Funded by the European Union

**Robot Competitions Kick Innovation
in Cognitive Systems and Robotics
FP7-ICT-601012**

RoCKIn@Work

– Innovation in Industrial Mobile Manipulation –

II: RoCKIn@Work Rule Book

Revision: 3.0
Date: September 18, 2014

Editors: Jakob Berghofer, Rhama Dwiputra, Gerhard Kraetzschmar
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, Viola Schiaffonati, Sven Schneider

Contents

1 Introduction to RoCKIn@Work

RoCKIn@Work is a competition that aims at bringing together the benefits of scientific benchmarking with the economic potential of innovative robot applications for industry, which call for robots capable of working interactively with humans and requiring reduced initial programming. The following ***user story*** is the basis upon which the RoCKIn@Work competition is built:

RoCKIn@Work is set in the RoCKIn'N'RoLLIn factory - a medium-sized factory that is trying to optimize its production process to meet the increasing number of unique demands from its customers. RoCKIn'N'RoLLIn specialises in the production of small to medium sized lots of mechanical parts and assembled mechatronic products. Furthermore, the RoCKIn'N'-RoLLIn production line integrates incoming shipments of damaged or unwanted products and raw materials.

Greater automation in broader application domains than today is essential for ensuring European industry remains competitive, production processes are flexible to custom demands and factories can operate safely in harsh or dangerous environments. In RoCKIn@Work, robots will assist with the assembly of a drive axle - a key component of the robot itself and therefore a step towards self-replicating robots. Tasks include locating, transporting and assembling necessary parts, checking their quality and preparing them for other machines and workers. By combining the versatility of human workers and the accuracy, reliability and robustness of mobile robot assistants, the entire production process is able to be optimised.

RoCKIn@Work is looking to make these innovative and flexible manufacturing systems, such as that required by the RoCKIn'N'RoLLIn factory, a reality. This is the inspiration behind the Challenge and the following scenario description.

A more detailed account of RoCKIn@Work, but still targeted towards a general audience, is given in the RoCKIn@Work in a Nutshell document (see [1]), which gives a brief introduction to the very idea of RoCKIn and RoCKIn@Work, 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 gives general descriptions of the task benchmarks and the functional benchmarks that make up RoCKIn@Work.

The document on hand is the rule book for RoCKIn@Work, 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 RoCKIn-@Work 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: The ***test bed*** for RoCKIn@Work competitions is described in some detail in the next section (Section 2). Subsections are devoted to the specification of the structure of the environment and its properties (2.1), to the mechanical parts and objects in the in the environment which can be manipulated (2.2.1, 2.2.2), to objects in the environment need to be recognized for completing the task (2.2.3), to the networked devices embedded in the environment and accessible to the robot (2.3), 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) (2.5). Next (Section 3), we provide some specifications and constraints applying to the ***robots and teams*** permitted to participate in RoCKIn@Work. The RoCKIn 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 4) and the ***functionality benchmarks*** (Section 5) comprising the RoCKIn@Work competition. Section 6 on ***competition structure*** provides information on how the competition is organized, in particular, in which order the benchmarks are executed, whether and how often they are repeated, and when. While information on scoring and ranking the performance of participating teams on each benchmark is already provided in the benchmark descriptions, Section 7, ***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. Last but not least, Section 8 provides details on ***organizational issues***, like the committees involved, the media to communicate with teams, qualification and setup procedures, competition schedules, and post-competition activities.

2 The RoCKIn@Work Test Bed

The test bed for RoCKIn@Work 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 RoCKIn@Work is, to the best of our knowledge, the first industry-oriented robot 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 like conveyor belts or machines.

Figure 1 illustrates an examplary RoCKIn'N'RoLLIn environment. Participating teams

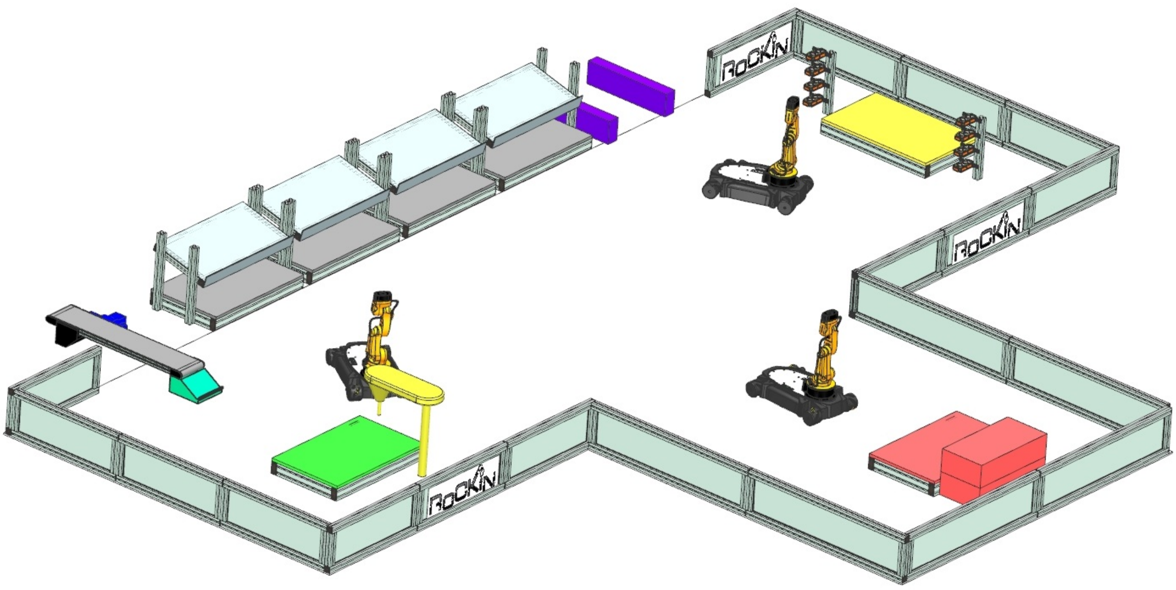


Figure 1: RoCKIn@Work environment with aid tray assembly (orange workstation), plate drilling (green workstation) and collecting for manual assembly (red workstation). The central station (purple) is for quality inspection and parking the robot.

should assume the competition environment to be as illustrated; deviations should only occur if on-site constraints (space available, safety regulations) enforce them. For detailed simulation models please see the RoCKIn website `http://rockinrobotchallenge.eu/rockin2014.php`

2.1 Environment Structure and Properties

The following set of scenario specifications must be met by the RoCKIn@Home environment.

Environment Specification 2.1 (*Structured Environment*)

The environment consists of six spatial areas: workstations #1, #4, and # 5, shelf, conveyor belt, and central station. The spatial areas extend beyond the space occupied by the respective workstations or objects and include the surrounding area as well.

Environment Specification 2.2 (*Flat Environment*)

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

Environment Specification 2.3 (*Spatial Areas and Rooms*)

The factory is a single, large open space; there are no rooms separated by walls in the environment. Spataial areas can be partially separated by dividing or protective walls or other objects present in the factory (e.g. shelves, workstations, platforms, tables).

Environment Specification 2.4 (*List of Areas*)

The environment features the following six spatial areas:

1. *a row of shelves*
2. *workstation #1 "assemble aid tray"*
3. *workstation #4 "plate drilling"*
4. *a conveyor belt*
5. *workstation #5 "fill a box"*
6. *a central station*

Environment Specification 2.5 (*Dimensions*)

The precise dimensions and the arrangement of the spatial areas are not predefined, but minimal sizes are given. The minimum sizes of the spatial areas are as follows (all dimensions in m): workstation #1 2×2*, workstation #4 including conveyor belt* 2×2*, workstation #5* 2×2*, central station* 2×3*. The bounding box of the environment has a minimum area of* $16m^2$ *and a maximum area of* $100m^2$*. More space is used, when areas and workstations are doubled for teams working parallel. See Figure 2*

Environment Specification 2.6 (*Set of Shelves*)

The shelves-area will have several different shelves where the robot can take and deliver objects, placed inside containers or boxes or directly onto shelves.

