d in red are doors
* doors c,b and d should have a minimum width of 80 cm

Figure 14: Layout and dimensions of the @Home test bed at ISR/IST, Lisbon.

Based on the general design and specifications of the ERL Consumer test bed detailed previously in this text, in this sub-section we present the exact design specifications of the ISRoboNet@Home test bed installed at the premises of the Institute for Systems and Robotics (ISR) of IST, University of Lisbon. Note that this test bed is not an exact replica of the actual ERL Consumer Competition test bed but fits its general specifications and, as such, can be seen as a concrete example of using them for an actual implementation. Pictures of this test bed are presented in Figures 15–17.

E.1 Environment Structure and Properties

- Ensemble of five spatial areas accessible to the robots and three others inaccessible. Rooms and spatial areas (accessible to the robot): Living room, dining room, kitchen, inside hallway, bedroom. Spatial areas (inaccessible to the robot): outside hallway, bathroom, patio.
- Flat with no stairs
- Open-plan architecture followed for the living room, dining room and kitchen. The bedroom is separated by walls.
- Sizes of spatial areas: Please refer to Figure 14.

- Connectivity of spatial areas: Same as depicted in Figure 9.
- Floor: Parquet, well-leveled and uniform all over the test bed.
- Walls: Final version not yet in place – will be reported in version 3 of this document.
- Ceilings: Uniform false roof made of coated and perforated aluminum segments without slopes.
- Bedroom specifications (and furniture): one open-able and tilt-able window, a double bed, two side tables, two table lamps and one large wardrobe with mirror.
- Living room specification (and furniture's): contains windows that cannot be opened, couch, two armchairs, one coffee table, one TV table and one large floor lamp.
- Dining room specification (and furniture's): One glass-top dining table and 2 dining chairs.
- Kitchen specification (and furniture's): One kitchen table and 2 chairs, kitchen cabinet with multiple drawers and wash sink, two wall-mounted kitchen shelves.
- Hallway: consists of one coat rack.

E.2 Objects in the Environment

A list of all objects present in the environment of this test bed is given below through Tables 1 to 3. As most of the objects were purchased from the IKEA furniture store, the IKEA-reference code of the objects are provided to facilitate the readers of this documents. Note that this reference code is from the Portuguese version of IKEA's homepage[26].

E.3 Network Devices

The @Home test bed at IST is equipped with network devices capable of opening/closing the blind and turning on/off the lamps. The network is organized as shown in Figure 11 followed by a description of each block.

- Server: A computer used to manage the network.
- Switch: An Ethernet switch used to connect all the devices.
- AP: An Access Point where the robot is supposed to connect. This is the only connection between the robot and the network. Acts as a bridge between WLAN and LAN. The Access Points used work in Dual-band Standalone 802.11a/g/n. The models used are Cisco AIR - AP1042N-E-K9 [27].
- Ethernet Camera: Perspective camera facing the Outside Hallway. The camera can have its parameters (frame rate, resolution, color gains) changed over Ethernet and it is not motor controlled (no pan-tilt). The model of the camera can be found here [28].
- Devices: Different devices may exist in the house. In our test bed the devices are: a motor to control the window blinds, 3 controlled power plugs, 1 light dimmer, and 1 door bell button.

[26] www.ikea.pt

[27] http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-1140-series/data_sheet_c78-609338.html

[28] http://www.axis.com/products/cam_p1344

Task-relevant Objects					
Navigation-related					
Object	**Quantity**	**IKEA Code**	**Size (cm)**	**Ref-code**	**Observations**
Double bed	1	BRUSALI	140x200	702.499.07	bed-frame
	1	BRUSALI	140x200	901.245.34	bars
Matress	1	HAFSLO	140x200x18	602.443.64	
Slatted bed base	1	SULTAN LÖDINGEN	140x200	401.602.37	
Bedside tables	2	BRUSALI	44x36	502.501.57	
Wardrobe + mirror	1	BRUSALI	131x190	402.501.67	
Rug	1	HAMPEN	80x80	502.037.88	green
	1	HAMPEN	80x80	102.037.90	red
Coffee table	1	LACK	90x55	401.042.94	black
Couch	1	KLIPPAN		100.722.56	couch
	1			202.788.55	cover
Armchairs	2	PELLO		500.784.64	
Bookshelf	1	BORGSJÖ	75x181 cm	002.209.50	shelf
				202.209.54	doors in glass
Dining table	1	INGATORP	59/88/117x78	802.214.27	without glass
Dining Chairs	3	SIGURD		002.522.48	black
	1	KAUSTBY		400.441.96	brown
Kitchen cupboard		FYNDIG			white
	1		80x60x86	702.266.80	closet with doors
	1		126x60.6	502.375.33	top cover
	1		40x60x86	702.266.75	closet with 1 door and 1 drawer
	1		70x50	502.021.33	sink
	1	SUNDSVIK		800.318.61	tap
	2	SATTA		602.700.70	(1 red and 1 transparent)
Coat Rack	1	KROGKIG	128 (height)	201.745.08	multi-color
	1	HEMNES	185 (height)	002.468.70	black
TV table	1	BYAS	160x42x45	802.277.97	with drawers
Kitchen Shelf	2	EKBY JARPEN / VALTER	119x28	699.265.93	
Glass Dining table	1	GLIVARP	75/110x70	802.423.02	with glass

