Funded by the European Union

EUROPEAN ROBOTICS LEAGUE
Brought to you by SPARC

Robotics Coordination Action
for Europe Two (RockEU2)
Horizon 2020

ERL-SR

– European Robotic League for Service Robots –

I: ERL-SR in a Nutshell
II: ERL-SR Rule Book

Latest Update: September 1, 2016
Revision: 1.1

Editors: Meysam Basiri, Pedro Lima, Matteo Matteucci, Luca Iocchi

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 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
KUKA
POLITECNICO DI MILANO

Introduction

The European Robotic League (ERL) is an innovative concept for robot competitions. The ERL is composed of multiple ''Local tournaments'', held in different research labs across Europe, with certified test beds, and a few competitions as part of ''Major tournaments'' such as RoboCup. Teams participate in a minimum of 2 tournaments (Local and/or Major) per year and get scores based on their performances. A final end of year score is computed for each team, per Task Benchmark (TBM) and Functionality Benchmark (FBM), using the best two participation in tournaments, and teams are ranked based on their final score. Prizes for the top teams (per TBM and FBM where they have participated) will be awarded during the next year's European Robotics Forum (ERF). For more information, please visit:

`http://sparc-robotics.eu/the-european-robotics-league/`

The **ERL Service Robots** tournament stems from its predecessor, the **Rockin@Home** competition, and focuses on the domain of service robotics for home application. This document describes the rules of the ERL Service Robots which is in fact an updated version of the latest rules of the RoCKIn@Home[1] 2015 competition to include the two additional task benchmarks (TBMS 5.5 and 5.6) that are shared with the **RoboCup@Home** competition.

Note: This version of the ERL-SR rule book is valid until 28.02.2017.

[1] `http://rockinrobotchallenge.eu/rockin_d2.1.3.pdf`

Contents

I: ERL-SR in a Nutshell

1 Introduction

ERL-SR is a competition that aims at bringing together the benefits of scientific benchmarking with the attraction of scientific competitions in the realm of domestic service robotics. The objectives are to bolster research in 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.

2 The ERL-SR User Story

The basic idea is that we have *an elderly person, named "Granny Annie", who lives in an in ordinary apartment. Granny Annie is suffering from typical problems of aging 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-SR 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-SR Scenario

The ERL-SR scenario description is structured into three sections, environment, tasks, and robots, which constitute the first contribution to the rules for the competition:

- 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 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. 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-SR Environment

The goal of the ERL-SR environment is to reflect 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.
The following embedded devices will 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.

Figure 1.1: Granny Annie's apartment.

- The shutters on the bedroom or living room window are accessible and controllable via network.

3.2 ERL-SR Benchmarks

3.2.1 Task Benchmarks

The following task benchmarks have to be performed:

1. ***Getting to know my home:*** The robot is told to learn about a new environment. It is supposed to generate a semantic map of the apartment within a limited time frame. It is left to the teams how exactly they approach this task. For example, a team member may "demonstrate" the apartment by guiding the robot through the apartment, pointing to objects and speaking aloud their names. Alternatively, a robot may explore the environment completely autonomously. The robot may also interrogate a team member about the names of objects or places. At the end of the environment learning phase, the robot must show through a behavior the understanding of the environment.

2. ***Welcoming visitors:*** Granny Annie stays in bed because she is not feeling well. The robot will handle visitors, who arrive and ring the door bell, as follows:

    - The Deli Man delivers the breakfast; the actual person is changing almost daily, but they all have a Deli Man uniform. The robot guides the Deli Man to the kitchen, then guides him out again. The robot is supposed to always observe the stranger.
    - An unknown person, trying to sell magazine subscription is ringing. The robot will tell him good-bye without letting the person in.
    - Dr. Kimble is her doctor stopping by to see after her. He is a known acquaintance; the robot lets him in and guides him to the bedroom.

- The Postman rings the door bell and delivers mail and a parcel; the actual person is changing almost daily, but they all have a Postman uniform. The robot has to receive the deliveries, and farewells him.

If a visitor has been admitted, the robot guides him out after the visit and close the door.

3. ***Catering for Granny Annie's comfort:*** This task is aimed at providing little kinds of help for Granny Annie throughout the day. After waking up in the morning, the robot is called by Granny Annie by touching an icon on her tablet computer. She wants the robot to lift the shutters, tilt the window, and switch off the lights. Then Granny Annie lets the robot know that she wants to read, but cannot find her reading glasses at the bedside table. She asks the robot to find them for her. The robot is expected to search for them at places where the glasses are likely to be, taking into account Granny Annie's habits. Information on these habits will be provided. Other comfort duties include lowering the shutters to block bright sunshine, bringing Annie a book, a cup of tea, or a glass of water.

4. ***Visiting my home (Shared with RoboCup):*** In this task, the robot should visit a set of predefined waypoints while avoiding obstacles on its path, following a person outside the arena and guiding that person back to the arena. This TBM focuses on tracking and recognizing a previously unknown person, obstacle avoidance, obstacle interaction, and safe navigation in dynamic environments. For more detail please refer to the *Navigation* test of the *RoboCup* rule book:
   `http://www.robocupathome.org/rules/2016_rulebook.pdf`

5. ***General Purpose Service Robot (Shared with RoboCup):***

   This TBM focuses on Human-Robot Interaction and the integration of other robot abilities such as navigation and robot-object interaction. In this test the robot has to solve multiple tasks upon request. This test is not incorporated into any predefined story and there is neither a predefined order of tasks nor a predefined set of actions. The actions that are to be carried out by the robot are randomly generated by the referees. The command is composed by three actions, which the robot has to recognize and execute. For more detail please refer to the *General Purpose Service Robot* test of the *RoboCup* rule book:
   `http://www.robocupathome.org/rules/2016_rulebook.pdf`

3.2.2 Functionality Benchmarks

The following functionality benchmarks have to be performed:

1. **Object Perception:** This functionality benchmark aims at evaluating the ability of a robot to recognize and localize a wide range of objects. A set of objects, selected from the list of ERL-SR 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:[2] 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 (see “Information provided to the team”). ii) Object recognition: association between the perceived object and one of the object instances belonging to the selected class (see “Information provided to the team”). iii) Object localization: estimation of the 3D pose of the perceived object with respect to the surface of the table.

2. **Navigation:** This functionality benchmark assesses the robot's capability to correctly, safely, and autonomously navigate in a ordinary apartment. The task includes: the navigation in a apartment-like environment with furniture, walls, and doors, i.e. in a previously

[2]This functionality benchmark corresponds to one of the functionality benchmarks of RoCKIn@Work.

mapped area; avoiding collisions with different type of unknown obstacles, in unknown positions (not previously mapped); and navigate in the presence of people in the arena. This functionality benchmark will be performed inside the ERL-SR apartment. The robots will receive a list of waypoints they have to follow in the respective order from the Referee, Scoring and Benchmarking Box (RSBB). When the robot reached the intended waypoint, it must report back to the RSBB and move on to the next waypoint.

3. **Speech Understanding:** This functionality benchmark aims at evaluating the ability of a robot to understand speech commands that a user gives in a domestic environment. A list of commands will be selected among the set of recognizable commands (i.e. commands that the robot should be able to perform in the competition), and they will be given to the robot as prerecorded utterances to the system or directly spoken by a user. The final representation for the recognized commands will have to respect a command/arguments structure where each argument is instantiated according to the arguments of command evoking verb. This representation is referred to as *Command Frame Representation* (CFR) (e.g. "go to the living room" will correspond to `GO( destination:"living room" )`).

3.3 ERL-SR 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.

II: ERL-SR Rule Book

1 Introduction to ERL-SR

ERL-SR is a competition that aims at bringing together the benefits of scientific benchmarking with the attraction of scientific competitions in the realm of domestic service robotics. The following ***user story*** is the basis upon which the ERL-SR Competition is built:

> *An elderly person, named "Granny Annie", lives in her own apartment together with some pets. Granny Annie is suffering from typical problems of aging 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 regularly. Sometimes she has days where she is 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-SR is looking into ways to support Granny Annie in mastering her life. A more detailed account of ERL-SR, but still targeted towards a general audience, is given in the ERL-SR in a Nutshell document, which gives a brief introduction to the very idea of ERL and ERL-SR, the underlying user story, and surveys the scenario, including the environment for user story, the tasks to be performed, and the robots targeted. Furthermore, this document already gives general descriptions of the task benchmarks and the functionality benchmarks that make up ERL-SR.

The document on hand is the rule book for ERL-SR, and it is assumed that the reader has already read the nutshell document. The audience for the current document are teams who want to participate in the competition, the organizers of events where the ERL-SR Competition is supposed to be executed, 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.