Environment Specification 2.7 (*Workstation #1*)

Workstation #1 one has a table for temporarily storing handled parts. The table is part of the force fitting machine which is operated by a robot or human worker. On each side is a rack to attach filled or unfilled aid trays.

Environment Specification 2.8 (*Workstation #4*)

Workstation #4 consists of a storing area to store "file card" boxes. The central area is a table with a customized fixture for the plates (part ID: AX-06). The plates are processed by a drilling machine, positioned above the fixture.

Environment Specification 2.9 (*Conveyor Belt*)

The conveyor belt transports parts from outside of the arena into the area close to workstation #4. At the end of the conveyor belt, parts fall down a ramp in a predefined position through guiders where they can be taken by the robot.

Environment Specification 2.10 (*Workstation #5*)

Workstation #5 consists of a table, where a human worker performs assembly of parts. The table features predefined areas where the robot can put boxes with supplies and pick up boxes *with finished parts, that have already been processed by the worker and need to be delivered elsewhere.*

Environment Specification 2.11 (*Central Station*)

The central station has several functions: It is a central station where the robot should start from in every task. It is a parking facility and charging dock.

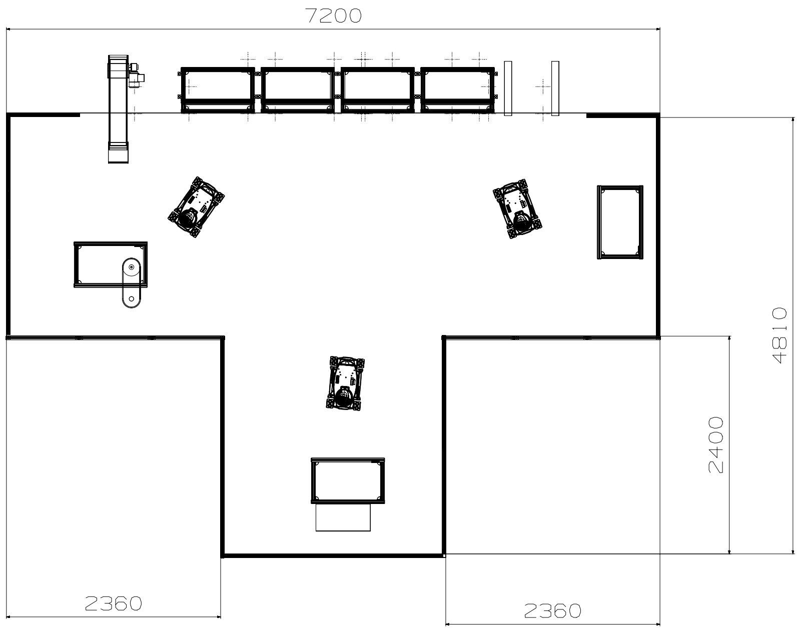


Figure 2: RoCKIn@Work environment top view with measures (in mm).

2.2 Objects in the Environment

The following lists describe the objects found in the environment. For datasheets as well as 3D models, please see RoCKIn website `http://rockinrobotchallenge.eu/rockin2014.php`. Three classes of objects are defined:

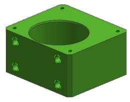
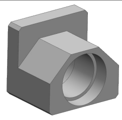
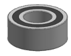
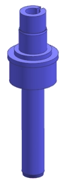
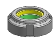
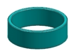
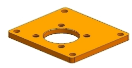
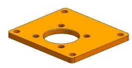
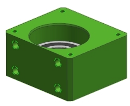
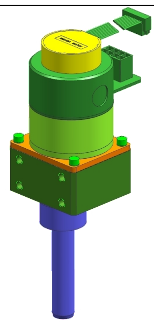
- parts of the final product, that have to be recognized and manipulated
- objects, that have to be recognized and manipulated
- objects, that have only to be recognized (because they are fixed to the environment, to heavy to lift and it is not necessary)

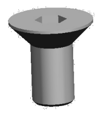
These objects are listed in tables 2.2.1, 2.2.2, 2.2.3.

2.2.1 Parts to manipulate

Table 1: List of parts for assembling the drive axle

Part ID	Part name	Picture

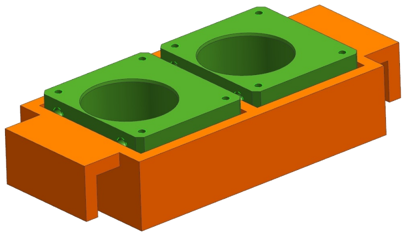
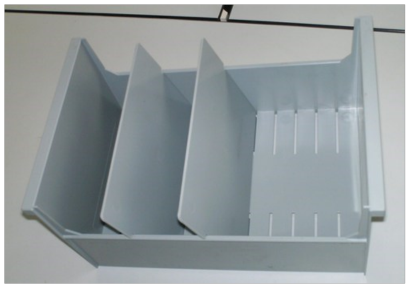
AX-01	Bearing Box	
AX-01A	Bearing Box type A	
AX-02	Bearing	
AX-03	Axis	
AX-04	Shaft nut	
AX-05	Distance tube	
AX-06	Cover Plate	
AX-07	Cover Plate Machined	
AX-08	Pre Assembled Bearing Box	
AX-09	Motor with Gearbox and Encoder	
AX-10	Screw Pack 1 (mounting cover plate to bearing box)	

AX-11	Screw Pack 2 (mounting cover plate to motor)	

2.2.2 Objects in the environment to manipulate

The abbreviation "EM" stands for *environment manipulate*.

Table 2: List of manipulated objects in the environment

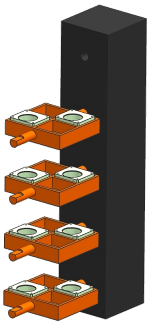
Object ID	Object name	Picture
EM-01	Aid Tray (QR code can be placed on any side)	
EM-02	"file card" box	
EM-03	box for "fill a box"	
EM-06	cone sink driller	

2.2.3 Objects to recognize

In the following list only "smaller" objects are described. For a detailed description of the environment's furniture see Section 2.1. The abbreviation "ER" stands for *environment recognize*.

Table 3: List of objects in the environment

Object ID	Object name	Picture

ER-01	foam container	
ER-02	common shelf container	
ER-03	tray rack for aid tray at force fitting workstation	

2.3 Networked Devices in the Environment

A quality control camera is installed in workstation # 4 for the task plate drilling. The quality control camera provides a RGB raw image with the resolution of 320x240 which can be triggered by the robot. The robot receives the input from the camera through wireless communication.

2.4 Central Factory Hub

The main idea of the RoCKIn@Work testbed software infrastructure is to have a central server-like hub (the RoCKIn@Work Central Factory Hub) that serves all the services that are needed for executing and scoring tasks and successfully realize the competition. This hub is derived from software systems well known in industrial business (e.g. SAP). It provides the robots with information regarding the specific tasks and tracks the production process as well as stock and logistics information of the RoCKIn'N'RoLLIn company. It is a plug-in driven software system. Each plug-in is responsible for a specific task, benchmarking or other functionality.
The following set of specifications must be met by the RoCKIn@Work Central Factory Hub(CFH).

CFH Specification 2.1 (*Central factory Hub*)

Central managing of all services needed for controlling data, devices and robots. In the following sections, plug-in will be defined. A web-based user interface would be useful.

CFH Specification 2.2 (*Time table and Scoring plug-in*)

A plug-in that is able to present actual scoring and time table of upcoming task executions by teams. This is used to keep the audience informed via a big screen or in the internet.

CFH Specification 2.3 (*Benchmarking plug-in*)

A plug-in to serve general benchmark functionalities or interact with a separated BM software.

CFH Specification 2.4 (*Production Tracking Database plug-in*)

An automatic plug-in with the central database tracking all QR Codes, status of finished products, sub-assemblies, stock level etc.

2.5 Benchmarking Equipment in the Environment

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

- **internal benchmarking data**, collected by the robot system under test (see Section 3);
- **external benchmarking data**, collected by the equipment embedded into the test bed.