Table 1: List of task-relevant navigation-related objects in the environment

Task-relevant Objects				
Manipulation-related				
Object	**Quantity**	**IKEA Code**	**Ref-code**	**Observations**
Coffee Mugs	8	FÄRGRIK (2) + OMBYTLIG (1) + TECKEN (1) + UNGDOM (1 pack of 4)	401.439.93 + 202.099.80 + 702.160.49 + 702.348.97	
Coffee cups	6	DINERA	001.525.50	
Dessert plates	1	ÖVERENS	202.097.20	(1 pack of 6)
Cake plate	1	ARV BRÖLLOP	401.255.50	
Small glasses	6	GODIS	800.921.09	
Large glasses	6	POKAL + KROKETT	102.704.78 + 201.952.52	
Water jug	1	LÖNSAM + VÄNLIG	202.135.43 + 101.316.99	

Table 2: List of task-relevant manipulation-related objects in the environment

- SMARTIF IO: This module controls the different devices/sensors existing in the house. It is prepared to add more devices in case of need.
- SMARTIF Server: Device responsible for the communication between the SMARTIF IO mentioned above and the network. It is also where the system configurations (through the "SMARTIF Config Too") are stored and changed. Technical details regarding SMARTIF products can be found at the official site [29].

In our network, robots are supposed to communicate with the devices by sending a message to a specific IP and port. A SDK existent on the server will receive that message and transmit it to the SMARTIF IO witch will then control the device. Images from the Ethernet camera are also available through the AP. The quick set-up in the SMARTIF Configuration Tool, along with the possibility of adding/removing more devices, allows us to change the network if needed and with ease.

[29] `http://www.smartif.com/smarthome/techspecs.html`

Non-task-relevant Objects				
Object	**Quantity**	**IKEA Code**	**Ref-code**	**Observations**
Curtains	1	RITVA	145x300	24 - number in the curtains' section. 1 pack of 2
Table mats	8		102.361.11	
Napkins	1		101.012.73	pack of 50
Couch Pillows	3	FJADRAR	400.667.39	inside of the pillow
	1	STOCKHOLM	302.366.76	cover of the pillow (multi-color, squares)
	1	GURLI	202.496.03	cover (plain blue)
	1	SVARTTALL	002.897.13	cover (spots)
Linien for bed	1	DVALA	401.499.52	
	1	SVARTTALL	602.911.38	
Pillow for bed	2	GOSA VADD	501.291.66	
Lamp (small, bed side)	2	KVART	601.524.58	
Lamp (floor, living room)	1	SAMTID	202.865.63	white
Plants in pots (small)	3	FEJKA	702.514.72	
Plants in pots (large)	1	FEJKA	302.340.07	
Small picture frames with pics	6	NYTTJA	601.674.93	
Medium pic frames	5	NYTTJA	601.170.35	
Large paintings	1		102.340.46	each pack contains 3 pics without frame
Decoration bowl	2		901.244.02	bowl
	2		902.508.86	stuffing for the bowl (dry flowers)
Triples of candles	3	FLORERA	302.514.69	
Flower Jar	2	BLOMSTER	301.136.18	jar
	12	SNARTIG	101.391.91	flowers

Table 3: List of Non-task-relevant objects in the environment

Figure 15: Living room in the @Home test bed at IST

Figure 16: Living room, kitchen and dining area in the @Home test bed at IST

Figure 17: Bedroom in the @Home test bed at IST

Bibliography

[1] (2015) The RoCKIn Wiki. [Online]. Available: http://thewiki.rockinrobotchallenge.eu

[2] D. Croce, G. Castellucci, and E. Bastianelli, "Structured learning for semantic role labeling." *Intelligenza Artificiale*, vol. 6, no. 2, pp. 163–176, 2012.

[3] A. Ahmad, I. Awaad, F. Amigoni, J. Berghofer, R. Bischoff, A. Bonarini, R. Dwiputra, G. Fontana, F. Hegger, N. Hochgeschwender, L. Iocchi, G. Kraetzschmar, P. Lima, M. Matteucci, D. Nardi, V. Schaffionati, and S. Schneider, "Description of Ground Truth System," Politecnico di Milano, Milan, Italy," Deliverable D-2.1.7 (public), The RoCKIn Project (FP7-ICT-601012 grant agreement number 601012), 2014.