This remainder of this document is structured as follows: Section 2, ***award categories*** surveys the number and kind of awards that will be awarded and how the ranking of the award categories is determined based on individual benchmark results. The ***test bed*** for ERL-SR Competitions is described in some detail in section (Section 3). Subsections are devoted to the specification of the structure of the environment and its properties (3.1), to the objects in the environment relevant to the tasks on hand (3.2), 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) (3.3), to the networked devices embedded in the environment and accessible to the robot (3.5), and to the benchmarking equipment which we plan to install in the environment and which may impose additional constraints to the robot's behavior (equipment presenting obstacles to avoid) or add further perceptual noise (visible equipment) (3.6). Next (Section 4), we provide some specifications and constraints applying to the ***robots and teams*** permitted to participate in ERL-SR. The ERL consortium is striving to minimize such constraints, but for reasons of safety and practicality such constraints are required. After that, the next two sections describe in detail the ***task benchmarks*** (Section 5) and the ***functionality benchmarks*** (Section 6) comprising the ERL-SR Competition. While information on scoring and ranking the performance of participating teams on each benchmark is already provided in the benchmark descriptions.

2 ERL-SR Award Categories

Awards will be given to the best teams in each of the ERL-SR *task benchmarks* and *functionality benchmarks* that are described in Sections 5 and 6. For every local/major tournament, and for every task and functionality benchmark, a score is computed by taking the median of the best (up to 5) trials. The final end of year score is computed by taking the median of the pooled trials that were used for scoring the best two Local/Major tournaments and teams are ranked based on this score. The ERL Competition awards will be given in the form of cups for the best teams. Every team will also receive a plaquette with the ERL logo and a certificate.

Please note that teams need to participate in a minimum of 2 tournaments (Local and/or Major) per year in order to obtain a score for the TBMs and/or FBMs that they intend to enter.

2.1 Awards for Task Benchmarks

The team with the highest score in each of the five *task benchmarks* will be awarded a cup ("ERL-SR Best-in-class Task Benchmark <*task benchmark* title>"). When a single team participates in a given *task benchmark*, the corresponding *task benchmark* award will only be given to that team if the Executive and Technical Committees consider the team performance of exceptional level.

2.2 Awards for Functionality Benchmarks

The teams with the highest score ranking for each of the three *functionality benchmarks* will be awarded a cup ("ERL-SR Best-in-Class Functionality Benchmark <*functionality benchmark* title>" and 'ERL-SR Second-Best-in-Class Functionality Benchmark <*functionality benchmark* title>"). When less than three teams participate in a given *functionality benchmark*, only the "ERL-SR 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.

3 The ERL-SR Test bed

The test bed for ERL-SR 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 aspect that is comparatively new in robot competitions is that ERL-SR is, to the best of our knowledge, the first open competition targeting an *environment with ambient intelligence*, i.e. the environment is equipped with networked electronic devices the robot can communicate and interact with, and which allow the robot to exert control on certain environment artefacts.

An environment fitting quite well the user story is depicted by Figure 2.1.

Figure 2.1: 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-SR Competition is illustrated in Figure 2.2. 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.

3.1 Environment Structure and Properties

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

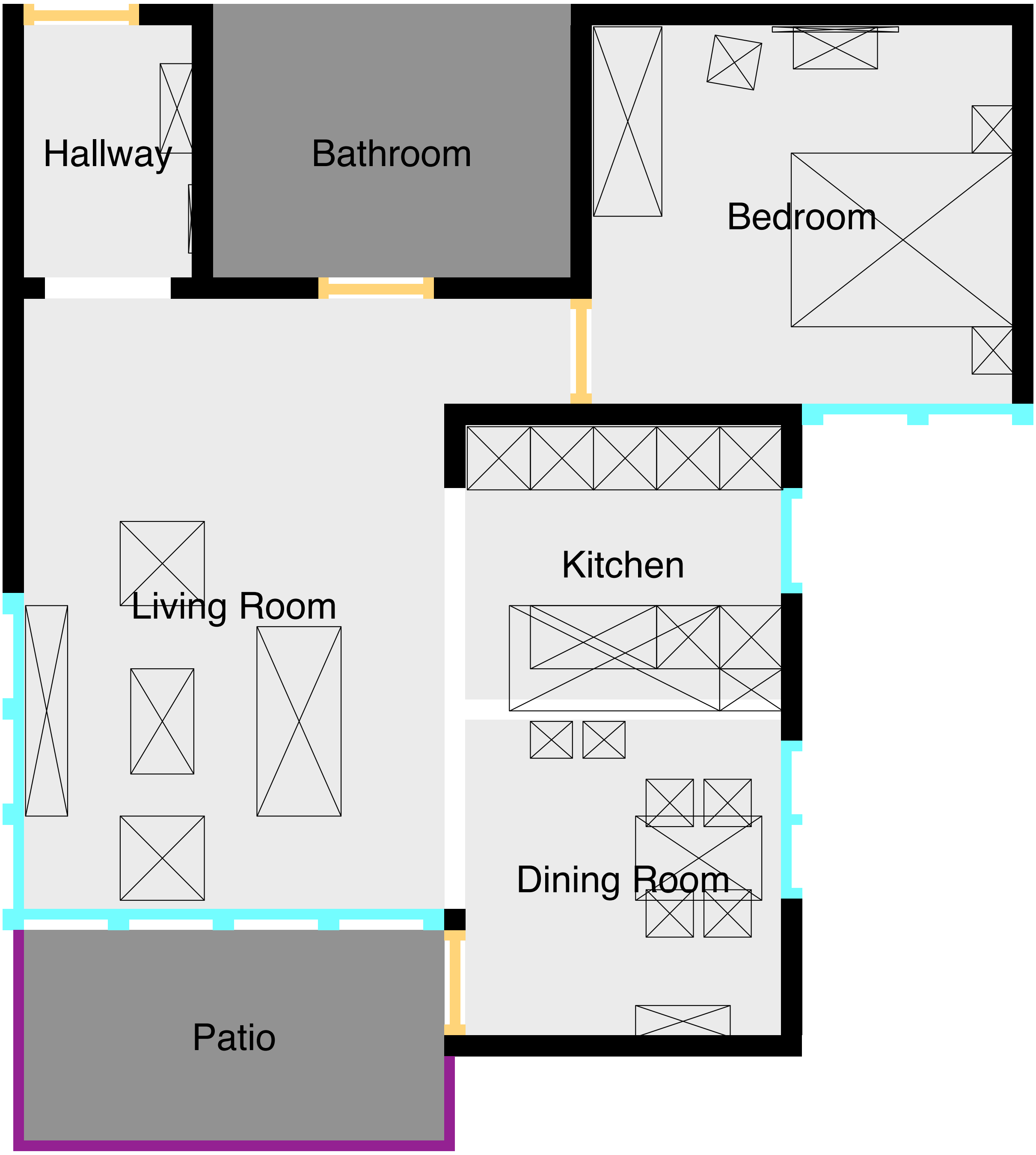


Figure 2.2: The test bed for the ERL-SR Competition (Patio and Bathroom will not be accessible to the robots in this edition).

Environment Specification 3.1 (*Structured Environment*)

The environment consist of an ensemble of five spatial areas.

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 3.2 (*Flat Environment*)

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

Environment Specification 3.3 (*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 3.4 (*List of Rooms*)

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

Environment Specification 3.5 (*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* $400cm \times 580cm$*.*

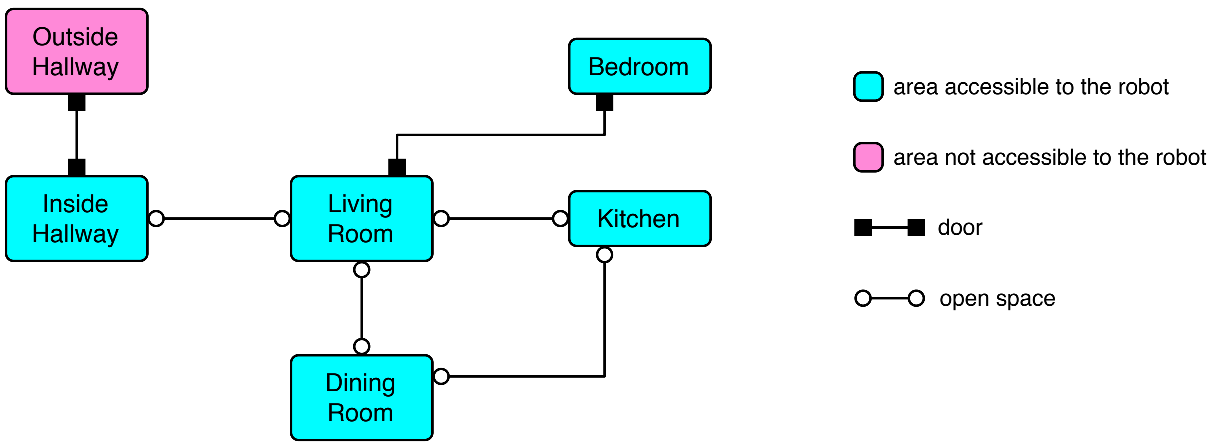


Figure 2.3: Graph showing the topological structure of environment.