External benchmarking data is generated by the RoCKIn test bed with a multitude of methods, depending on their nature.

One of the types of external benchmarking data used by RoCKIn are pose data about robots and/or their constituent parts. To acquire these, RoCKIn uses a camera-based commercial motion capture system (NaturalPoint OptiTrack), 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 the wrist). Once generated by the OptiTrack system, 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.

Pose data is especially significant because it is used for multiple benchmarks. There are other types of external benchmarking data that RoCKIn acquires: however, these are usually collected using devices that are specific to the benchmark. For this reason, such devices (such as the test panels used for the Visual Servoing functionality benchmark: see Section 5.2) are described in the context of the associated benchmark, rather than here.

Finally, 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 3.2).

3 Robots and Teams

The purpose of this section is twofold:

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.

The description of the robot should be included in the team description paper.

3.1 General Specifications and Constraints on Robots and Teams

Robot Specification 3.1 (*System*)

A competition entry may use a single robot or multiple robots acting as a team. It is not required that the robots are certified for industrial use. At least one of the robots entered by a team is capable of:

- *mobility and autonomous navigation.*
- *manipulate and grasp at least several different task-relevant objects. The specific kind of manipulation and grasping activity required is to be derived from the task specifications.*

The robot subsystems (mobility, manipulation and grasping) should work with the environment and objects specified in the RoCKIn@Work rulebook.

Robot Specification 3.2 (*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 the sensor system in the environment provided by the organizer by using a wireless communication protocol specified for such purpose. Sensor systems used for benchmarking and any other systems intended for exclusive use of the organisers are not accessible by the robot system. Teams are not allowed to modify the environment or to install their own embedded devices in the environment, e.g., additional sensors or actuators.*

Robot Specification 3.3 (*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 3.4 (*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 3.1 (*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 3.2 (*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 3.3 (*Operation*)

In the competition, the robot should perform the tasks autonomously. An external device is allowed for additional computational power. It must be clear at all times that no manual or remote control is exerted to influence the behavior of the robots during the execution of tasks.

Robot Constraint 3.4 (*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.

3.2 Benchmarking Equipment in the Robots

Preliminary Remark: Whenever teams are required to install some element provided by RoCKIn 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.

Hardware As a general rule, RoCKIn does not require that teams install additional robotic hardware on their robots. Moreover, permanent change to the robot's hardware is never required. However, RoCKIn may require that additional standard PC hardware (such as an external, USB-connected hard disk for logging) is temporarily added to the robot in order to collect internal benchmarking data. When this is the case, the additional hardware is provided by RoCKIn during the Competition, and its configuration for use is either automatically performed by the operating system, or very simple.

To allow the acquisition of external benchmarking data about their pose, robots need to be fitted with special reflective markers, mounted in known positions. The teams will be required to prepare their robots so to ease the mounting of the markers. Teams will also be required to provide the geometric transformation from the position the marker to the odometric center of the robot[1].

Software RoCKIn may require that robots run RoCKIn-provided (or publicly available) software during benchmarks. A typical example of such software is a package that logs data provided by the robot, or a client that interfaces with a RoCKIn server via the wireless network of the test bed. Whenever a team is required to install and run such a package, it will be provided

[1] Benchmarking data related to poses will refer to the marker position: this is why additional information is required to know the position of the base.

as source code, its usage will be most simple, and complete instruction for installation and use will be provided along with it. All RoCKIn software is written to have a minimal impact on the performance of a robot, both in terms of required processing power and in terms of (lack of) interaction with other modules. When required by a benchmark, the relevant RoCKIn software to be run by participating robots is provided well in advance with respect to the Competition.

RoCKIn will make any effort to avoid imposing constraints on the teams participating to the Competition in terms of software architecture of their robots. This means that any provided piece of software will be designed to have the widest generality of application. However, this does not mean that the difficulty of incorporating such software into the software architecture of a robot will be independent from such architecture: for technical reasons, differences may emerge. A significant example is that of software for data logging. At the moment, it appears likely that any such software by RoCKIn will be based on the established *rosbag* software tool, library and file format. As rosbag is part of ROS (Robot Operating System), robots based on ROS can use it to log data without any modification; on the contrary, robots not using ROS will be required to employ the rosbag library to create rosbag files (*bagfiles*) or to develop ad-hoc code to convert their well established logging format into the rosbag one by using the rosbag API. If this will be the case, RoCKIn will provide tools to ease the introduction of software modules for creation of bagfiles into any software architecture; yet, teams not using ROS will probably have to perform some additional work to use such tools.

3.3 Robot Communication with Benchmarking Equipment

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 testbed, 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 RoCKIn Competition are required to:

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

[2]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 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) 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 can ask the team to stop their robot at any time which must be done immediately.
2. A member of the OC 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 3.

All teams are required to perform each task according to the steps mentioned in the rules and procedures sub-subsections for the tasks. During the first two days of the competition, all teams are required to repeat the task (twice on day 1 and twice on day 2). On the third day, only a selected number of top teams will be allowed to perform the task. Maximum time allowed for one task run is 10 minutes.

Scoring and Ranking Evaluation of the performance of a robot according to this task benchmark is based on performance equivalence classes and they are related to the fact that the robot has done the required task or not.

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 penalizations is considered as better
- If the two robots received the same amount of penalizations, 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).

4.1 Task *Prepare Assembly Aid Tray for Force Fitting*

This task serves as an example for collecting and assembling parts from different locations. Additionally, the teams can show off their robot's capability in loading and unloading machines (a well known industrial task).

4.1.1 Task Description

The robot's task is to collect bearing boxes from stock (shelves) and insert them into specialized aid trays. Once the assembly aid tray is filled with the bearing boxes, these aid trays is loaded to a force fitting machine, where bearings are force fitted into bearing boxes. After the bearing boxes of the assembly aid tray are force fitted, the robot needs to do a final examination before delivering the final product. By scanning QR codes as part of the task, the robot ensures tracking of the production process and the parts belonging to a particular product itself.

4.1.2 Feature Variation

The bearing boxes can occur in different shapes (see list of parts in table 2.2.1). This is caused by a modular concept of the final product where the bearing box has to be inserted in alternating chassis. The robots are allowed to collect and inserted the bearing boxes to the assembly aid tray individually or collectively.

4.1.3 Input Provided

The team will be provided with the following information:

- description of the set of possible assembly aid tray and bearing boxes.
- description and location(s) of the container(s) used for the bearing boxes.

During the execution of the task, the robot should perform the task autonomously and without any additional input.

4.1.4 Expected Robot Behavior or Output

The robot goes to the central station and register itself to the central factory hub. After receiving the task of Assembly Aid Tray for Force Fitting, the robot locates the assembly aid tray in *shelf*. The robot proceeds with identifying the QR code on the assembly aid tray. The QR code has the information regarding the assembly aid tray's serial number and the type of the bearing box which can be fitted. Based on the examination of the assembly aid tray, the robot needs to find the correct bearing boxes in the set of shelves area. After finding the right bearing boxes, the robot records the QR codes of their containers, collects the bearing boxes and places them into the assembly aid tray. The robot has the option to deliver the bearing boxes collectively or individually. After placing the bearing boxes in the assembly aid tray, the robot delivers the assembly aid tray to the force fitting workstation. In the force fitting workstation, the assembly aid tray will be processed and the robot will be informed when the process is completed. The robot will check the final product and can request for another force fitting process when the result is unsatisfactory. At the end of the task, the robot will provide a report with the following information:

- The QR code of the assembly aid tray.
- The QR code of the container.
- The number of the bearing boxes successfully inserted into the assembly aid tray.
- The chronological time stamp of the following actions:
  - Start of task execution.
  - Detection of the assembly aid tray .
  - Detection of the QR code of the assembly aid tray.
  - Detection of the QR code of the containers of the bearing boxes.
  - Force fitting process.
  - Completion of the task.