Environment Specification 3.6 (*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 2.3.

Environment Specification 3.7 (*Floors*)

The floor of each spatial area must be such that safe operation of robots meeting the specifications laid down in Section 4 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 3.8 (*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 3.9 (*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 3.10 (*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 2.4.

Figure 2.4: An example bedroom created with the IKEA bedroom planner.

Environment Specification 3.11 (*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 3.12 (*Dining Room*)

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

Environment Specification 3.13 (*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 3.14 (*Hallway*)

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

3.2 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 5 and 6, 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.

3.2.1 Navigation-Relevant Objects

Environment Specification 3.15 (*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 3.1 (*Rugs*)

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

Object Specification 3.2 (*Furniture*)

The furniture placed in each room or spatial area is listed in the environment specifications 3.10 to 3.14. Further details on the furniture will be provided in due time and will be added as an appendix to this document.

Object Specification 3.3 (*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.*

3.2.2 Manipulation-Relevant Objects

Environment Specification 3.16 (*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 3.1 to 3.4.*

Object Specification 3.4 (*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 3.5 (*Mail Items and Print Material*)

The mail items to be used include:

- *A parcel as specified in Section 5.3.3.*

An example for such a parcel is illustrated in Figure 2.10.

Object Specification 3.6 (*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 3.7 (***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 3.8 (*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 3.9 (*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. 2.11 illustrates the look and feel of one door handle in the environment.

Object Constraint 3.1 (*Object Weight*)

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

Object Constraint 3.2 (*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 3.3 (*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 3.4 (*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.

3.2.3 Perception-Relevant Objects

This section provides some clarifications with respect to perception.

Environment Specification 3.17 (*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 3.3) 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) will be provided in due time and will be added as an appendix to this document.

3.3 Non-Task-Relevant Objects in the Environment

The test bed environment for ERL-SR 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 3.5) and benchmarking equipment (see Section 3.6), the environment will contain numerous other objects, mostly for decoration and providing the required realism.

Environment Specification 3.18 (*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 3.10 (*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 3.11 (*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 3.12 (*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 3.13 (*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 3.14 (*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 3.15 (*Decoration Objects*)

The decoration objects to be used include:

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

3.4 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/at_home_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. In the ERL-SR Competitions the use of the RefBox **will be mandatory** for task benchmarks 5.2,5.3 and 5.4 and for all the functionalities benchmarks.

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-SR 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-SR 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:

- for teams NOT using ROS: `https://github.com/rockin-robot-challenge/at_home_rsbb_comm`
- for teams using ROS: `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.

3.4.1 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-SR Competition are required to:

1. possess a fully functional IEEE 802.11n network interface[3];
2. be able to keep the wireless network interface permanently connected to the test bed WLAN for the whole duration of the benchmarks

3.5 Networked Devices in the Environment

In order to facilitate certain aspects of the tasks that need to be performed by the robot, networked sensors and actuators will be provided as a part of the environment. These devices are as enumerated and described below.

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[4].

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[5]. 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.

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

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

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

A ROS package that can be used to acquire images from the camera is available[6].

The above mentioned devices will be accessible to all participating teams only during the following time periods:

- To all teams during the setup days.
- To a particular team, during its time-slot for the actual run of a task or functionality benchmark on each of the competition days.

The diagram in Fig. 2.5 illustrates the network infrastructure which will be used throughout the competition. The network consists of the following devices:

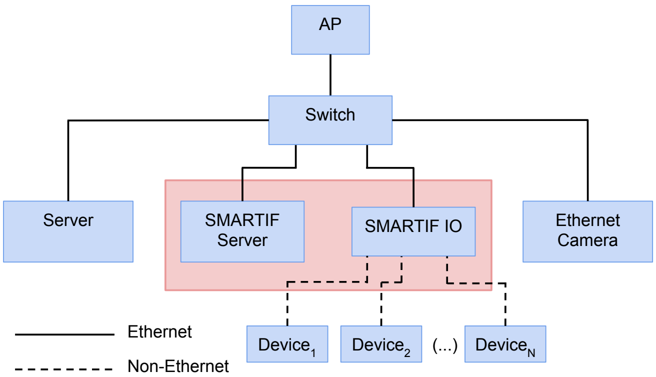


Figure 2.5: Network infrastructure of the ERL-SR Competition

- **Server**: computer used to manage the network.
- **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[7].

3.6 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 4);
- **External benchmarking data**, collected by the equipment embedded into the test bed.

[6] https://github.com/rockin-robot-challenge/at_home_ipcam
[7] http://www.smartif.com

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 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 4.3).

4 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.

4.1 General Specifications and Constraints on Robots and Teams

Robot Specification 4.1 (*Type/Class*)

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

Robot Specification 4.2 (*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-SR and on the kind of floors defined in the ERL-SR environment specifications.

Robot Specification 4.3 (*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 3.5, 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 4.4 (*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 4.5 (*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 4.1 (*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 4.2 (*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 4.3 (*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.

4.2 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 4.1. 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.

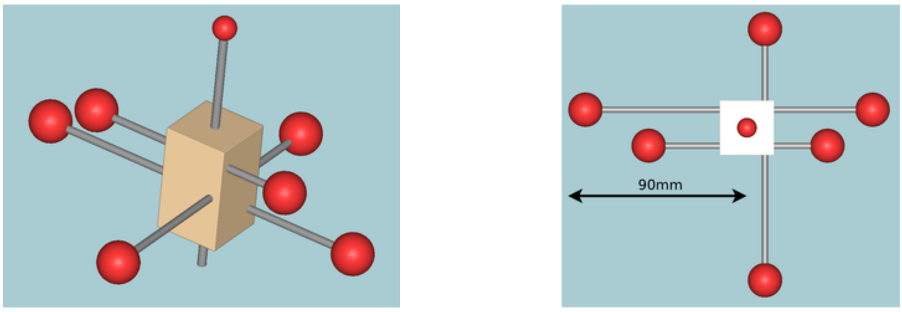
Robots that are not considered safe by the TC/OC are not allowed to participate in the competition!

4.3 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.

4.3.1 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. 2.6, 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 2.6: 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 4.3.2) 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).

4.3.2 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. 2.7 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 2.7: Example of a test bed reference frame for ERL-SR. The z-axis points towards the reader.

specified in Section 4.4, 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
/rockin/robot_pose[8]	geometry_msgs/PoseStamped	/map	10 Hz
/rockin/marker_pose[9]	geometry_msgsPoseStamped	/map	10 Hz
/rockin/trajectory[10]	nav_msgs/Path	/map	Each (re)plan
/rockin/<device>/image[11]	sensor_msgs/Image	/<device>_frame	–
/rockin/<device>/camera_info[12]	sensor_msgs/CameraInfo	–	–
/rockin/depth_<id>/pointcloud[13]	sensor_msgs/PointCloud2	/depth_<id>_frame	–
/rockin/scan_<id>[14]	sensor_msgs/LaserScan	/laser_<id>_frame	10-40Hz
tf[15]	tf	–	–

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

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

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

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

[11]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.

[12]Calibration info for /rockin/<device>/image.

[13]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.

[14]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.

[15]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.

4.4 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 [1]. 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.

4.4.1 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 JaU8JYOkGXUIAZOUDWzgAXjgAbOkIglwbkGsnkWwoiWUfiGUhi2olhmUgc1YRaUw
```


4.4.2 YAML-to-ROSbag Conversion Tool

A tool to convert ERL-SR YAML files into ROS bag files is available at the RoCKIn Github repository:

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

5 Task Benchmarks

Details concerning rules, procedures, as well as scoring and benchmarking methods, are common to all task benchmarks.

Rules and Procedures 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 Section 4.

All teams are required to perform each task according to the steps mentioned in the rules and procedures sections for the tasks.

Scoring and Ranking Evaluation of the performance of a robot according to 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).

One further common issue concerns all (task or functionality) benchmark runs in which a spoken sentence from a human to the robot is expected, and is specified as follows. ERL will provide to the teams at least eight weeks before the competition date:

- the lexicon that will be used, i.e., verbs, nouns referring to objects/locations/people, and terms for spatial relations
- a set of audio files and/or strings of text with examples of sentences

More precisely, ERL will define a large set of sentences for each phase of a test in which such sentences are needed. Then the complete lexicon will be extracted from this set and the Organizing Committee will select and publish a subset of it. In the actual runs a randomly chosen sentence in the remaining part of the data set will be used.

5.1 General Procedures

This section specifies the procedures that will be followed for the start, restart or exit of each TBM.

5.1.1 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.*

5.1.2 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.

5.1.3 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.

5.2 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 understand the changes in the environment either through user interaction or automatically or with a mixed approach.

5.2.1 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 one door, and perception-relevant objects (two pieces of furniture and three, perception-relevant or not, objects). The robot has to detect these changes, either automatically or with the help of a user (a team member), and provide an explicit representation of them 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).

5.2.2 Feature Variation

The following elements will be rearranged before each run:

- one door connecting two rooms
- two pieces of furniture
- three objects

The names of rooms, furniture and object categories will be distributed to the teams during the setup days and will be according to environment specifications in Section 3.1.

5.2.3 Input Provided

The teams must create a map of the environment and label it with pieces of furniture and objects in there during the set-up days. This information will be used for this task, together with the topological map, rooms and objects specifications described in Section 3.

5.2.4 Expected Robot Behavior or Output

Phase 1: knowledge acquisition.
The robot in any way (human-robot interaction (HRI) or autonomously or mixed) has to detect these changes and represent them in an explicit format (see Expected Output below). In case of an HRI-based approach, a team member can guide the robot in the environment, e.g. by following a person, and show the changes with only natural interactions (speech and gesture). No input devices are allowed (e.g., touch screens, tablets, mouse, keyboard, etc.). In this phase, the robot has to provide an output containing an explicit representation of the acquired knowledge. At any time the teams can decide to move to Phase 2, even if not all the 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 executing a user command mentioning one of the items affected by the change.

The user command must be given to the robot in a natural way. The preferred way is using speech interaction. Other natural ways are welcome, but should be agreed in advance with the TC, e.g., using an external or onboard keyboard should not be allowed.

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 will be evaluated:

- semantic map file
- pictures of objects/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)
type(door126,door)
connects(door36, kitchen, office).
connects(door126, kitchen, bathroom).
isOpen(door36, true).
isOpen(door126, false).
```

(b) Predicates about location of pieces of furniture

- **Definition:**

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

- **Example:**

```
type(ch1, kitchen_chair).
in(ch1, living_room).
```

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).
position(object_ID, [X, Y, Z]).
color(object_ID, color_name).
picture(object_ID, image_filename).
```

- **Example:**

```
type(obj33,apple).
in(obj33,kitchen).
on(obj33,kitchen_table).
position(obj33, [3.0, 3.0, 1.0]).
color(obj33,red).
picture(obj33,image_obj33.jpg).
```

Notes:

- Object_ID can be any valid identifier (one letter followed by additional letters, digits or underscore '_' symbols).
- The language is case insensitive (all lowercase is preferred).
- 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.
- In the position predicate *X, Y and Z* are in meters and in the global reference system predefined.

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 in the kitchen table (that is in the kitchen),
- a box of biscuits (with main color yellow) is placed on the coffee table (that is in the living room), and
- a green apple is placed on the kitchen chair moved to the living 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).

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

type(object_1, coke).
in(object_1, kitchen).
on(object_1, kitchen_table).
position(object_1, [3.0, 2.5, 1.0]).
color(object_1, red).
picture(object_1, object_1.jpg).

type(object_2, biscuits).
in(object_2, living_room).
on(object_2, coffee_table).
position(object_2, [11.0, 9.5, 0.5]).
color(object_2, yellow).
picture(object_2, object_2.jpg).

type(object_3, apple).
in(object_3, living_room).
on(object_3, kitchen_chair).
position(object_3, [23.0, 7.5, 0.5]).
color(object_3, green).
picture(object_3, object_3.jpg).
```


Tool: ERL provides a Python script to parse and evaluate a file against a ground truth, which can be found in the RoCKIn Wiki [2].

5.2.5 Procedures and Rules

Before the robot enters the apartment, any of the doors in the environment will be closed, two randomly-selected pieces of furniture will be moved from their original position to a different position (can even be either in the same room or in a different room) and any three randomly-selected objects will be placed in random places (above some piece of furniture). Note that the closed door could be also the entrance door. In this case, this door will be closed right 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, possibly accompanied by the user (a team member) and interacting with him/her: The robot has to detect the changes specified above, and then represent them in an explicit format as described in sub-subsection 5.2.4.

Phase 2 (Assuming that the robot has either finished the task requested in the previous steps, or has shown its incapability to do so and is physically idle) the robot will be asked (e.g. by receiving a voice command or by following a person, i.e., a team member) to move one of the objects recognised in Phase 1 to a piece of furniture, also recognised in Phase 1.

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.

The accomplishment of the behavior in Phase 2 will be rewarded only if it refers to an object/piece of furniture that has been correctly reported in the output of Phase 1.

Note also that the metric map can be generated offline (i.e., before the task run) and may not contain the changes. The metric map will not be evaluated specifically. However, a poor quality metric map or an out-of-date map can affect the evaluation of the position of the objects/piece of furniture selected for the task.

5.2.6 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 4.3.2 will be collected.

5.2.7 Scoring and Ranking

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

- The robot detects the door with changed state.
- The robot detects each piece of moved furniture.
- The robot detects each changed object.
- The robot correctly executes the command given in Phase 2.

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

- The robot requires multiple repetitions of human gesture/speech.
- The robot bumps into the furniture.

- The robot stops working.
- The robot was helped to manipulate an object.

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 another person 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.

5.3 Task *Welcoming visitors*

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

5.3.1 Task Description

Granny Annie stays in bed because she is not feeling well. The robot will handle visitors, who arrive and ring the door bell, as follows (doors are assumed to be always open):

- Dr. Kimble is her doctor stopping by to see after her. He is a known acquaintance; the robot lets him in and guides him to the bedroom.
- The Deli Man delivers the breakfast; the actual person is changing almost daily, but they all have a Deli Man uniform. The robot guides the Deli Man to the kitchen, then guides him out again. The robot is supposed to always observe the stranger.
- The Postman rings the door bell and delivers mail and a parcel; the actual person is changing almost daily, but they all have a Postman uniform. The robot just receives the deliveries, and farewells him.
- An unknown person, trying to sell magazine subscription is ringing. The robot will tell him good-bye without letting the person in.

The task involves handling several visitors arriving in any sequence, but separately from each other. The robot must be able to handle/interact with an Ethernet camera (see Section 3.5). If a visitor has been admitted, either by the face or the kind of uniform, the robot guides him out after the visit.

5.3.2 Feature Variation

In all the runs of this task the four persons indicated above will ring the door bell. The robot is thus required to deal with all the situations described above. However, the order in which the people will appear will be randomized for each run/participant. Every visit will terminate before the next one.

5.3.3 Input Provided

The following information about the people that will act as visitors is provided below.

- Dr. Kimble is a person of the Organizing or Technical Committee running all the runs. Pictures of Dr. Kimble will be available in the Wiki [2].

(a) Mockup Postman Uniform (b) Mockup Deli Man Uniform

Figure 2.8: The postman and deli man uniforms.

- Deli Man (the actual person will change in every run, but s/he will always wear a known uniform): images of the uniform and of some people wearing this uniform (not the people used in the test) are provided in Fig. 2.8 and Fig. 2.9.;
- Postman (the actual person may change in every run, but s/he will always wear a known uniform): images of the uniform and of some people wearing this uniform (not the people used in the test) are provided below in Fig. 2.8 and Fig. 2.9;

(a) Postman Uniform (b) Deli Man Uniform

Figure 2.9: The postman and deli man uniforms.

- Unknown person: no information is provided.

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	Deli Man	Postman	Unknown
Allow to enter	Yes	Yes	Yes	No
Allow to deliver objects	No	Yes	Yes	No
Allow access to kitchen	Yes	Yes	No	No
Allow access to bedroom	Yes	No	No	No

(a) Parcel Front (b) Parcel (c) Parcel Side

Figure 2.10: Pictures of the parcel

Pictures of the parcel that was used in the RoCKIn 2015 competition can be found in Figure 2.10. The dimensions are 9.5 inches / 24.13 cm (width) x 6.5 inches / 16.51 cm (length) x 1.75 inches / 4.45 cm (height).

5.3.4 Expected Robot Behavior or Output

Start of the test

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. Then for each visiting person, the following three phases are expected.