4.1.5 Procedures and Rules

During the execution of this task, which needs to be carried out as per the next three steps, an additional robot might be randomly moving in the arena which has to be avoided by the participating robot.

Step 1 The robot is provided with multiple assembly aid trays and the information regarding the storage area of the bearing boxes.

Step 2 Based on the QR codes provided beforehand to the teams, the robots must identify the appropriate bearing boxes needed to be put on this tray.

Step 3 The robot must pick (from the storage area) and insert the bearing boxes, identified in the Step 2 above, in the provided assembly tray.

Step 4 The robot must deliver the assembly aid tray (with the bearing boxes) to the force fitting workstation to be processed.

Step 5 The robot needs to examine the result of the force fitting process and decide whether the process is successful. Assembly aid trays with bearing boxes which are successfully force fitted are placed in the designated assembly aid tray rack while others are returned to the force fitting machine.

4.1.6 Acquisition of Benchmarking Data

During the execution of the benchmark, the following data will be collected[3]

- ID of the assembly aid tray, provided by the robot (by analyzing the QR code); [online]
- ID of the container, provided by the robot (by analyzing the QR code); [online]

[3]In the following, 'offline' identifies data produced by the robot that will be collected by the referees when the execution of the benchmark ends (e.g., as files on a USB stick), while 'online' identifies data that the robot has to transmit to the testbed during the execution of the benchmark. Data marked neither with 'offline' nor with 'online' is generated outside the robot. 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.

- images used by the robot to analyze the QR codes; [offline]
- number of the bearing boxes successfully fitted to the assembly aid tray;
- trajectory planned by the robot and their execution (as perceived by the robot) including replanning; [offline]

Formats and interfaces for the transmission of internal robot data will be provided to the teams well in advance of the Competitions.

4.1.7 Scoring and Ranking

Evaluation of the performance of a robot according to this task benchmark is based on performance equivalence classes. Classes are defined in dependence to:

1. The fact that the robot correctly identifies the assembly aid tray or not;
2. The fact that the robot correctly identifies the container or not;
3. The number of bearing boxes successfully inserted by the robot into the aid tray;
4. The successful execution of the force fitting procedure.

The set A of achievements for this task includes:

- The robot correctly identifies the assembly aid tray and the corresponding QR code.
- The robot correctly identifies the container and the corresponding QR code.
- The robot inserts the right objects into the aid tray.
- The robot correctly delivers the tray to the force fitting station.
- The robot correctly executes the force fitting operation.

The set PB of penalized behaviors for this task includes:

- The robot bumps into obstacles in the test bed.
- The robot drops an object.
- The robot stops working.

The set DB of disqualifying events for this task are:

- The robot damages or destroys the objects requested to manipulate.
- The robot damages the test bed.

4.2 Task *Plate Drilling*

This task simulates handling of an incomplete or faulty delivery from an external component supplier. The factory has to quickly react on such issues and create a process to correct the faulty parts.

4.2.1 Task Description

The cover plate of the bearing box has eight holes for connecting the motor with the bearing box. The four central holes need to have a cone sink. Unfortunately the supplier forgot to drill these sinks. The robot has to identify faulty parts and sort them out in file card boxes and supply these parts to a workstation that has an integrated drilling machine. The robot should perform the task with a camera for quality control which can be triggered by the robot to collect raw image data.

4.2.2 Feature Variation

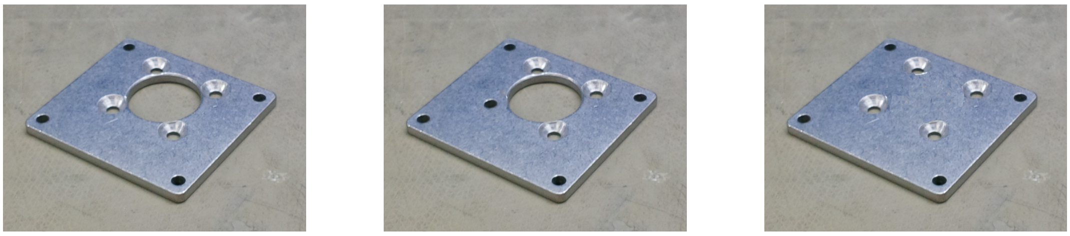
The faulty plates can have a variety of defects enumerated as follows.

1. The sequence of faulty, unusable and perfect plate flowing through the conveyer belt.
2. The cover plate orientation on the conveyor belt.
3. The number of plates delivered in each category (faulty, unusable and perfect).

4.2.3 Input Provided

The team will be provided with the following information:

- 3D CAD textured model of the plate
- description of three different states of the plate (faulty, unusable, perfect). The three different state of the cover plate are shown in Figure 3. Perfect cover plates are those with



(a) Perfect (b) Faulty (c) Unusable

Figure 3: Three possible states of the cover plate.

  four cone sink whereas faulty cover plates have one missing cone sink. The unusable cover plates are those without the hole in the middle.
- description of the file card box.

4.2.4 Expected Robot Behavior or Output

The robot starts with going to the conveyor belt area where the cover plates are being delivered. The robot has control over the conveyer belt (start and stop) which allows the robot to regulate the flow of the incoming cover plates. After collecting the cover plate, the robot proceeds with placing the cover plate in front of the quality control camera in the environment and collects the image data of the plate. The teams are allowed to use the camera available in the robot too. Placing additional networked camera in the arena is not allowed. The teams must have their own software for examining the cover plate. Based on the examination, the robot needs to determine whether the cover plate is faulty, unusable or perfect. Subsequently, the cover plate needs to be delivered to workstation #4 and placed in the card boxes accordingly (faulty, perfect and unusable). At the end of the task and during intermediate steps (see sub-subsection4.2.5), the robot must provide a report with the number of cover plates examined in each category (faulty, unusable and perfect).

4.2.5 Procedures and Rules

Step 1 The robot must control the conveyor belt so that it is able to observe and collect the cover plates.

Step 2 The robot should pick, examine (using the robot's camera or using the external camera), sort and put the cover plate in the appropriate card box.

Step 3 After processing all the cover plates, the robot should transport the card box filled with faulty plates to the drilling rig and provide a report with the number of cover plates examined in each category (faulty, unusable and perfect).

Step 4 Once at the drilling rig, the robot must place perform the correct drilling on each faulty plate.

Step 5 After drilling each faulty plate the robot must re-examine the plate using the external camera, sort and keep it in the appropriate card box.

Step 6 Steps 4 and 5 should be performed for each faulty plate found in Step 2 and a report must be provided as mentioned in Step 3 after each time Steps 4 and 5 are repeated.

4.2.6 Acquisition of Benchmarking Data

During the execution of the benchmark, the following data will be collected[4]

- Number and condition (unusable, faulty, perfect) of all plates provided to the robot via the conveyor belt.
- Trajectory planned by the robot and its execution (as perceived by the robot), including replanning. [offline]
- Condition of each plate, as evaluated by the robot, after receiving the plate; [online]
- Drilling commands issued by the robot; [online]
- Condition of each plate, as evaluated by the robot, after drilling; [online]
- Effect of the robot's sorting actions;
- Sensor data used by the robot to perform plate analysis (images). [offline]

Formats and interfaces for the transmission of internal robot data will be provided to the teams well in advance of the Competitions.

4.2.7 Scoring and Ranking

Evaluation of the performance of a robot according to this task benchmark is based on performance equivalence classes. Classes are defined in dependence to:

1. The number and percentage of correctly classified plates.
2. The number and percentage of correctly sorted plates (before and after drilling).
3. The number and percentage of correctly machined plates.

The *set A* achievements for this task are

- The robot correctly identifies perfect plates and sorts them in the corresponding card file box.

[4] In the following, 'offline' identifies data produced by the robot that will be collected by the referees when the execution of the benchmark ends (e.g., as files on a USB stick), while 'online' identifies data that the robot has to transmit to the testbed during the execution of the benchmark. Data marked neither with 'offline' nor with 'online' is generated outside the robot. 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.

- The robot correctly identifies unusable plates and sorts them in the corresponding card file box.
- The robot correctly identifies faulty plates, machines them, and sorts them in the corresponding card file box.

The set PB of penalized behaviors for this task include:

- The robot bumps into obstacles in the test bed.
- The robot drops a plate.
- The robot misses a report.
- The robot stops working.

The *set DB* of disqualifying events for this task are

- The robot damages or destroys the objects requested to manipulate.
- The robot damages the test bed.

4.3 Task *Fill a Box with Parts for Manual Assembly*

This task reflects one of the primary requisites of a mobile robotic service assistant, i.e., to work together with humans. In this case the gaol is to assist humans at a manual assembly workstation.

4.3.1 Task Description

The robot must compose boxes with parts for the manual, final assembly of the drive axle. The boxes have no special subdivisions; they only have foam material at the bottom to guarantee safe transport. Therefore, the robot has to plan the order of collecting the parts to arrange them next to each other.

4.3.2 Feature Variation

The standardized boxes can be used for several groups of parts. Because of variations in containing parts (e.g., bearing box variations) the groups of parts in this task vary the same way.

4.3.3 Input Provided

The team will be provided with the following information:

- The list of possible parts used in the task.
- Description of the box used for collecting the parts.
- Location of the parts in the *set of shelves* area.

4.3.4 Expected Robot Behavior or Output

The task execution is triggered by the robot receiving the list of parts required for the assembly process. The robot proceeds with collecting an empty box in the *set of shelves* area and begin collecting the parts (individually or collectively). When the parts are placed in the box, the robot needs to deliver the box to the *workstation #5* and provide the human worker with the list of parts in the box and, (if exist any), the missing parts.

4.3.5 Procedures and Rules

There can be multiple obstacles or other robots present in the scene that might block the direct path of the competing robot. The robot must avoid all the obstacles/other robots during the execution of the next steps in this task.

Step 1 The robot will receive, through a computer console, a list of products that need to be assembled. Also, for each product on this list there will be an associated priority.

Step 2 The robot must plan the best path to the designated delivery area, passing through each storage area where the required parts for a requested product in the list can be found.

Step 3 The robot must execute the above path, collect the parts and then deliver it to the designated area for assembly of that product.

Step 4 The Steps 2 and 3 above must be done for all the products in the list mentioned in Step 1. Also, the robot must follow, as much as possible, the priorities as per the philosophy of *first-in first-out* of the products when executing Step 2 and 3.

4.3.6 Acquisition of Benchmarking Data

During the execution of the benchmark, the following data will be collected[5]

- Initial plan, as defined by the robot after receiving the information about the product to be assembled; [offline]
- Trajectory planned by the robot, including additional ones due to replanning; [offline]
- Any new plan defined by the robot during part collection (e.g., to manage obstructions); [offline]
- Ground truth pose of the base of the robot while it is collecting parts;
- Pose of the base of the robot as estimated by the robot, while it is collecting parts; [offline]
- Number and identity of the parts provided by the robot to the human worker at the end of the collection.

Formats and interfaces for the transmission of internal robot data will be provided to the teams well in advance of the Competitions.

4.3.7 Scoring and Ranking

Evaluation of the performance of a robot according to this task benchmark is based on performance equivalence classes. Classes are defined in dependence to the number of parts of the product to be assembled actually provided by the robot to the human worker and their order according to the desired one.
The set A of achievements for this task consists of:

- The robot correctly identifies perfect required parts and bring them to destination provided all the parts required earlier are already there[6]

[5]In the following, 'offline' identifies data produced by the robot that will be collected by the referees when the execution of the benchmark ends (e.g., as files on a USB stick), while 'online' identifies data that the robot has to transmit to the testbed during the execution of the benchmark. Data marked neither with 'offline' nor with 'online' is generated outside the robot. 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.

[6]In practice lets number the parts by their login order, expressed in the list, if the robot brings the second in the list without having delivered the first, it does not achieve anything, but if the robot brings the first part right after the second, then both parts are at the same place and the human can start the assembly. This makes the robot achieving the fact of having 2 parts at destination.

The set PB of penalized behaviors for this task include:

- The robot bumps into obstacles in the test bed.
- The robot drops an object.
- The robot stops working.

The set DB of disqualifying behaviors for this task includes:

- The robot damages or destroys the objects requested to manipulate.
- The robot damages the test bed.

5 Functionality Benchmarks

5.1 Object Perception Functionality

5.1.1 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. Objects presented to the robot in this task benchmark are chosen to be representative of the type of factory scenario that RoCKIn@Work is based on. Teams are provided with a list of of individual objects (object instances), subdivided into classes (see Section 5.1.3). The benchmark requires that the robot, when presented with objects from such list, detects their presence, estimate their class, identity and location. As for example when presented with a segment of T-section metal profile the robot should detect it is in front of a profile (class) with a T-shaped section (instance) and that it is at a given position with respect to a known reference frame (this will be the benchmark setup reference frame). Objects that are used here are described in Section 2.

5.1.2 Feature Variation

For this benchmark, the variation space for object features is represented by the (known) set of objects that the robot may be presented with. Variation space for object location is the surface of the benchmarking area where objects are located (see Section 5.1.3).

5.1.3 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 ("object instances"). Object instances are subdivided into classes of objects that have one or more properties in common ("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/models of all the object instances in the form of 3D textured models;
- Subdivision of the object instances into object classes (for instance: profiles, screws, joints);
- Reference systems associated to the table surface and to each object instance (to be used to express object poses).

Object descriptions will be expressed according to widely accepted representations, well in advance of the Competition.

5.1.4 Expected Robot Behavior or Output

The objects that the robot is required to perceive are positioned, one at the time, on a table located directly in front of the robot. The robot does not move during the benchmark, and the team is allowed to choose its location wrt the table (e.g., wrt the table reference system). However, no parts of the robot are allowed in the space above the table.

The actual pose of the objects presented to the robot is unknown before they are set on the table. The robot, for each object presented to it, must perform all of the following:

- 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 5.1.3).
- Object recognition: association between the perceived object and one of the object instances belonging to the selected class (see 5.1.3).
- Object localization: estimation of the 3D pose of the perceived object wrt to the surface of the table.

5.1.5 Procedures and Rules

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

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 (10 times in a row on day 1 and 10 times in a row 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. Note that all objects within this functionality benchmark are from within the RoCKIn@Work environment.

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

Step 2 The robot must determine the object's class, its instance within that class as well as the 3D pose of the object and save it in the given format (see Section 5.1.6).

5.1.6 Acquisition of Benchmarking Data

During the execution of the benchmark, the following data will be collected[7]:

- Number of objects presented to the robot;
- Detection, recognition and localization data associated to the objects, provided by the robot; [online]
- Ground truth for object pose, object class, and object instance;
- Sensor data used by the robot to perform classification (e.g., images, point clouds, etc.). [offline]

This list will be kept open so as to receive further suggestions from stakeholders. Formats and interfaces for the transmission of internal robot data will be provided to the teams well in advance of the Competitions.

[7]In the following, 'offline' identifies data produced by the robot that will be collected by the referees when the execution of the benchmark ends (e.g., as files on a USB stick), while 'online' identifies data that the robot has to transmit to the testbed during the execution of the benchmark. Data marked neither with 'offline' nor with 'online' is generated outside the robot. 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.

5.1.7 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 errors for correctly identified objects as measured by the ground truth system;
4. Execution time (if less than the maximum allowed for the benchmark).

Being this Functionality benchmark focused on object recognition, the previous criteria are in order of importance; the first criterion is applied first and teams will be scored according to the common F-measure metrics, the ties are broken by using the second one still applying F-measure 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 "distance classes" and in case of ties we will resort to execution time.

5.2 Visual Servoing Functionality

5.2.1 Functionality Description

This functionality benchmark assesses the robot capability in controlling the manipulator motion based on its own visual perception. An object from a known set of possible objects is presented in the test for the robot to identify it Afterwards, the robot will position its end effector with the position of the perceived object as the reference point (e.g., on top of the objects). Finally, the robot performs the grasping motion, lift the object and notify that the object is acquired.