Phase 1: detection and recognition of the visitor.

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. The robot has to understand who is the person asking for a visit, according to the descriptions given above, including using the Ethernet camera. If the robot does not detect the ring call after three times, then the person will leave and the task will continue with the next person after a while.

Naturally, the front door will be closed. The robot can choose any way of opening the door, either using its manipulator or requesting a referee, team member or the visitor to open the door (e.g. using speech). There will be no penalties applied in any case. Fig. 2.11 shows the handle which will be mounted on the entrance door.

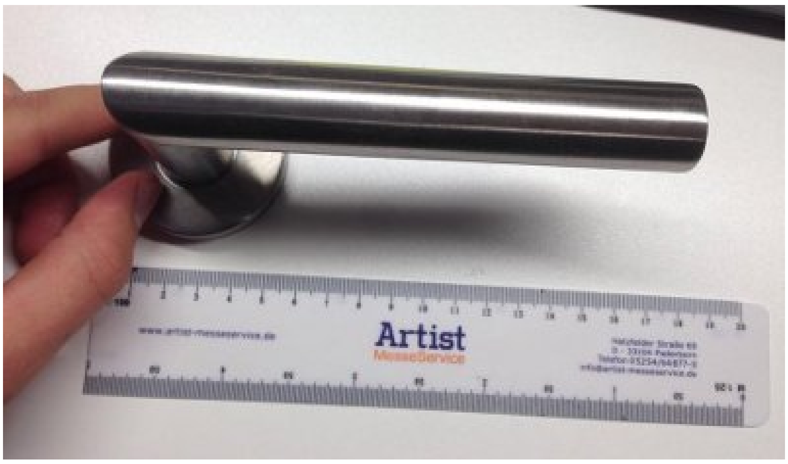


Figure 2.11: Door handle at the main entrance

Phase 2: greeting of the visitor.

For each detected visitor, the robot has greet the visitor. In this spoken sentence, the robot has to demonstrate that it understood the category of the person. For example, these sentences spoken by the robot will be considered adequate.

Visitor	Robot's Greeting message
Dr. Kimble	"Hi Dr. Kimble, I am coming to open the door."
Deli Man	"Hello, I am coming to get the breakfast."
Postman	"Hello, I am coming to get the post mail."
Unknown	"Sorry, I don't know you. I cannot open the door."

Phase 3: executing the visitor-specific behavior.

The following behaviors are expected depending on the visitor (doors are assumed to be open).

Dr. Kimble: the robot goes to the entrance door allows the Doctor to enter and 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.

Deli Man: the robot goes to the entrance door, allows the Deli Man to enter, 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 the Deli Man to exit.

PostMan: the robot goes to the entrance door, allows the Postman to enter, receives the postal mail (or ask the Postman to put it in the table in the hall), and allows the Postman to exit.

Unknown person: do nothing.

After the execution of the visitor-specific behavior, the robot can return to the initial position where it can receive the next visit.

5.3.5 Procedures and Rules

The visitors will be from all the four categories mentioned previously (including the unknown person). The following three steps will be repeated for each visitor (in a random order).

Step 1 The door-bell will be rung by the visitor, played by a member of the OC/TC. 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 uses the networked camera to identify the person in front of the door.

Step 3 The visitor can be either a known person or a completely unknown person. The robot must correctly identify the person first as known or unknown and then proceed as follows. If the person is known according to his/her face and/or uniform, the robot must proceed to do the rest of the actions as per mentioned in the task description. If the person is unknown, the robot just says goodbye to him/her.

5.3.6 Acquisition of Benchmarking Data

During the execution of the benchmark, the Internal Data defined in Section 4.3.2 will be collected together with the additional information described in the following table

Topic	Type	Frame Id	Notes
/rockin/command[16]	std_msgs/String	–	–
/rockin/visitor[17]	std_msgs/String	–	–
/rockin/audio[18]	audio_common_msgs/AudioData	–	–
/rockin/notification[19]	std_msgs/String	–	–

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

5.3.7 Scoring and Ranking

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

- The robot opens the door when the door bell is rung by Dr. Kimble and correctly identifies him.
- The robot opens the door when the door bell is rung by the Deli Man and correctly identifies him.
- The robot opens the door when the door bell is rung by the Postman and correctly identifies him.
- The robot opens the door when the door bell is rung by an unknown person and correctly identifies the person as such.
- The robot exhibits the expected behavior for interacting with Dr. Kimble.
- The robot exhibits the expected behavior for interacting with the Deli Man.
- The robot exhibits the expected behavior for interacting with the Postman.
- The robot exhibits the expected behavior for interacting with an unknown person.

Please note, the robot can either open the doors, e.g. using its manipulator, or asking for assistance, without being penalized.

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

- The robot fails in making the visitor respect the proper rights.
- The robot generates false alarms.
- The robot fails in maintaining the original state of the environment.
- The robot requires extra repetitions of speech.
- The robot bumps into the furniture.

[16] The event or command causing the activation of the robot.

[17] The result of any attempt by the robot to detect and classify a visitor

[18] The audio signals of the conversation with the visitors. 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.

[19] Any notifications from the robot (e.g., alarm if a visitor shows anomalous behavior)

- The robot stops working.

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.

5.4 Task *Catering for Granny Annie's Comfort*

This benchmark aims at assessing the robot's performance of executing requests about Granny Annie's comfort in the apartment.

5.4.1 Task Description

The robot helps Granny Annie with her daily tasks throughout the day. After waking up in the morning, Granny Annie calls the attention of the service robot by touching a button on her tablet computer. When the robot approaches her, Granny Annie uses spoken commands to ask the robot to operate on several home automated devices, for instance, lifting the shutters, switching on a light, etc. Besides operating on home automated devices, Granny can also ask the robot to further provide comfort by looking for several of her belongings and bringing them back to her. There is no specific amount of requests that Granny Annie has for the robot, and they do not follow any specific order.

5.4.2 Feature Specifications and Variation

Calling the Robot for Attention: Granny Annie has a tablet computer which runs an application provided by the consortium. On the tablet's screen, the application presents a button to call the robot (see Fig. 2.12). After pressing the button, the robot should communicate with the tablet to have it bring up a map of the apartment, so that Granny can indicate where in the house she is currently. The communication with the tablet is realized with the RSBB. Further details are specified in the ERL Wiki [2].

Language for Speech-Based Interaction: The language used for speech-based interaction between Granny Annie and robot is a set of sentences in English.

Subtasks: In the context of this task, a subtask is considered to be the resulting behaviour taken by the robot to accomplish something that Granny asked it to. In practical terms, if she asks the robot to, for instance, get her a cup, the resulting subtask is the process of looking for and bringing the cup back to her.

Order of Granny Annie's Commands: During each run of this task, the robot will be asked to perform several subtasks. Granny Annie may only give one command at a time, and only after the robot executes the corresponding subtask another one may be given.

Device Operation: Each team can choose whether the robot operates the devices with its manipulator or over the network. The networked communication follows a pre-established common protocol which is specified in the ERL Wiki [2].

Figure 2.12: The ERL-SR tablet app: using the *call button*, Granny Annie can call the robot without giving any information about her location. With an additional click on the map, Granny Annie can inform the robot about her actual location inside the apartment.

5.4.3 Input Provided

For each run of this task, in no specific order, the robot will be asked to operate on three devices and to find and bring back an object. Afterwards, the robot will be given a finalizing command, asking it to return to its idling position. In this Section, we specify a list of objects, a list of devices and a list of spoken commands that the can be given to the robot. All of the objects will be made available to the teams for calibration during the setup days.

Objects: The list of objects to be used in this task is divided into classes as described in the following table:

Class	Object
1	Cardboard Box
1	Coca-Cola Can
2	Mug
2	Candle
3	Cup
3	Reading Glasses

Images of these objects can be found in the ERL Wiki [2].

Likely Locations: Only the Cardboard Box's location is subject to be unknown in this task. There are three possible locations for it, varying in likelihood. The other objects' location is fixed.

Object	Location	Likelihood
Cardboard Box	on the kitchen counter on the kitchen table on the coffee table on the bedside table	Very High High Low Very Low
Coca-Cola Can	on the kitchen table	-
Mug	on the kitchen counter	-
Candle	on the coffee table	-
Cup	on the kitchen table	-
Reading Glasses	on the bedside table	-

Home Automated Devices: The list of home automated devices, with their operation type specified, is the following:

Class	Device	Operation Type
1	Right Bedroom Light	Switch On/Off
1	Left Bedroom Light	Switch On/Off
2	Blinds	Set Percentage
2	Light Dimmer	Set Percentage

Idling Position for the Robot: The OC/TC defines an idling position for the robot, so that the task can be terminated by asking the robot to move to it. The only rule for this position is that it must be outside of the bedroom.