5.2.2 Feature Variation

The objects used in the benchmark will be selected from the list of parts to manipulate as presented in the Section 2.2. Additionally, the precise position of the object differs in each test.

5.2.3 Input Provided

The team will be provided with the following information:

- The list of possible objects used in the functionality benchmark.

5.2.4 Expected Robot Behaviour or Output

The robot is placed in front of the test area (a planar surface). Different objects will be placed in the test area. For each object, the robot will identify the object and move its end effector on top of it. Afterwards, the robot performs the grasping motion and notifies that the grasping has occurred. The task is repeated with different objects.

5.2.5 Procedures and Rules

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

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 (5 times in a row on day 1 and 5 times in a row 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 4 minutes. Note that all objects within this functionality benchmark are from within the RoCKIn@Work environment.

Step 1 Any object from a larger set of known objects will be placed in front of the robot.

Step 2 The robot must determine which object is presented and provide this information according to the the given format (see Section 5.2.6).

Step 3 The robot must position the end effector with respect to the object in order to grasp it.

Step 4 The robot must grasp the object, lift it, and notify that grasping has occurred.

Step 5 The robot must keep the grip on the object for a given time, then set the object down, release the object and move the end effector away from it.

5.2.6 Acquisition of Benchmarking Data

During the execution of the benchmark, the following data will be collected[8]:

- Number of objects presented to the robot;
- Identification of the objects, provided by the robot; [online]
- Grasp notifications issued by the robot; [online]
- Ground truth concerning the fact that the object does not touch the table;
- Sensor data used by the robot to perform identification (images). [offline]

This list will be kept open so as to receive further suggestions from stakeholders. Formats and interfaces for the storing and transmission of internal robot data will be provided to the teams well in advance of the Competitions.

5.2.7 Scoring and Ranking

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

1. Number and percentage of correctly identified objects;
2. Number and percentage of correctly grasped objects (the object stops touching the table, see definition below);
3. Execution time (if less than the maximum allowed for the benchmark).

Being this Functionality benchmark focused on Visual Servoing, the scoring of teams is based on the number of objects correctly grasped. A correct grasp is defined as the object being lifted from the table so to be possible for the judge to pass a hand below it. For a grasping to be "correct" the position has to be kept for at least 5 seconds from the time the judge has passes the hand below the object. The time the judge will require to verify the lifting of the object might be up to 10 seconds. In case of ties the overall execution time will be taken into account.

[8] In the following, 'offline' identifies data produced by the robot that will be collected by the referees when the execution of the benchmark ends (e.g., as files on a USB stick), while 'online' identifies data that the robot has to transmit to the testbed during the execution of the benchmark. Data marked neither with 'offline' nor with 'online' is generated outside the robot. 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.

5.3 Planning and Scheduling Functionality

5.3.1 Functionality Description

This functionality benchmark has the objective to test planning and scheduling capabilities of teams. It is important to separate this functionality from almost every other influence. Therefore, the teams will be presented with several robot platforms containing control software from the organizer. The platforms will have no robot manipulators and will only simulate grasping and collection of objects. Thereby it is possible to exclude possible time delays, which are not caused by the team, but e.g., by grasping failures of real objects. The platforms movements are then to be scheduled by the team's software and shall drive through several spots in the environment to simulate collecting or delivering of objects. Collecting is simulated by stopping for a while (can vary from one up to ten seconds) at the particular location (the duration of the stop will be consistent for all the teams).

5.3.2 Feature Variation

The number of available platforms can vary from one up to three platforms. The number of demanded sets of objects can vary from one up to 4 sets (different and more of one kind).

5.3.3 Input Provided

The team is provided with several virtual lists of objects: e.g. A (A1, A2, A3, A4), B (B1, B2, B3, B4, B5, B6, B7), C (C1, C2, C3, C4, C5) as well as the location(s) of each object in the test bed. One object can possibly have multiple storage locations. The start positions of platforms is also known to the team. The platform control software includes a prompt which displays the objects already collected (see Figure 4). Workstation # 5 serves as a dropping location.

A	A1	A2	A3	A4			
B	B1	B2	B3	B4	B5	B6	B7
C	C1	C2	C3	C4	C5		

Figure 4: Scheme of display mounted on each platform to visualize status of collected virtual objects. Sets of objects (A,B,C), objects (1,2,3,...) and already "collected" objects marked orange.

Additionally, the rules and conditions that apply for several objects are provided:

- Objects within each list are ordered, i.e. they should be collected by the robot in order (e.g., B1 needs to be collected before B2)
- Some objects have multiple storage locations and multiple objects can be stored at one location (e.g., a shelf containing several containers with different objects). Therefore, in case of a blocked location, the robot needs to decide whether to wait or proceed with the

same object in another location more far away (or collect the next object from a different set and return again).

- Sets of objects can have different priorities (e.g., if set A has a higher priority than set B it has to be delivered, entirely, before set B).

5.3.4 Expected Robot Behaviour or Output

Expected behavior here relies to organizers robot platforms. The robots start from the central station and are commanded through the planning and scheduling software of the team. The platforms should drive by and by to the demanded spots in the arena. The platforms should also go to workstation # 5 as planned by the team's software to "empty" their load. During the execution of the task the control software from the team will monitor the execution of the benchmark and possibly issue new plans according to the current situation. When the test is finished the platforms should stop at the central station again.

5.3.5 Procedures and Rules

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

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 times in a row on day 1 and 2 times in a row 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 10 minutes.

Step 1 The central scheduler defines several sets of objects, that have to be collected.

Step 2 The teams software pre-plans the scheduling of platforms.

Step 3 The team's software sends commands to platforms, receives feedback and tentatively re-plans, if apparently platforms are blocking each other or priority conditions are violated.

Step 4 The team's software receives feedback from platforms, when they are ready with collecting and can drop their load at workstation # 5. The platforms return status of display (means objects already collected)

5.3.6 Acquisition of Benchmarking Data

During the execution of the benchmark, the following data will be collected[9]:

- Original plan generated by the scheduler and each new plan generated; [offline]
- The current situation in terms of collected items and blocked robots;
- Time at which each object of each list has been picked up;
- Time at which each object of each list has been delivered to the destination;
- Time at which the last item of each list has been delivered to the destination.

This list will be kept open so as to receive further suggestions from stakeholders. Formats and interfaces for the transmission of internal robot data will be provided to the teams well in advance of the Competitions.

[9] In the following, 'offline' identifies data produced by the robot that will be collected by the referees when the execution of the benchmark ends (e.g., as files on a USB stick), while 'online' identifies data that the robot has to transmit to the testbed during the execution of the benchmark. Data marked neither with 'offline' nor with 'online' is generated outside the robot. 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.

5.3.7 Scoring and Ranking

Scoring is based on the following criteria:

- Number of completed lists of objects[10];
- Order in which the lists have been completed;
- Order in which the objects of each list have been delivered;
- Time for the completion of all lists, if less than the maximum allowed for the benchmark.

Correct ordering, according to priority, between completed lists is considered as essential, and items of lists completed out of priority order are ignored. For instance, if list B is completed before higher-priority list A, all the items of B are considered as not delivered and do not contribute to the score.

The score of a team is given by the following elements:

1. Number of lists completed. A higher number of completed lists is preferred.
2. Ordering in which elements of each completed list have been delivered. Correct ordering of the items is preferred. For each list, scoring follows the Levenshtein distance between the required list of items (e.g., [$A1$ $A2$ $A3$ $A4$ $A5$]) and the list of items actually delivered (e.g., [$A1$ $A3$ $A2$ $A5$]).[11]

The scoring elements above are prioritized: whatever the ordering of elements in completed lists, a robot which has completed a higher number of lists always gets a higher score.

Between teams that obtain the same score according to the preceding scoring elements and completed all the lists, ranking is defined according to the time of completion of the last-completed list. Between teams that obtain the same score according to the preceding scoring elements but did not complete all the lists, ranking is defined by the ordering of the elements of incompletely delivered lists, evaluated using the Levenshtein distance.

[10] A list is *completed* when the last item of it is delivered to the destination.

[11] This distance takes into account also out of order execution and missing items (the last property is only useful for incomplete lists, which will be considered later).

6 Competition Structure

6.1 Competition Elements

RoCKIn competitions are scientific competitions where the rules are designed in such a way that the rankings also take the role of measurements of the performance of participants, according to objective criteria. This is called in RoCKIn jargon, a *benchmarking competition*.

The elements composing a *benchmarking competition* were defined in RoCKIn Deliverable D1.1 ("Specification of General Features of Scenarios and Robots for Benchmarking Through Competitions"). We recover the most relevance here for the rulebook document to be self-contained.

Definition 6.1 (*Functionality*)

One of the basic abilities that a robot system *is required to possess in order to be subjected to a given* experiment.

The list of *functionalities* for RoCKIn@Work is defined in Section 5.

Definition 6.2 (*Functional Module*)

The (hardware and/or software) components of a robot system *that are involved in providing it with a specific* functionality.

Definition 6.3 (*Task*)

An operation or set of operations that a robot system *is required to perform, with a given (set of) goal(s), in order to participate in a* benchmarking competition.

The list of *tasks* for RoCKIn@Work is defined in Section 4.

Definition 6.4 (*Benchmarking*)

The process of evaluating the performance of a given robot system *or one of its* functional modules, *according to a specified* metric.

Definition 6.5 (*Benchmark*)

The union of one or more benchmarking experiments *and a set of* metrics *according to which the course and the outcome of the experiments – described by suitable data acquired during the experiments – will be evaluated.*

Definition 6.6 (*Functional Benchmark*)

A benchmark *which aims at evaluating the quality and effectiveness of a specific* functional module *of a* robot system *in the context of one or more* scenarios.

Definition 6.7 (*Task Benchmark*)

A benchmark *which aims at evaluating the quality of the overall execution of a* task *by a* robot system *in the context of a single* scenario.

Definition 6.8 (*Score*)

The result obtained when a robot system *is subjected to a* benchmark *(*task benchmark *or* functionality benchmark*).*

The *scores* will be used in the RoCKIn@Work competitions to order the teams according to their performance in *tasks* and *functionalities*.

Benchmarking data will be logged by the Organizing Committee for offline, ex-post analysis of team performances in the RoCKIn@Work *tasks* and *functionalities*, so as to provide relevant scientific information, such as the impact of *functional modules* performance in the *robot system task* performance, or to improve the scoring system of future RoCKIn@Work competitions.

6.2 Structure of the Competition

Each *task benchmark* and *functional benchmark* will be performed by each of the competing teams several times, to ensure some level of repeatability of the results.

Task benchmarks and *functional benchmarks* will be executed as much as possible in parallel, i.e., while one team executes a *task benchmark*, another team executes a *functional benchmark* simultaneously in another area of the arena.

7 RoCKIn@Work Award Categories

RoCKIn Competition 2014 awards will be given in the form of cups for the best teams, as specified below. Every team will also receive a plaquette with the RoCKIn logo and a certificate. Awards will be given to the best teams in RoCKIn@Work *task benchmarks*, *functional benchmarks* and overall.

7.1 Awards for Task Benchmarks

The team with the highest score in each of the three *task benchmarks* will be awarded a cup ("RoCKIn@Work 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.

7.2 Awards for Functionality Benchmarks

The two top teams in the score raking for each of the three *functionality benchmarks* will be awarded a cup ("RoCKIn@Work Best-in-class Functionality Benchmark <*functionality benchmark* title>" and 'RoCKIn@Work Second-Best-in-class Functionality Benchmark <*functionality benchmark* title>").

When less than three teams participate in a given *functionality benchmark*, only the "RoCKIn-@Home Best-in-class Functionality Benchmark <*functionality benchmark* title>" award will be given to a team, and this will occur only if the Executive and Technical Committees consider that team's performance as excellent.

7.3 Competition Winners

Teams participating in RoCKIn@Work Competition 2014 will be ranked taking into account their overall rank in all the *task benchmarks*.

The overall ranking will be obtained by combining task benchmark rankings using the Social Welfare principle (see `http://en.wikipedia.org/wiki/Social_welfare_function`); the overall winning team of RoCKIn@Work Competition 2014 will be the top team in this combined ranking, and will receive the corresponding award cup ("Best Team RoCKIn@Work Competition 2014"). The second and third placed teams in the ranking will also receive award cups (respectively "2nd place RoCKIn@Work Competition 2014" and "3rd place RoCKIn@Work Competition 2014")).

The three awards will be given only if more than 5 teams participate in the competition. Otherwise, only the best team will be awarded, except if it is the single team participating, in which case the Executive and Technical Committees must consider that team performance of exceptional level so as for the team to be awarded. Only teams performing the total of the three tasks will be considered for the "Best Team RoCKIn@Work Competition 2014" award.

8 RoCKIn@Work Organization

8.1 RoCKIn@Work Management

The management structure of RoCKIn@Work has been divided into three committees, namely *Executive Committee*, *Technical Committee* and the *Organization Committee*. The roles and responsibilities of those committees are described in the following paragraphs.

8.1.1 RoCKIn@Work Executive Committee

The Executive Committee (EC) is represented by the coordinators of each RoCKIn partner related to the respective activity area. The committee is mainly responsible for the overall coordination of RoCKIn@Work activities and especially for dissemination in the scientific community.

- Pedro Lima (Instituto Superior Técnico, Portugal)
- Daniele Nardi (Sapienza Università di Roma, Italy)
- Gerhard Kraetzschmar (Bonn-Rhein-Sieg University, Germany)
- Rainer Bischoff (KUKA Laboratories GmbH, Germany)
- Matteo Matteucci (Politecnico di Milano, Italy)

8.1.2 RoCKIn@Work Technical Committee

The Technical Committee (TC) is responsible for the rules of the league. Each member of the committee is involved in maintaining and improving the current rule set and also in the adherence of these rules. Other responsibilities include the qualification of teams, handling general technical issues within the league, deciding about giving awards in case the number of participants is lower than the thresholds specified in Section 7, as well as resolving any conflicts inside the league during an ongoing competition. The members of the committee are further responsible for maintaining the RoCKIn@Work Infrastructure.

The Technical Committee currently consists of the following members:

- Pedro Miraldo (Instituto Superior Técnico, Portugal)
- Alberto Pretto (Sapienza Università di Roma, Italy)
- Rhama Dwiputra (Bonn-Rhein-Sieg University, Germany)
- Jakob Berghofer (KUKA Laboratories GmbH, Germany)
- Francesco Amigoni (Politecnico di Milano, Italy)

This committee can also include members of the Executive Committee.

8.1.3 RoCKIn@Work Organizing Committee

The Organizing Committee (OC) is responsible for the actual implementation of the competition, i.e. providing everything what is required to perform the various tests. Specifically, this means providing setting up the test arena(s), providing any kind of objects (e.g. manipulation objects), scheduling the tests, assigning and instructing referees, recording and publishing (intermediate) competition results and any other kind of management and advertisement duties before, during and after the competition.

The Organizing Committee currently consists of the following members:

- **Chair:** Pedro Lima (Instituto Superior Técnico, Portugal)
- João Mendes (Instituto Superior Técnico, Portugal)
- Roberto Capobianco(Sapienza Università di Roma, Italy)
- Frederik Hegger (Bonn-Rhein-Sieg University, Germany)
- Graham Buchanan, (InnoCentive EMEA, U.K.)

8.2 RoCKIn@Work Infrastructure

8.2.1 RoCKIn@Work Web Page

The official RoCKIn@Work website can be reached at

`http://rockinrobotchallenge.eu/work.php`

On those web pages, teams can find introductory information about the league itself as well as relevant information about upcoming events, the most recent version of the rulebook, videos and pictures of past competitions and links to further resources like the official mailing list or wiki.