Spoken Commands: There are three different types of commands that Granny Annie can give: those that refer to the operation of home actuated devices; those that refer to finding objects; and the finalizing command. They are specified as such:

Devices
Switch on/off the left light of the bedroom.
Switch on/off the right light of the bedroom.
Switch on/off both lights of the bedroom.
Open the blinds.
Close the blinds.
Leave the blinds half open.
Turn on/off the light of the living room.
Set the light of the living room to half.

Objects
Get me the + <u>object</u>.
Bring me the + <u>object</u>.
Please find my + <u>object</u>.

Finalizing
Go home.

In this list, if the sentence contains a forward slash (/), both versions of the sentence are valid. E.g., for "Switch on/off the left light of the bedroom", both "Switch on the left light of the bedroom" and "Switch off the left light of the bedroom" are valid commands. Also, <u>object</u> is a placeholder for a name of an object (the list of objects can be found in Input Specification 5.4.3). For instance, some valid sentences are: "Get me the coca-cola can", "Bring me the coca-cola can", "Bring me the mug", etc.

More elaborated examples of sentences which Granny Annie will utter during this benchmark are collected in an archive which is available in the ERL Wiki [2].

In addition to the sentences and the lexicon, ERL provides also the audio version of the examples. These files have been recorded by the same person that will play the role of Granny Annie, and can be found in the ERL Wiki [2] as well.

5.4.4 Expected Robot Behavior or Output

The robot is expected to reach the room where Granny Annie is located when she calls upon its service, approaching her in such a way that spoken communication is possible. The robot should then state its readiness to receive orders of subtasks to execute. When given a command, it should be confirmed in an appropriate way (e.g., by repeating it back to Granny and asking if it was correctly understood). The subtask corresponding to the given command should then be executed.

After executing each subtask, the robot should ask Granny what else she desires the robot to do. The last subtask that she gives the robot will always be “Go home” (i.e., return to its idling position).

5.4.5 Procedures and Rules

A successful execution of this task would be the robot going to Granny Annie (after being called from the tablet computer), executing a certain amount of subtasks and returning to its “idling” position afterward. The comfort providing task involves the following activities:

1. Granny Annie initiates the task by calling the robot’s attention.
2. The robot approaches her in such a way that spoken communication is possible, stating its readiness to receive commands.
3. Granny, using a sentence from the predefined list of possible commands, asks the robot to do something for her.
4. The robot confirms that it understood the command and executes the corresponding task.
5. After executing the task, the robot returns to Granny, asking if she has further commands for it.
6. Until Granny orders the robot to return to its idling position, it should continue to execute the tasks that she gives it.

During the execution of the task, Granny Annie (played by a member of the OC/TC), will behave as follows:

- If the robot fails to understand/perform a certain command after three tries, Granny Annie will move on to the next one.
- The teams are allowed to give instructions as to how Granny should speak with the robot. However, she has to be able to do so from the bed and maintaining at least a 30cm distance from the robot.

5.4.6 Acquisition of Benchmarking Data

During the execution of the benchmark, the Internal Data defined in Section4.3.2 will be collected together with the additional information described in the following table

Topic	Type	Frame Id	Notes
/rockin/command[20]	std_msgs/String	–	–
/rockin/audio[21]	audio_common_msgs/AudioData	–	–

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

5.4.7 Scoring and Ranking

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

- The robot enters the room where Granny Annie is waiting.
- The robot understands Annie's command(s).
- The robot operates correctly the right device(s).
- The robot finds the right object(s).
- The robot brings to Annie the right object(s).

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

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

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 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.

5.5 Task *Visit my home* (Shared with RoboCup@Home)

This task focuses on tracking and recognizing a previously unknown person, obstacle avoidance, obstacle interaction, and safe navigation in dynamic environments. In this task, the robot should visit a set of predefined waypoints while avoiding obstacles on its path, following a person outside the arena and guiding that person back to the arena.

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

[21] The audio of the conversation between Annie 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.

5.5.1 Task Description

For description of the task please refer to the *Navigation* test of the *RoboCup@Home 2016* rule book (Sections 5.2.1 to 5.2.6):
`http://www.robocupathome.org/rules/2016_rulebook.pdf`

5.5.2 Acquisition of Benchmarking Data

During this Task Benchmark, the Internal Data described in Section 4.3.2 will be collected.

5.5.3 Scoring and Ranking

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

- The robot reaches waypoint 1.
- The robot reaches waypoint 2.
- The robot reaches waypoint 3.
- The robot reaches waypoint 4 following the operator.
- The robot reaches waypoint 3 after re-entering the arena.
- The robot recognizes properly the nature of obstacles.
- The robot interacts with obstacles accordingly to their nature.

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

- The robot robot does not enter the arena autonomously.
- 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 does not exit the arena autonomously.
- 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.
- Any behavior towards attendees and/or operator that is deemed as dangerous by the referees (e.g. stopping too close to a person, or moving at inappropriately high speed while close to a person)

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

5.6 Task *General Purpose Service Robot* (Shared with RoboCup@Home)

This TBM focuses on the integration of different robot abilities such as human-robot interaction, navigation and robot-object interaction. In this test the robot has to solve multiple tasks upon request. This test is not incorporated into any predefined story and there is neither a predefined order of tasks nor a predefined set of actions. The actions that are to be carried out by the robot are randomly generated by the referees. The command is composed by three actions, which the robot has to recognize and execute.

5.6.1 Task Description

For detail description of the task please refer to the *General Purpose Service Robot* test of the *RoboCup@home* rule book (Sections 5.6 to 5.6.5):
`http://www.robocupathome.org/rules/2016_rulebook.pdf`

5.6.2 Acquisition of Benchmarking Data

During this Task Benchmark, the Internal Data described in Section 4.3.2 will be collected together with the additional information described in the following table

Topic	Type	Frame Id	Notes
/rockin/command[22]	std_msgs/String	–	–
/rockin/audio[23]	audio_common_msgs/AudioData	–	–

5.6.3 Scoring and Ranking

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

- The robot reaches the expected position.
- The robot correctly understands the command (no Continue rule).
- The robot asks one or more reasonable questions to obtain missing information.
- The robot correctly performs the first task of the command.
- The robot correctly performs the second task of the command.
- The robot correctly performs the complete command or a proper variation if not executable.
- The robot provides a reasonable description of the reason(s) why the command could not be executed as requested, and describes the way it solved the issue.

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

- The robot does not enter the arena autonomously.

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

[23]The audio of the conversation between operator 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 requires repetition of a command (each repetition brings an additional penalty).
- The team requests that a command (or a repetition of it) is provided by a team member instead of the operator.
- The robot bumps into any non-numan obstacle.
- The robot does not exit the arena autonomously.

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.

6 Functionality Benchmarks

6.1 Object Perception Functionality

6.1.1 Functionality Description

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. All objects presented to the robot in this task benchmark are commonplace items that can be found in a domestic environment. Teams are provided with a list of individual objects (instances), subdivided into classes (see Section 6.1.2). 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.

6.1.2 Feature Variation

The variation space for this FBM is represented by the set of objects, which are described in the ERL-SR Wiki [2].

6.1.3 Input Provided

The set of individual objects that will actually be presented to the robot during the execution of the functionality benchmark is a subset of a larger set of available objects, here denoted as "object instances" (see Section 3 for complete list of possible objects). This can, in general, be a superset of the objects used in the actual ERL-SR Task Benchmarks.

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.

All object instances and classes are known to the team before the benchmark, but the team does not know which object instances will actually be presented to the robot during the benchmark. More precisely, the team 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).

6.1.4 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. No parts of the robot are allowed in the space above the table interfering with the ground truth acquisition system. 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 6.1.3).

- Object localization (i.e., pose estimation): estimation of the 3D pose of the perceived object with respect to the benchmark setup reference frame.

6.1.5 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. 2.13).

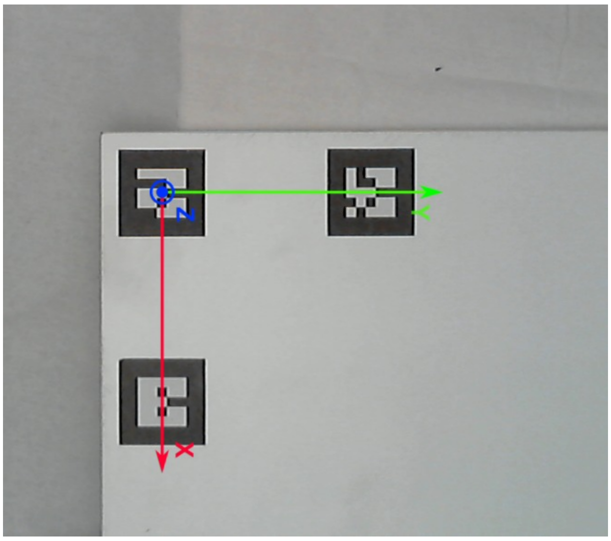


Figure 2.13: Reference frame on the FBM table.

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

6.1.6 Acquisition of Benchmarking Data

During the execution of the benchmark, the Internal Data defined in Section 4.3.2 will be collected together with the additional information described in the following table

Topic	Type	Frame Id	Notes
/rockin/notification[24]	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.

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

6.1.7 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 classified objects.
2. The number and percentage of correctly identified objects.
3. Pose error for all correctly identified objects.
4. Execution time (if less than the maximum allowed for the benchmark).

The previous criteria are in order of importance (since this functionality benchmark is primarily focused on object recognition): the first criterion is applied first and teams will be scored according to the common accuracy metrics; the ties are broken by using the second criterion, again applying accuracy metrics; finally the position error is evaluated as well. Since the position error is highly affected by the precision of the ground truth system we will use a set of classes in this as well and in case of ties we will resort to execution time.

6.2 Navigation Functionality

6.2.1 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-SR test bed (see Sec. 3). From a predefined starting position, the robot will receive a list of waypoints that it must visit to reach a goal position.

6.2.2 Feature Variation

Before each run, the robot will receive a list of waypoints that it must follow, where the final position is the last of the list. Teams will have to take into account the following changes between different runs:

- Distinct starting points, waypoints, and goal positions.
- Different number of waypoints to reach the goal.
- Different number of obstacles blocking the path.

6.2.3 Input Provided

Before each run, the robot will receive the following information (from the test bed):

- The starting position.
- The number of waypoints that the robot must visit.
- The coordinates of the waypoints that the robot must visit (the last waypoint corresponds the goal position). 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).
- The maximum time allowed to the robot to go from each waypoint to the next waypoint, without penalization.

- The deadline for the robot to go from each waypoint to the next, receiving points for that path. If robots spent more time than this deadline when going from one waypoint to the next waypoint, they will not get points on that path. More information will be provided at least eight weeks before the competition in the ERL Wiki [2].

6.2.4 Expected Robot Behavior or Output

Teams are required to set their robot on a specific starting position (that will be given to the teams before each run). Then, the robot receives, through the RSBB, the start signal, as well as an ordered list of waypoints that it must reach. The robot must follow the order in which the waypoints are sent from the test bed, sending back a signal to the RSBB, each time it reaches a waypoint. The evaluation of the navigation will take into account the following three items (a complete formula for the scoring will be presented at Sec. 6.2.7):

- The distance between the robot's position and the respective position of the waypoint. It will be accounted both the Euclidean distance between the waypoint and the robot, and the difference in the orientation.
- The time spent by the robot to go from each waypoint to the next waypoint.
- The number of times that the robot hits each obstacle. If the robot hits the same obstacle more than once, it will count as multiple hits.

The functionality benchmark ends as soon as one of the following situations occurs:

- The robot reaches the last waypoint which corresponds to the goal position.
- The time available for the functionality benchmark expires.
- The robot hard-hits an obstacle. A hard-hit is defined further in the document.

6.2.5 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. Two minutes to move the robot to the correct starting position plus eight minutes to do the benchmarking.

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 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 goal position (which corresponds to 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 3.2.1. 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. Obviously, the movement people will do is unpredictable.

6.2.6 Acquisition of Benchmarking Data

During the execution of the benchmark, the Internal Data defined in Section4.3.2 will be collected together with the additional information described in the following table

Topic	Type	Frame Id	Notes
/rockin/robot_pose_waypoint[25]	geometry_msgs/PoseStamped	/map	when reached
/rockin/marker_pose_waypoint[26]	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.

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).

6.2.7 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.

- 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

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

  where $d\left(\mathbf{p}_i, \mathbf{r}_i\right)$ is the Euclidean distance from the robot's position $\mathbf{r}_i$ to the target waypoint $\mathbf{p}_i$ (as shown in Figure 2.14) and N the total number of points.

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

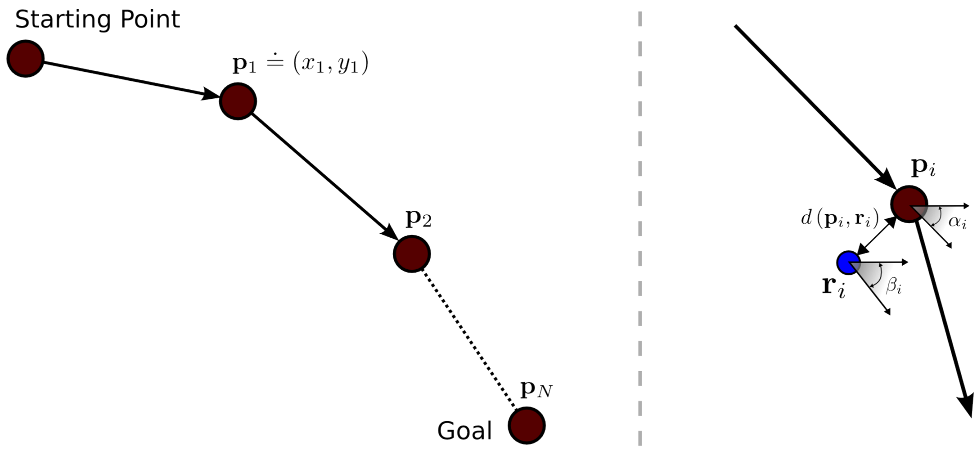


Figure 2.14: Depiction of the sequence of waypoints that the robot must follow, during the navigation functionality benchmark.

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.

6.3 Speech Understanding Functionality

6.3.1 Functionality Description

This functionality benchmark aims at evaluating the ability of a robot to understand speech commands that a user gives in a home environment. A list of commands will be selected among the set of predefined recognizable commands (i.e., commands that the robot should be able to recognize within the tasks of the competition or in similar situations).

Each implemented system should be able to capture audio from an on-board microphone, to record the captured audio in a file and to interpret the corresponding utterance. A standard format for audio files will be chosen (e.g., WAV) and communicated to the teams in advance before the competition. 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*"`)`).

Summarizing, for each interpreted command the following relevant information will be collected: an audio file, its correct transcription and the corresponding correct CFR.

6.3.2 Feature Variation

For this benchmark, the variation can affect mainly four aspects: different complexity in the syntactic structures of the spoken commands; use of complex grammatical features, as pronouns; use of synonyms for referring to objects; use of sentences where more than one action is expressed, resulting in a composed command (e.g. "take the bottle and bring it to me"). Furthermore, variation in the quality of the audio corresponding to the user utterances can be considered, as for representing more or less noisy conditions.

6.3.3 Input Provided

Some information about the lexicon (verbs and nouns of objects) used in the benchmark will be made available to the teams at least eight weeks before the competition. Moreover, a set of samples is already available among the ERL competition datasets in the ERL Wiki [2].

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 [2] 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 6.3.3) 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 [2].

6.3.4 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 user utterance 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 6.3.1 and defined in Section 6.1.3.

For each command uttered or for each audio file directly provided during the Speech Understanding Functionality Benchmark, 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 or audio file, following the format

$$\texttt{audio\_file\_name}_i\texttt{|command\_transcription}_i\texttt{|CFR}_i$$

where `audio_file_name`, `command_transcription` and `CFR` represent respectively the name, 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 file names must be consistent. Audio files corresponding to commands acquired through the microphone system of the robot must be named according to the following syntax

`fb_mic_phase_speech_audio_`i`.wav`,

representing the name of the i-th audio file of the i-th uttered command. In the case of audio files provided directly to the robotic platform without passing from the audio system, thus on a USB stick, the name in this field must be the same of the audio file analyzed.

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:

```
fb_mic_phase_speech_audio_1.wav|move to the living room|MOTION(goal:"living room")
fb_mic_phase_speech_audio_2.wav|BAD_RECOGNITION|NO_INTERPRETATION
...
```

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

NOTE 2: if both the evaluation phases are performed, both results must be written inside the same `results.txt` files. The audio file name associated to each line will define the belonging of the analysis outcome to one phase or the other.

The audio files that will be provided to the teams and the audio file produced by the teams must follow a precise audio format. They must be stereo (2 channels) `.WAV` files, encoded with a 44100 Hz sampling rate.