8.2.2 RoCKIn@Work Mailing List

The official RoCKIn@Work mailing list maintained by the league is as follows

`rockin-at-work@rockinrobotchallenge.eu`

Anyone can subscribe by using the following subscription page.

`http://rockinrobotchallenge.eu/mailman/listinfo/rockin-at-work`

Every subscriber is requested to register either with an email address which already encodes the real name or alternatively specify it in the provided field at the subscription page. In order to prevent the mailing list from spammers, this mailing list is moderated.

The mailing list will be used for any kind of official announcement, e.g. upcoming RoCKIn-@Work competitions, rule changes, registration deadlines or infrastructure changes. Teams are also welcome to raise any kind of question regarding the league on this list.

8.3 RoCKIn@Work Competition Organization

8.3.1 Qualification and Registration

Participation in RoCKIn@Work requires successfully passing a qualification procedure. This procedure is to ensure a well-organized competition event and the safety of participants. Depending on constraints imposed by a particular site or the number of teams interested to participate, it may not be possible to admit all interested teams to the competition.

All teams that intend to participate at the competition have to perform the following steps (using the forms at the web site http://rockincompetition.eu):

1. Preregistration (deadline: 15 April 2014) – optional
2. Submission of qualification material (e.g. team description paper and video; deadline: 9 May 2014) – mandatory
3. Final registration (between 1 June and 1 July 2014) – mandatory, and for qualified teams only

Preregistration A team must provide the following information during the preregistration process:

- Team name + Affiliation
- Team leader name
- Team leader email address
- Expected number of team members
- Whether the team plan to bring their own robot or not
- Middleware used for software development

This step can be considered as an *Intention of Participation* declaration and serves as planning basis for the Organizing Committee.

Qualification The qualification process serves a dual purpose: It should allow the Technical Committee to assess the safety of the robots a team intents to bring to a competition, and it should allow to rank teams according to a set of evaluation criteria in order to select the most promising teams for a competition, if not all interested teams can be permitted. The TC will select the qualified teams according to the qualification material provided by the teams.

The evaluation criteria will include:

- Team description paper
- Team web site
- Relevant scientific contribution/publications
- Professional quality of robot and software
- Novelty of approach
- Relevance to domestic service robotics
- Performance in other competitions
- Contribution to RoCKIn@Work league (e.g. by organization of events or provision and sharing of knowledge)

The Team Description Paper (TDP) is a central element of the qualification process and has to be provided by each team as part of the qualification process. The TDP should at least contain the following information in the author/title section of the paper:

- Name of the team (title)
- Team members (authors), including the team leader
- Link to the team web site
- Contact information

The body of the TDP should contain information on the following: focus of research/research interests:

- Description of the hardware, including an image of the robot(s)
- Description of the software, esp. the functional and software architectures
- Main involved research areas in the team work
- Innovative technology (if any)
- Reusability of the system or parts thereof
- Applicability and relevance to industrial robotics

The team description paper should cover in detail the technical and scientific approach, while the team web site should be designed for a broader audience. Both the web site and the TDP have to be written in English.

The length of the team description paper is limited to 6 pages and has to be to submitted in the IEEE Conference Proceedings format[12].

Registration Only if a team has passed the qualification procedure successfully it is allowed to register officially for the competition and has to provide additional information e.g. the exact number of team members. Further information about the registration procedure will be communicated through the mailing list of qualified teams. The number of people to register per team may be unlimited, but during the competition the organizers will provide space only for 6 persons to work at tables in the team area. During the final registration, each team has to designate one member as team leader. A change of the team leader must be communicated to the Organizing Committee.

[12] `http://www.ieee.org/conferences_events/conferences/publishing/templates.html`

8.3.2 Setup and Schedule

RoCKIn Competition 2014 will take place in La Cité de L'Espace in Toulouse, France, from 26-30 November 2014.

24–25 November will be the assembly days, during which the arenas, team areas, power, audiovisual equipment and other infrastructure will be put in place.

26–27 November will be setup days, that the teams can use to unpack their robots, calibrate the robot systems, and get information about the test bed, important objects and other relevant details. The site will be closed to the public.

There will be three competition days: 28, 29 and 30 November. During those days, the competitions will occur following the procedures and rules described in the subsections of this document with the same title. The site will be accessible to the public during the actual competitions.

The award and closing ceremony will take place in the evening of the last day, 30 November 2014.

Several satellite events, with the participation of industry and academia stakeholders, will take place during the five days of the main event. These include talks by members of RoCKIn's Advisory Board, and the assessment of the Competition by the members of RoCKIn's Experts Board.

Schedule: For the scheduling of particular stages, tests, and technical challenges of the competition the following applies:

- The exact schedule of task-functionality tests will be announced one week before the actual competition by the OC on both the website and the mailing list of qualified teams.
- In order to avoid to much "traffic" inside the testbed, an additional schedule only for test slots will be established on site by the OC.
- A set of test slots will be assigned to each team between the official test slots, where a team has exclusive access to the testbed without any other team/robot inside the arena.

Setup: For the arrival, setup, and preparation of teams participating in the competition, the following procedures apply:

- A first draft of the rulebook will be made public on 1 April 2014.
- Revisions will be possible and updated in the online versions of the document, based on suggestions of all relevant stakeholders (including pre-registered and registered teams) until 31 May 2014.
- The final version of the rulebook will be made public, no later than eight weeks before the actual event, by the the TC, including all the items referred as open in this document (e.g., some benchmarking and scoring items) and revisions resulting from the discussion referred in the previous item.
- The competition side will be divided into a competition arena and a team area.
- The competition arena consists of one or more testbeds (the arena) and is open for public.
- The arena must be kept clean and in a representable condition all the time.
- The team area is a dedicated area only for team members, no public access here.
- Each team will be assigned to a designated area with tables and chairs (based on the number of team members), with power sockets, a shelf internet connection and a reasonable area to park their robot and other equipment.

8.3.3 Competition Execution

- Referees will be determined by the OC out of the group of team leaders and TC members.
- The referees ensure the correct execution of a benchmark run, are in charge of keeping the time and counting the scores, being always helped by a TC or OC member.
- In case of any dangerous situation the referees are allowed to immediately stop a run and trigger the emergency stop functionality of the respective robot.
- The official language for all kind of communication within the league is English (e.g., team leader meetings, announcements, schedule)
- The order in which the teams have to perform a particular benchmark run will be determined by a draw through the OC.
- The order will be announced on the day before the actual run.
- No team members or other persons are allowed to be in the arena during an official benchmark run (only if the rulebook explicitly allows/requires this).
- Regular team leader meetings (every day) will be organized and announced by the TC/OC during the competition in order to discuss open issues for upcoming benchmark runs.

8.3.4 Measurements Recording

Several variables of interest will be recorded by the EC, TC and OC during the actual runs of the teams during the competition, while performing their *task* and *functionality benchmarks*. Some of these will be performed by RoCKIn equipment, though requiring the installation of markers on the team robots. Other logging will require the teams to accommodate, in their software, modules that respond to solicitations from test bed-installed software. Details on these procedures will be provided closer to the competition dates, but the teams must be ready to commit to such requirements as one of the key requirements to be selected for the RoCKIn competitions. The logging and benchmarking activities will be under the responsibility of Giulio Fontana (Politecnico di Milano, Italy).

References

[1] J. Berghofer, R. Dwiputra, G. K. (editors), A. Ahmad, F. Amigoni, I. Awaad, J. Berghofer, R. Bischoff, A. Bonarini, R. Dwiputra, F. Hegger, N. Hochgeschwender, L. Iocchi, G. Kraetzschmar, P. Lima, M. Matteucci, D. Nardi, and S. Schneider, "RoCKIn@Work – Innovation in Industrial Mobile Manipulation – RoCKIn@Work in a Nutshell," KUKA Laboratories GmbH, Bonn-Rhein-Sieg University of Applied Sciences, Augsburg, Germany, and Sankt Augustin, Germany, Deliverable D-2.1.4 nutshell (public), The RoCKIn Project (FP7-ICT-601012 grant agreement number 601012), March 2014.