6.3.5 Procedures and Rules

All the teams will be evaluated on the same set of spoken sentences. These spoken sentences are divided in two groups: a first group is formed by pre-recorded audio files, a second group by voice commands uttered by a user during the benchmark. The robots will be disposed in circle, and the audio will be broadcast using a 360° speaker (or an equivalent structure of speakers) with high fidelity performance placed in the center. In this way, all the robots will receive the same audio at the same time.

All teams are required to perform this functionality benchmark according to the steps mentioned below. During the first two days of the competition, all teams are required to repeat it (2 runs on day 1 and 2 runs on day 2). On the third day, only a selected number of top teams will be allowed to perform it. Maximum time allowed for one functionality run is 2 minutes.

The benchmarking phase will last 20 minutes. All the teams that will deliver their USB stick after 20 minutes from the beginning, will receive a penalty on the final score. The benchmarking procedure is performed for all the teams in parallel.

1. Each team receives a USB stick containing the audio files randomly selected among the predefined set. This subset will be the same for each team in order to reproduce fair conditions in the evaluation. After inserting the USB stick in the computer of the robot, only one button can be pressed (either a button in a GUI or a key in the keyboard). So the starting time for processing the files will be (approximately) the same for all the teams.

2. For each audio file in the USB stick, the system should generate the corresponding interpretation in the CFR format together with the correct transcription of the corresponding utterance. The time for this processing will be restricted to an amount that will be communicated in advance by the OC (approximately not more than 2 minutes). Results must be saved according to the specifications described in Section 6.3.4.

3. After a proper communication, a member of the OC will pronounce some commands using a microphone. The audio will be instantly reproduced using a loud-speaker, conveniently positioned to be equally distant from each robot involved in the benchmark. Each command will be given after an interval of about 15 seconds of silence from the previous one. During this second part of the test, a designated member of the team will be allowed to press a button of the robot pc once for each sentence uttered by the speaker. **NOTE:** Loud-speakers will be available during the set-up days and audio tests will be performed before the benchmark, so teams can properly calibrate their audio systems.

4. For these additional sentences, the interpretation process explained above will be repeated and the results will be added to the `results.txt`, again according to the specifications in Section 6.3.4. Moreover, the corresponding audio files must be recorded in the USB stick. **NOTE:** For each missing audio file among these, the corresponding line in the output file will not be considered during the final evaluation. A referee will control the consistency of each audio file acquired during this phase, in order to check if they correspond to what has been uttered by the OC/TC member. Any file not corresponding to the uttered command

(e.g. containing white noise or only a portion of the uttered command), will be considered as not valid. **NOTE 2:** the audio files must respect the specs described in Section 6.3.4.

5. After the test is completed, only one button can be pressed to stop the processing.

6. The USB stick is removed from the robot and it should contain: new audio files, `results.txt` file with the recognition of both the already present audio files and the new ones recorded during the benchmark.

NOTE: the number of commands and the relative audio files can vary according to the operational condition of the test bed environment during the set up days.

6.3.6 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[27];
- 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.

Formats and interfaces for the transmission of internal robot data will be provided to the teams well before the Competitions. Please note that, according to the procedure described by Section 6.3.5 and to the definitions of 'offline' and 'online' used for the other benchmarks[28], all data acquisition occurs offline.

6.3.7 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 [3]. For the AcC this measures will be defined as follow:

[27] 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.

[28] 'Offline' identifies data produced by the robot that are collected by the referees when the execution of the benchmark ends; 'online' identifies data that the robot has to transmit to the test bed during the execution of the benchmark. NOTE: the online data should also be displayed by the robot on its computer screen, for redundancy purposes, in case problems with wireless communications arise.

- 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;
- 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.

A The @Home Test bed at IST - Implementation Details

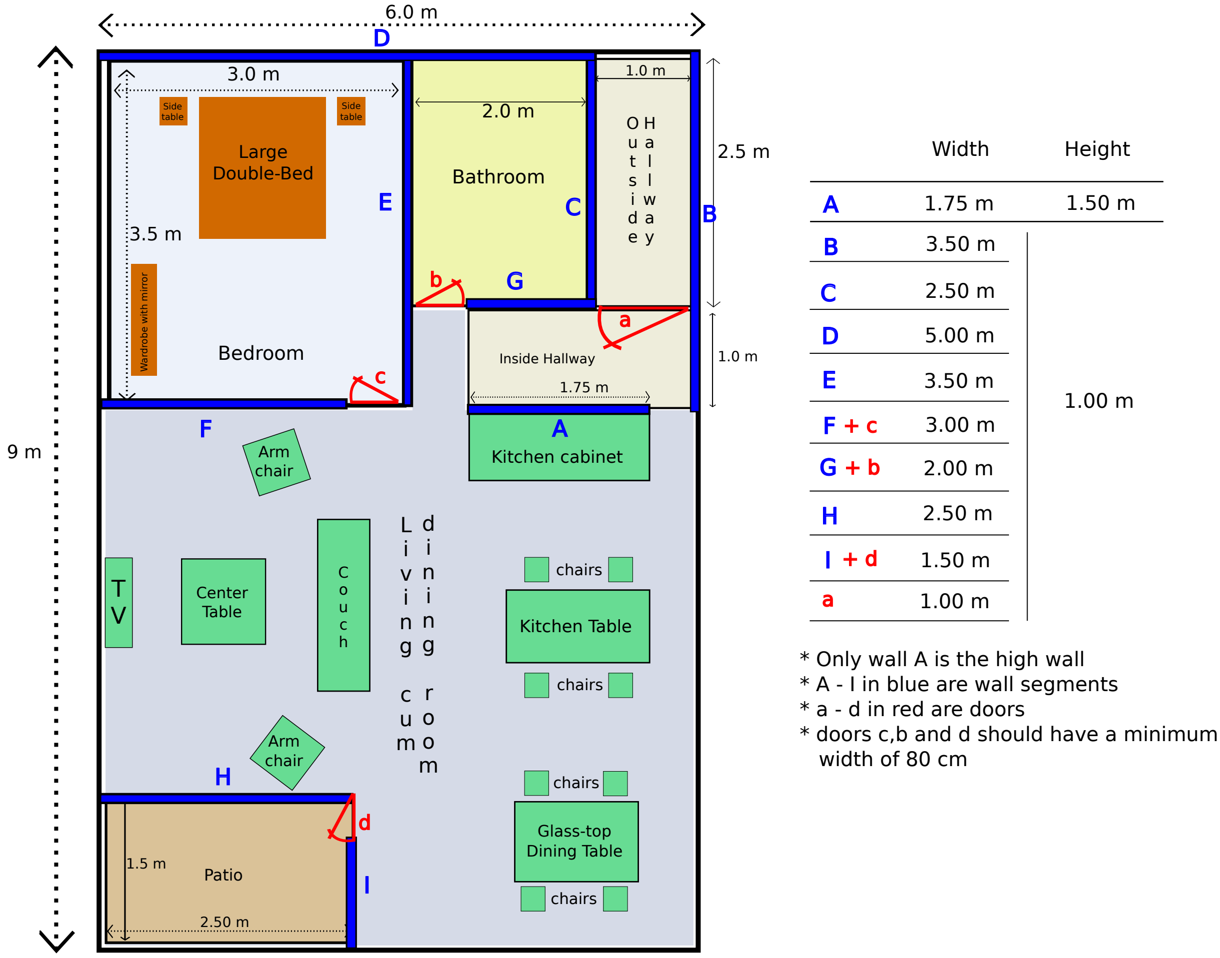


Figure 15: Layout and dimensions of the @Home test bed at ISR/IST, Lisbon.

Based on the general design and specifications of the ERL-SR test bed detailed previously in this text, in this sub-section we present the exact design specifications of the @Home test bed installed at the premises of the Institute for Systems and Robotics (ISR) of IST, Lisbon. Note that this @Home test bed is not an exact replica of the actual ERL-SR 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 16–18.

A.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 15.
- Connectivity of spatial areas: Same as depicted in Figure 2.3.
- 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.

A.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[29].

A.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 2.5 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 [30].
- 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 [31].
- 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.

[29] `www.ikea.pt`

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

[31] `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	
Dinning table - With Glass	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 Tool") are stored and changed. Technical details regarding SMARTIF products can be found at the official site [32].

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.

[32] 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 16: Living room in the @Home test bed at IST

Figure 17: Living room, kitchen and dining area in the @Home test bed at IST

Figure 18: Bedroom in the @Home test bed at IST

Bibliography

[1] 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.

[2] (2015) The RoCKIn Wiki. [Online]. Available: http://thewiki.rockinrobotchallenge.eu

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